High (Physical) Security & Lightweight (Symmetric) Cryptography

François-Xavier Standaert
UCL Crypto Group, Belgium

HIGHLIGHT – LIGHTCRYPTO, November 2016
Outline

• Preliminary questions / definitions
• Side-channel basics (attack steps)
• Noise (aka hardware) is not enough
• Noise amplification (aka masking)
• Reductions help (aka leakage resilience)
• Mitigating hardware defaults (is hard)
• Transparency is needed (open source)
• Summary and conclusions
Outline

• Preliminary questions / definitions
• Side-channel basics (attack steps)
• Noise (aka hardware) is not enough
• Noise amplification (aka masking)
• Reductions help (aka leakage resilience)
• Mitigating hardware defaults (is hard)
• Transparency is needed (open source)
• Summary and conclusions
• Preliminary questions / definitions
• Side-channel basics (attack steps)
• Noise (aka hardware) is not enough
• Noise amplification (aka masking)
• Reductions help (aka leakage resilience)
• Mitigating hardware defaults (is hard)
• Transparency is needed (open source)
• Summary and conclusions
What does secure mean? (I)

• For block ciphers, best attack has:
  • Time complexity $2^{n_1}$
  • Data complexity $2^{n_2}$
  • Memory complexity $2^{n_3}$
What does secure mean? (I)

• For block ciphers, best attack has:
  • Time complexity $2^{n_1}$
  • Data complexity $2^{n_2}$
  • Memory complexity $2^{n_3}$

• With typical security parameters:
  • $n_L = 80, n_S = 128, n_{PQ} = 256$
• Function of the algorithms’ deployment time and the adversary’s computational power
What does secure mean? (II)

• When also considering physical attacks:
  • Measurement complexity $2^{m_1}$
  • Fault complexity $2^{m_2}$
• Typical security parameters: ???
What does secure mean? (II)

- When also considering physical attacks:
  - Measurement complexity $2^{m_1}$
  - Fault complexity $2^{m_2}$
- Typical security parameters: ???

Q1. Deployment time of implementations?
- IMO not much less (minimum 5-10 years)
- So we can gain a factor 2 to 4 (i.e., 1,2 bits)

Q2. What’s the adversary’s measurement power?
What does secure mean? (III)

Q2. What’s the adversary’s measurement power?

• Generic answer:
  • “Cost” of collecting one side-channel meas. vs. cost of collecting one pt/ct pair?
    • Min. $\times 2^{10}$, avg. $\times 2^{20}$, opt. $\times 2^{30}$
  • Roughly assume $\approx$ the same for faults
  • So we can gain 10 to 30 bits (roughly)

$\Rightarrow m_L > 60$ and $m_S > 100$ (no clue about $m_{PQ}$)

What does secure mean? (IV)

• Specific answer (if physical access is limited):
  • $m_{VL} > 40$ (≈ some days/weeks)
  • $m_{UL} > 20$ (≈ some hours)
  • Excluding network access (timing attacks)!
What does secure mean? (IV)

• Specific answer (if physical access is limited):
  • \( m_{VL} > 40 \) (\( \approx \) some days/weeks)
  • \( m_{UL} > 20 \) (\( \approx \) some hours)
  • Excluding network access (timing attacks)!

• Fact 1. Currently, we mostly design for VL/UL physical security / Fact 2. Current physical security evaluations are limited to VL security
What does secure mean? (IV)

- Specific answer (if physical access is limited):
  - $m_{VL} > 40$ (∼ some days/weeks)
  - $m_{UL} > 20$ (∼ some hours)
  - Excluding network access (timing attacks)!

- Fact 1. Currently, we mostly design for VL/UL physical security / Fact 2. Current physical security evaluations are limited to VL security

- VL or UL security ∼ no security
• Preliminary questions / definitions
• **Side-channel basics (attack steps)**
  • Noise (aka hardware) is not enough
  • Noise amplification (aka masking)
  • Reductions help (aka leakage resilience)
• Mitigating hardware defaults (is hard)
• Transparency is needed (open source)
• Summary and conclusions
Standard DPA

 leakage trace

\( l_i \)

comparison

\( \tilde{k} \)

subkey candidate

\( m_i^{k^*} \)

S-box

target intermediate value \( V_i \)

\( x_i \)

\( k \)

executed operations
Standard DPA

measurement & pre-processing

leakage trace

executed operations

S-box

$\begin{align*}
x_i &\quad y_i &\quad z_i &\quad V_i
\end{align*}$

$\begin{align*}
k &\quad m_i^{k^*}
\end{align*}$

comparison

$\tilde{k}$

subkey candidate
Standard DPA

measurement & pre-processing

leakage trace

comparison

\( \tilde{k} \)

subkey candidate

\( m_i^{k^*} \)

\( x_i \)

\( k \)

target intermediate value

\( y_i \)

\( z_i \)

prediction & modeling

executed operations

S-box

model
Standard DPA

leakage trace

measurement & pre-processing

info $I_i$

comparison

$	ilde{k}$
subkey candidate

exploitation

$m_i^{k^*}$

prediction & modeling

executed operations

$S$-box

target intermediate value $Vi$

$x_i$

$y_i$

$k$
Standard DPA

leakage trace

measurement & pre-processing

post-processing

comparison

exploitation

\[ \tilde{k} \]

subkey candidate

\[ m_j^{k^*} \]

executed operations

S-box

\[ k \]

target intermediate value

\[ x_i \]

\[ y_i \]

\[ z_i \]

prediction & modeling
Measurement & pre-processing

- Noise reduction via good setups
- Filtering, averaging (FFT, SSA, ...)
- Detection of Points-Of-Interest (POI)
- Dimensionality reduction (PCA, LDA, ...)
- ...

Victor Lomné, Emmanuel Prouff, Thomas Roche: *Behind the Scene of Side Channel Attacks*. ASIACRYPT (1) 2013: 506-525.
Measurement & pre-processing

- Noise reduction via good setups
- Filtering, averaging (FFT, SSA, ...)
- Detection of Points-Of-Interest (POI)
- Dimensionality reduction (PCA, LDA,...)
- ...

- Inherently heuristic (!) ⇒ hard to determine what is the optimal solution (≠ next steps)

Victor Lomné, Emmanuel Prouff, Thomas Roche: *Behind the Scene of Side Channel Attacks*. ASIACRYPT (1) 2013: 506-525.
• General case: profiled DPA
  • Build “templates”, i.e., \( \hat{f}(l_i|k, x_i) \)
    • e.g. Gaussian, regression-based
  • Which directly leads to \( \Pr[k|l_i, x_i] \)

Prediction and modeling

• General case: profiled DPA
  • Build “templates”, i.e., \( \hat{f}(l_i|k, x_i) \)
    • e.g. Gaussian, regression-based
  • Which directly leads to \( \Pr[k|l_i, x_i] \)

• “Simplified” case: non-profiled DPA
  • Just assumes some model
    • e.g., \( m_i^{k^*} = \text{HW}(z_i) \)

Prediction and modeling

• General case: profiled DPA
  • Build “templates”, i.e., $\hat{f}(l_i|k, x_i)$
    • e.g. Gaussian, regression-based
  • Which directly leads to $\Pr[k|l_i, x_i]$

• “Simplified” case: non-profiled DPA
  • Just assumes some model
    • e.g., $m_i^k = \text{HW}(z_i)$

• Separation: only profiled DPA is guaranteed to succeed against any leaking device (!)

• Profiled case: maximum likelihood

Omar Choudary, Markus G. Kuhn: Efficient Template Attacks. CARDIS 2013: 253-270
Exploitation

• Profiled case: maximum likelihood

• Unprofiled case:
  • Difference-of-Means
  • Correlation (CPA)
  • « On-the-fly » regression
  • Mutual Information Analysis (MIA)

Exploitation

- Profiled case: maximum likelihood
- Unprofiled case:
  - Difference-of-Means
  - Correlation (CPA)
  - « On-the-fly » regression
  - Mutual Information Analysis (MIA)

- Advanced topic: analytical (algebraic) attacks

\[ \tilde{k} = \arg \max_{k^*} \prod_{i=1}^{q} \frac{1}{\sqrt{2 \cdot \pi \cdot \sigma(L)}} \exp \left( -\frac{1}{2} \left( \frac{l_i - m_i^{k^*}}{\sigma(L)} \right)^2 \right) \]

- More efficient (why?)
- Outputs probabilities

\[ \tilde{k} = \arg \max_{k^*} \frac{E(L \cdot M^{k^*}) - E(L) \cdot E(M^{k^*})}{\sigma(L) \cdot \sigma(M^{k^*})} \]

- Less efficient (why?)
- Outputs scores

Illustration

Gaussian templates

CPA

correct key candidate

number of measurement queries

likelihood

number of measurement queries

correlation
• CPA: $\tilde{k} = \arg\max_k \frac{E(L \cdot M^{k^*}) - E(L) \cdot E(M^{k^*})}{\sigma(L) \cdot \sigma(M^{k^*})}$
• CPA: \( \tilde{k} = \arg\max_{k^*} \frac{E(L \cdot M^{k*}) - E(L) \cdot E(M^{k*})}{\sigma(L) \cdot \sigma(M^{k*})} = 0 \) (normalization)
CPA vs. Gaussian templates

• CPA: \[ \tilde{k} = \arg\max_{k^*} \frac{\mathbb{E}(L \cdot M^{k^*}) - \mathbb{E}(L) \cdot \mathbb{E}(M^{k^*})}{\sigma(L) \cdot \sigma(M^{k^*})} \]

\[= 0 \text{ (normalization)}\]

independent of \( k^* \)
• CPA: \[ \tilde{k} = \arg\max_k \frac{E(L \cdot M^{k*}) - E(L) \cdot E(M^{k*})}{\sigma(L) \cdot \sigma(M^{k*})} = 0 \text{ (normalization)} \]

independent of \( k^* \)

asymptotically independent of \( k^* \)
CPA vs. Gaussian templates

- CPA: $\tilde{k} \propto \arg\max_{k^*} E(L \cdot M^{k^*})$
• CPA: $\tilde{k} \propto \arg\max_{k^*} E(L \cdot M^{k^*})$

• Gaussian templates:

$$\tilde{k} = \arg\max_{k^*} \prod_{i=1}^{q} \frac{1}{\sqrt{2\pi} \cdot \sigma(L)} \cdot \exp \left( -\frac{1}{2} \cdot \left( \frac{l_i - m_i^{k^*}}{\sigma(L)} \right)^2 \right)$$
CPA vs. Gaussian templates

- CPA: \( \tilde{k} \propto \arg \max_{k^*} E(L \cdot M^{k^*}) \)

- Gaussian templates:

\[
\tilde{k} = \arg \max_{k^*} \prod_{i=1}^{q} \frac{1}{\sqrt{2 \cdot \pi \cdot \sigma(L)}} \cdot \exp \left( -\frac{1}{2} \cdot \left( \frac{l_i - m_i^{k^*}}{\sigma(L)} \right)^2 \right)
\]

independent of \( k^* \)
• CPA: \( \tilde{k} \propto \arg\max_{k^*} E(L \cdot M^{k^*}) \)

• Gaussian templates:

\[
\tilde{k} \propto \arg\max_{k^*} \prod_{i=1}^{q} \exp \left( -\frac{1}{2} \cdot \left( \frac{l_i - m_i^{k^*}}{\sigma(L)} \right)^2 \right)
\]
CPA vs. Gaussian templates

- CPA: \( \tilde{k} \propto \arg\max_{k^*} E(L \cdot M^{k^*}) \)

- Gaussian templates:

\[
\tilde{k} \propto \arg\min_{k^*} E(L^2) - 2 \cdot E(L \cdot M^{k^*}) + E((M^{k^*})^2)
\]
CPA vs. Gaussian templates

- **CPA:** \( \tilde{k} \propto \operatorname{argmax}_{k^*} \mathbb{E}(L \cdot M^{k^*}) \)

- **Gaussian templates:**

\[
\tilde{k} \propto \operatorname{argmin}_{k^*} \mathbb{E}(L^2) - 2 \cdot \mathbb{E}(L \cdot M^{k^*}) + \mathbb{E}((M^{k^*})^2)
\]

independent of \( k^* \)
• CPA: \( \tilde{k} \propto \arg\max_{k^*} \mathbb{E}(L \cdot M^{k^*}) \)

• Gaussian templates:

\[
\tilde{k} \propto \arg\min_{k^*} \mathbb{E}(L^2) - 2 \cdot \mathbb{E}(L \cdot M^{k^*}) + \mathbb{E}((M^{k^*})^2)
\]

independent of \( k^* \) asymptotically independent of \( k^* \)
CPA vs. Gaussian templates

• CPA: $\tilde{k} \propto \arg\max_{k^*} E(L \cdot M^{k^*})$

• Gaussian templates: $\tilde{k} \propto \arg\max_{k^*} E(L \cdot M^{k^*})$
CPA vs. Gaussian templates

- CPA: \( \tilde{k} \propto \arg \max_{k^*} E(L \cdot M^{k^*}) \)

- Gaussian templates: \( \tilde{k} \propto \arg \max_{k^*} E(L \cdot M^{k^*}) \)

\( \Rightarrow \) Both attacks are asymptotically equivalent

- For 1st-order leakages
  - i.e., unprotected implementations
  - Given they exploit the same model
CPA vs. Gaussian templates

- CPA: \( \tilde{k} \propto \arg\max_{k^*} E(L \cdot M^{k^*}) \)

- Gaussian templates: \( \tilde{k} \propto \arg\max_{k^*} E(L \cdot M^{k^*}) \)

\( \Rightarrow \) Both attacks are asymptotically equivalent

- For 1st-order leakages
- i.e., unprotected implementations
- Given they exploit the same model

\( \Rightarrow \) Gaussian templates outperforms CPA because it (usually) exploits a better (profiled) model

Outline

- Preliminary questions / definitions
- Side-channel basics (attack steps)
- **Noise (aka hardware) is not enough**
- Noise amplification (aka masking)
- Reductions help (aka leakage resilience)
- Mitigating hardware defaults (is hard)
- Transparency is needed (open source)
- Summary and conclusions
• **Lemma 1.** The mutual information between two normally distributed random variables $X, Y$ with means $\mu_X, \mu_Y$ and variances $\sigma_X^2, \sigma_Y^2$ equals:

$$\text{MI}(X; Y) = -\frac{1}{2} \log_2(1 - \rho(X, Y)^2)$$
• **Lemma 1.** The mutual information between two normally distributed random variables $X, Y$ with means $\mu_X, \mu_Y$ and variances $\sigma_X^2, \sigma_Y^2$ equals:

$$MI(X; Y) = -\frac{1}{2} \log_2(1 - \rho(X, Y)^2)$$

• **Lemma 2.** In a CPA, the number of samples required to distinguish the correct key with model $M_k$ from the other key candidates with models $M_{k*}$ is $\propto \frac{c}{\rho(M_k, L)^2}$ (with $c$ a small constant depending on the SR & # of key candidates)

Lemma 3. Let $X, Y$ and $L$ be three random variables s.t. $Y = X + N_1$ and $L = Y + N_2$ with $N_1$ and $N_2$ two additive noise variables. Then:

$$\rho(X, L) = \rho(X, Y) \cdot \rho(Y, L)$$
• **Lemma 3.** Let $X, Y$ and $L$ be three random variables s.t. $Y = X + N_1$ and $L = Y + N_2$ with $N_1$ and $N_2$ two additive noise variables. Then:

$$
\rho(X, L) = \rho(X, Y) \cdot \rho(Y, L)
$$

• **Lemma 4.** The correlation coefficient between the sum of $n$ independent and identically distributed random variables and the sum of the first $m < n$ of these equals $\sqrt{m/n}$

• FPGA implementation of the AES
• Adversary targeting the 1st byte of key
• Hamming weight leakage function/model
• 8-bit loop architecture broken in 10 traces
• FPGA implementation of the AES
• Adversary targeting the 1st byte of key
• Hamming weight leakage function/model
• 8-bit loop architecture broken in 10 traces

• How does the attack data complexity scale
  • For a 32-bit architecture?
    • i.e., with 24 bits of « algorithmic noise »
  • For a 128-bit architecture?
    • i.e., with 120 bits of « algorithmic noise »
• Hint: $L = M + N = (M_P + M_U) + N$
• Hint: \( L = M + N = (M_P + M_U) + N \)

• Lemma 3: \( \rho(M_P, L) = \)
• Hint: \( L = M + N = (M_P + M_U) + N \)

• Lemma 3: \( \rho(M_P, L) = \rho(M_P, M) \cdot \rho(M, L) \)

• Lemma 4: \( \rho(M_P, M) = ? \)
  • For the 8-bit architecture:
  • For the 32-bit architecture:
  • For the 128-bit architecture:
• Hint: $L = M + N = (M_P + M_U) + N$

• Lemma 3: $\rho(M_P, L) = \rho(M_P, M) \cdot \rho(M, L)$

• Lemma 4: $\rho(M_P, M) = ?$
  • For the 8-bit architecture: $\sqrt{8/8}$
  • For the 32-bit architecture:
  • For the 128-bit architecture:
• Hint: \( L = M + N = (M_P + M_U) + N \)

• Lemma 3: \( \rho(M_P, L) = \rho(M_P, M) \cdot \rho(M, L) \)

• Lemma 4: \( \rho(M_P, M) = ? \)
  - For the 8-bit architecture: \( \sqrt{8/8} \)
  - For the 32-bit architecture: \( \sqrt{8/32} \)
  - For the 128-bit architecture: \( \sqrt{8/128} \)
• Hint: \( L = M + N = (M_P + M_U) + N \)

• Lemma 3: \( \rho(M_P, L) = \rho(M_P, M) \cdot \rho(M, L) \)

• Lemma 4: \( \rho(M_P, M) = ? \)
  • For the 8-bit architecture: \( \sqrt{8/8} \)
  • For the 32-bit architecture: \( \sqrt{8/32} \)
  • For the 128-bit architecture: \( \sqrt{8/128} \)

• Lemma 2: \( \frac{c}{(\sqrt{8/8} \cdot \rho(M,L))^2} = 10 \)
• Data complexity for the 32-bit case:
• Data complexity for the 128-bit case:
• Data complexity for the 32-bit case: 40
• Data complexity for the 128-bit case: 160

• Is noise an efficient countermeasure?
• Data complexity for the 32-bit case: 40
• Data complexity for the 128-bit case: 160

• Is noise an efficient countermeasure?
• 32-bit case: security \times 4, cost \times ?
• Data complexity for the 32-bit case: 40
• Data complexity for the 128-bit case: 160

• Is noise an efficient countermeasure?
  • 32-bit case: security $\times 4$, cost $\times 4$

• How to trade data for time?
• Data complexity for the 32-bit case: 40
• Data complexity for the 128-bit case: 160

• Is noise an efficient countermeasure?
  • 32-bit case: security $\times 4$, cost $\times 4$

• How to trade data for time?
  • Target more than 8 bits at once
  • Cancels (a part of) the « algorithmic noise »
  • e.g., 32-bit architecture: $\rho(M_P, M) =$
• Data complexity for the 32-bit case: 40
• Data complexity for the 128-bit case: 160

• Is noise an efficient countermeasure?
  • 32-bit case: security $\times$ 4, cost $\times$ 4

• How to trade data for time?
  • Target more than 8 bits at once
  • Cancels (a part of) the « algorithmic noise »
  • e.g., 32-bit architecture: $\rho(M_P, M) = \sqrt{32/32}$
  • (10 < data complexity < 40 because of $c$)
Outline

• Preliminary questions / definitions
• Side-channel basics (attack steps)
• Noise (aka hardware) is not enough
• Noise amplification (aka masking)
• Reductions help (aka leakage resilience)
• Mitigating hardware defaults (is hard)
• Transparency is needed (open source)
• Summary and conclusions
Masking & 2\textsuperscript{nd}-order DPA

\begin{itemize}
\item \text{leakage trace}
\item \text{combination (optional)}
\item \text{comparison}
\item \text{adversary}
\item \text{subkey candidate}
\item \text{executed operations}
\end{itemize}

\begin{align*}
Sbox(x_i \oplus k) \oplus b_i
\end{align*}
More generally (I)

- Let \( z = S(x \oplus k) = S(y) \) be a leaking S-box
- Let \( y = y_1 \oplus y_2 \oplus \cdots \oplus y_d \) be a sharing of \( y \)

![Diagram showing leakage over time samples]

- Perform computations on “shared” variables
• **Linear operations:** \( f(a) = f(a_1) \oplus f(a_2) \oplus \cdots \oplus f(a_d) \)
More generally (II)

- **Linear operations:** \( f(a) = f(a_1) \oplus f(a_2) \oplus \cdots \oplus f(a_d) \)

- **Multiplications:** \( c = a \times b \) in three steps

More generally (II)

- Linear operations: \( f(a) = f(a_1) \oplus f(a_2) \oplus \cdots \oplus f(a_d) \)

- Multiplications: \( c = a \times b \) in three steps

\[
\begin{bmatrix}
    a_1b_1 & a_1b_2 & a_1b_3 \\
    a_2b_1 & a_2b_2 & a_2b_3 \\
    a_3b_1 & a_3b_2 & a_3b_3
\end{bmatrix}
\]

partial products

More generally (II)

- Linear operations: \( f(a) = f(a_1) \oplus f(a_2) \oplus \cdots \oplus f(a_d) \)

- Multiplications: \( c = a \times b \) in three steps

\[
\begin{bmatrix}
a_1 b_1 & a_1 b_2 & a_1 b_3 \\
a_2 b_1 & a_2 b_2 & a_2 b_3 \\
a_3 b_1 & a_3 b_2 & a_3 b_3 \\
\end{bmatrix} + \begin{bmatrix}
0 & r_1 & r_2 \\
-r_1 & 0 & r_3 \\
-r_2 & r_3 & 0 \\
\end{bmatrix}
\]

partial products refreshing

More generally (II)

- **Linear operations:** \( f(a) = f(a_1) \oplus f(a_2) \oplus \cdots \oplus f(a_d) \)

- **Multiplications:** \( c = a \times b \) in three steps

\[
\begin{bmatrix}
a_1 b_1 & a_1 b_2 & a_1 b_3 \\
a_2 b_1 & a_2 b_2 & a_2 b_3 \\
a_3 b_1 & a_3 b_2 & a_3 b_3 \\
\end{bmatrix}
+ \begin{bmatrix}
0 & r_1 & r_2 \\
-r_1 & 0 & r_3 \\
-r_2 & r_3 & 0 \\
\end{bmatrix}
\Rightarrow \begin{bmatrix}
c_1 \\
c_2 \\
c_3 \\
\end{bmatrix}
\]

partial products    refreshing    compression

More generally (II)

- **Linear operations:**  \( f(a) = f(a_1) \oplus f(a_2) \oplus \ldots \oplus f(a_d) \)

- **Multiplications:** \( c = a \times b \) in three steps

\[
\begin{bmatrix}
    a_1b_1 & a_1b_2 & a_1b_3 \\
    a_2b_1 & a_2b_2 & a_2b_3 \\
    a_3b_1 & a_3b_2 & a_3b_3
\end{bmatrix} + \begin{bmatrix}
    0 & r_1 & r_2 \\
    -r_1 & 0 & r_3 \\
    -r_2 & r_3 & 0
\end{bmatrix} \Rightarrow \begin{bmatrix}
    c_1 \\
    c_2 \\
    c_3
\end{bmatrix}
\]

- partial products
- refreshing
- compression

⇒ Quadratic overheads & randomness

---

More generally (II)

- **Linear operations:** \( f(a) = f(a_1) \oplus f(a_2) \oplus \cdots \oplus f(a_d) \)

- **Multiplications:** \( c = a \times b \) in three steps

\[
\begin{bmatrix}
a_1 b_1 & a_1 b_2 & a_1 b_3 \\
a_2 b_1 & a_2 b_2 & a_2 b_3 \\
a_3 b_1 & a_3 b_2 & a_3 b_3 \\
\end{bmatrix}
+ \begin{bmatrix}
0 & r_1 & r_2 \\
-r_1 & 0 & r_3 \\
-r_2 & r_3 & 0 \\
\end{bmatrix}
\Rightarrow \begin{bmatrix}
c_1 \\
c_2 \\
c_3 \\
\end{bmatrix}
\]

- partial products
- refreshing
- compression

\( \Rightarrow \) Quadratic overheads & randomness

\( \Rightarrow \) Composable (from gadgets to circuits)

---

CRYPTO 2003: 463-481.

Matthieu Rivain, Emmanuel Prouff: *Provably Secure Higher-Order Masking of AES.*
CHES 2010: 413-427.

Main theorem (informal)

• Assume leakage variables $L_{Z_i} = \delta(Z_i) + N$ s.t.
  • $\text{MI}(Z_i; L_{Z_i}) \leq \frac{c}{d}$ (why $d$? – or $d^2$ in proofs)
  • The leakages of the shares are independent
• For a masking scheme with $d$ shares
• And an adversary using $m$ measurements

• Then: $\text{SR} \leq 1 - (1 - \text{MI}(Z_i; L_{Z_i})^d)^m$
Main theorem (informal)

• Assume leakage variables $L_{Z_i} = \delta(Z_i) + N$ s.t.
  • $\text{MI}(Z_i; L_{Z_i}) \leq \frac{c}{d}$ (multiplications)
  • The leakages of the shares are independent
• For a masking scheme with $d$ shares
• And an adversary using $m$ measurements

• Then: $\text{SR} \leq 1 - \left(1 - \text{MI}(Z_i; L_{Z_i})^d\right)^m$
Main theorem (informal)

• Assume leakage variables $L_{Z_i} = \delta(Z_i) + N$ s.t.
  • $\text{MI}(Z_i; L_{Z_i}) \leq \frac{c}{d}$ (multiplications)
  • The leakages of the shares are independent
• For a masking scheme with $d$ shares
• And an adversary using $m$ measurements

• Then: $\text{SR} \leq 1 - \left(1 - \text{MI}(Z_i; L_{Z_i})^d\right)^m$

• For $m = 1$, $\text{SR} \leq \text{MI}(Z_i; L_{Z_i})^d \propto (\sigma_N^2)^d$
• (Intuitively $\approx$ “noisy” piling up lemma)

Alexandre Duc, Sebastian Faust, François-Xavier Standaert: Making Masking Security Proofs Concrete - Or How to Evaluate the Security of Any Leaking Device. EUROCRYPT (1) 2015: 401-429
1-bit, 2-shares example

(a) \( Z = 0 \), serial.

(b) \( Z = 1 \), serial.

(c) \( Z = 0 \), parallel.

(d) \( Z = 1 \), parallel.
• 1-bit, 2-shares example

(a) $Z = 0$, serial.

(b) $Z = 1$, serial.

(c) $Z = 0$, parallel.

(d) $Z = 1$, parallel.
• Slope of the IT curves = $d$ (if independent leaks)
Wrapping up

• Is masking an efficient countermeasure?
  • Security (data) is exponential in $d$
  • Cost is [...]
Is masking an efficient countermeasure?

Security (data) is exponential in $d$

Cost is [...] quadratic in $d$
• Is masking an efficient countermeasure?
• Security (data) is exponential in \( d \)
• Cost is \([...]\) quadratic in \( d \)

• If the leakages are noisy and independent (!)
Wrapping up

• Is masking an efficient countermeasure?
• Security (data) is exponential in $d$
• Cost is [...] quadratic in $d$

• If the leakages are noisy and independent (!)
• How does the time complexity scale in $d$?
• Is masking an efficient countermeasure?
• Security (data) is exponential in $d$
• Cost is [...] quadratic in $d$

• If the leakages are noisy and independent (!)
• How does the time complexity scale in $d$?
  • Depends on the implem. (e.g., serial or //)
• Preliminary questions / definitions
• Side-channel basics (attack steps)
• Noise (aka hardware) is not enough
• Noise amplification (aka masking)
• **Reductions help (aka leakage resilience)**
• Mitigating hardware defaults (is hard)
• Transparency is needed (open source)
• Summary and conclusions
• Most natural construction: forward-secure PRG

Stateful PRGs

- Most natural construction: forward-secure PRG

- Re-keying impact: bounds the number of (noisy) measurements per key (*prevents averaging*)

Stateless PRFs (or PRPs)

- Most natural construction: GGM tree
Stateless PRFs (or PRPs)

- Most natural construction: GGM tree

- Re-keying impact: bounds the number of noise-free observations per key (*allows averaging*)

The stateful / stateless separation

- Key recovery security (standard DPA):
  - "Bounded security" for the PRG only
  - (Analytical/algebraic attacks not considered)

Sonia Belaïd, Vincent Grosso, François-Xavier Standaert: *Masking and leakage-resilient primitives: One, the other(s) or both?* Cryptography and Communications 7(1): 163-184 (2015)
• A call to a stateless primitive is always needed
  • For initialization / randomization
  • For authentication and encryption
Pragmatic view

• A call to a stateless primitive is always needed
  • For initialization / randomization
  • For authentication and encryption

• But we can try to encrypt large messages with a single call to this (more expensive) primitive

Pragmatic view

• A call to a stateless primitive is always needed
  • For initialization / randomization
  • For authentication and encryption

• But we can try to encrypt large messages with a single call to this (more expensive) primitive

• And to use leakage-resilient PRGs otherwise

• Green: public value, orange: ephemeral secret, red: long-term secret (protected with leak-free F*)
Example I: authentication

- Green: public value, orange: ephemeral secret, red: long-term secret (protected with leak-free $F^*$)
- $\tau$ unforgeable even with leakage (during enc.)
- Security of 1-block $\approx$ security of $l$-blocks
- & high-security levels expected
  - Because it is an unpredictability game!
**Example II: encryption**

- Similar reduction but lower security levels
- Because it is an indistinguishability game!
• In theory, the proof challenge remains open

• Yet, the pragmatic model seems sound
• In theory, the proof challenge remains open


• Yet, the pragmatic model seems sound

• In practice, how to design $F^*$ is open too
• In theory, the proof challenge remains open


• Yet, the pragmatic model seems sound

• In practice, how to design $F^*$ is open too, e.g.,
  • Masking ($\Rightarrow$ bitslice ciphers)

• In theory, the proof challenge remains open


• Yet, the pragmatic model seems sound

• In practice, how to design $F^*$ is open too, e.g.,

• Masking ($\Rightarrow$ bitslice ciphers)


• PRFs with non-standard assumptions

• In theory, the proof challenge remains open


• Yet, the pragmatic model seems sound

• In practice, how to design F* is open too, e.g.,

• Masking (⇒ bitslice ciphers)


• PRFs with non-standard assumptions


• Key homomorphism & fresh re-keying

Christoph Dobraunig, François Koeune, Stefan Mangard, Florian Mendel, François-Xavier Standaert: Towards Fresh and Hybrid Re-Keying Schemes with Beyond Birthday Security. CARDIS 2015: 225-241
A recent proposal (Crypto 2016)

\[ \text{ GenSK } \]

\[ \text{ (T)BC}_{sk}(x) \rightarrow y \]

\[ \text{ R } \rightarrow \text{ sk} \]

\[ \text{ msk} \]

\{ 
  \text{ DPA resistance (masking)} \\
  \text{ SPA resistance (shuffling)} \\
  \text{ leakage requirements} 
\}
A recent proposal (Crypto 2016)

- Cryptographically strong re-keying function
  - \( \text{sk} = \langle R, \text{msk} \rangle = \sum (\langle R, \text{msk}_i \rangle) \)

A recent proposal (Crypto 2016)

- Cryptographically strong re-keying function
  - \( \text{sk} = \langle \text{R, msk} \rangle = \sum (\langle \text{R, msk}_i \rangle) \)
- Security based on hard lattice problems
- Simple & efficient: all computations in \( \mathbb{Z}_{2^m} \)

• Authenticated encryption is also possible
  • But combination with IV misuse is tricky
  • Because controlling the IV transforms ephemeral secrets into long-term ones
• Same reason makes LR-decryption tricky
• Authenticated encryption is also possible
  • But combination with IV misuse is tricky
  • Because controlling the IV transforms ephemeral secrets into long-term ones
  • Same reason makes LR-decryption tricky

• Full misuse resistance does not seem possible
  • (In the symmetric crypto setting)
⇒ Current answer: ciphertext integrity with misuse (possible because unpredictability-based)

Francesco Berti and François Koeune and Olivier Pereira and Thomas Peters and François-Xavier Standaert: Leakage-Resilient and Misuse-Resistant Authenticated Encryption. IACR ePrint, 2016
Outline

• Preliminary questions / definitions
• Side-channel basics (attack steps)
• Noise (aka hardware) is not enough
• Noise amplification (aka masking)
• Reductions help (aka leakage resilience)
• **Mitigating hardware defaults (is hard)**
• Transparency is needed (open source)
• Summary and conclusions
• Can break leakage independence requirements

⇒ Makes secure masked implementation hard to obtain

• Can break leakage independence requirements

⇒ Makes secure masked implementation hard to obtain

• e.g., recombine the shares of a masking scheme

• Default-tolerant protections would be (very) handy

Outline

• Preliminary questions / definitions
• Side-channel basics (attack steps)
• Noise (aka hardware) is not enough
• Noise amplification (aka masking)
• Reductions help (aka leakage resilience)
• Mitigating hardware defaults (is hard)
• Transparency is needed (open source)
• Summary and conclusions
Security evaluation tools

Standard practice

Attack-based evaluations

Bounds

Measurements

Success probability
Security evaluation tools

Standard practice

attack-based evaluations

bounds

computation

success probability

1000 2000 3000 measurements

$2^{128}$ $2^{64}$ $2^0$

$2^{30}$ $2^{40}$ $2^{80}$
Security evaluation tools

standard practice

attack-based evaluations

proof-based evaluations

transparency helps evaluations

\[ 2^{128} \]

\[ 2^{64} \]

\[ 2^0 \]

1000 2000 3000 measurements

\[ 2^{128} \]

\[ 2^{64} \]

\[ 2^0 \]

... \[ 2^{80} \]

measurements

bounds

success probability

tighter bounds
• As masking order increases, the # of $d$-tuples of informative samples increases (say by $d$)

$\Rightarrow$ the gap between “simple” attacks targeting one $d$-tuple and $d$ ones increase by a factor $d$
• As masking order increases, the # of $d$-tuples of informative samples increases (say by $d$)

$\Rightarrow$ the gap between “simple” attacks targeting one $d$-tuple and $d$ ones increase by a factor $d$

• If shares are re-used (allowing averaging before combination) this factor becomes $d^d$
Impact for design

- As masking order increases, the # of $d$-tuples of informative samples increases (say by $d$)

$\Rightarrow$ the gap between “simple” attacks targeting one $d$-tuple and $d$ ones increase by a factor $d$

- If shares are re-used (allowing averaging before combination) this factor becomes $d^d$

$\Rightarrow$ It means security depends on efficiency (in cycles), e.g., parallelism reduces # of leaking tuples
• Preliminary questions / definitions
• Side-channel basics (attack steps)
• Noise (aka hardware) is not enough
• Noise amplification (aka masking)
• Reductions help (aka leakage resilience)
• Transparency is needed (open source)
• Mitigating hardware defaults (is hard)
• **Summary and conclusions**
• Effective countermeasures against side-channel attacks always combine sound hardware assumptions & mathematical amplification
• Effective countermeasures against side-channel attacks always combine sound hardware assumptions & mathematical amplification

• High physical security is not mission impossible but has a cost! (e.g., time $\times >10$, area $\times >2$)
  • Yet, good designs can mitigate this cost
Conclusions (I)

• Effective countermeasures against side-channel attacks always combine sound hardware assumptions & mathematical amplification.

• High physical security is not mission impossible but has a cost! (e.g., time $\times 10$, area $\times 2$)
  • Yet, good designs can mitigate this cost.

• Metric I: # of operations per sensitive variable
  • Physical security $\propto$ efficiency

• Metric II: non-linearity (because hard to mask)
• Systematic ways to deal with hardware defaults also has a price (e.g., doubling the # of shares)
• But is probably worth it (to reduce risk)
Conclusions (II)

- Systematic ways to deal with hardware defaults also has a price (e.g., doubling the # of shares)
  - But is probably worth it (to reduce risk)

- Security should not depend on adversaries
  - Beware of too specific evaluations (T-tests)
  - Especially for protected implementations
Conclusions (II)

• Systematic ways to deal with hardware defaults also has a price (e.g., doubling the # of shares)
  • But is probably worth it (to reduce risk)

• Security should not depend on adversaries
  • Beware of too specific evaluations (T-tests)
  • Especially for protected implementations

• Long-term: open source codes/chips that can be used by any (non SCA expert) engineer
THANKS

http://perso.uclouvain.be/fstandae/