Evaluating and Designing Against Side-Channel Leakage: White Box or Black Box?

François-Xavier Standaert
UCLouvain, ICTEAM, Crypto Group (Belgium)
IHMMSec 2021, Virtual
• Introduction to side-channel analysis

• Masking (aka secret sharing) countermeasure

• Leakage evaluation and certification
  • Problem statement & first approach
  • Bounding the Perceived Information

• Conclusions: white box design & evaluation
• Introduction to side-channel analysis

• Masking (aka secret sharing) countermeasure

• Leakage evaluation and certification
  • Problem statement & first approach
  • Bounding the Perceived Information

• Conclusions: white box design & evaluation
Cryptographic algorithms

- e.g. encryption:

- Public algorithms and secret keys
  - Essential for both security and trust
• e.g. encryption:
Cryptographic implementations

• e.g. encryption:

[Diagram showing a flow of data from 'Hello!' through an ENC (encryption) device, then a channel, and finally to another 'Hello!', with additional elements indicating fault analysis and side-channel analysis.]
• ≈ physical attacks that decreases security exponentially in the # of measurements
• ... & where each bit of secret is learned by distinguishing noisy (leakage) distributions

PDF

sample space

$K = 0$

$K = 1$
Standard DPA [KJJ99]

leakage trace

measurement & pre-processing

$\tilde{k}$

subkey candidate

comparison

$\tilde{k}$

$\tilde{m}_i^{k^*}$

model

$S$-box

target intermediate value $V_i$

executed operations

$\chi_i$

$y_i$

$k$

$z_i$
Standard DPA [KJJ99]

measurement & pre-processing

leakage trace

\( l_i \)

\( k \)

subkey candidate

\( m_i^{k^*} \)

prediction & modeling

executed operations

\( x_i \)

\( y_i \)

\( k \)

target intermediate value

\( V_i \)

\( S-box \)

model

\( \tilde{k} \)
Standard DPA [KJJ99]
Standard DPA [KJJ99]

Measurement & pre-processing

Comparison

Subkey candidate

Exploitation

Prediction & modeling

Executed operations

Leakage trace
• General case: profiled DPA [CRR02]
  • Build “templates”, i.e. $\hat{f}(l_i|k, x_i)$
    • e.g. Gaussian, regression-based
  • Maximum likelihood attack
Prediction and modeling

- General case: profiled DPA [CRR02]
- Build “templates”, i.e. $\hat{f}(l_i|k, x_i)$
- e.g. Gaussian, regression-based
- Maximum likelihood attack

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

- Side-channel attacks are continuous
- Better evaluated with information theoretic metrics that capture the attack data complexity

\[ \text{SR} \leq 1 - (1 - \text{MI}(Y; L_Y))^m \]

⇒ # of traces \( m \) to reach \( \text{SR} \approx 1 \propto \frac{c(n)}{\text{MI}(Y; L_Y)} \)
Important attack features

• Side-channel attacks are continuous
  • Better evaluated with information theoretic metrics that capture the attack data complexity

\[
SR \leq 1 - (1 - \text{MI}(Y; L_Y))^m
\]

⇒ # of traces \( m \) to reach \( SR \approx 1 \propto \frac{c(n)}{\text{MI}(Y; L_Y)} \)

• Attacks target two secrets in parallel
  • The block cipher long-term key
  • The leakage model of the implementation

⇒ An optimal attack requires a perfect model
• Introduction to side-channel analysis

• Masking (aka secret sharing) countermeasure

• Leakage evaluation and certification
  • Problem statement & first approach
  • Bounding the Perceived Information

• Conclusions: white box design & evaluation
Noise (hardware) is not enough

\[ Y = 0 \]
\[ Y = 1 \]
Noise (hardware) is not enough

\[ Y = 0 \]
\[ Y = 1 \]
Noise (hardware) is not enough

- Additive noise $\approx \text{cost} \times 2 \Rightarrow \text{security} \times 2$
  $\Rightarrow$ not a good (crypto) security parameter
- $\approx$ same holds for all hardware countermeasures
Example: Boolean encoding

\[ y = y_1 \oplus y_2 \oplus \cdots \oplus y_{d-1} \oplus y_d \]

With \( y_1, y_2, \ldots, y_{d-2}, y_{d-1} \leftarrow \{0,1\}^n \)
• Private circuits / probing security [ISW03]

\[ y = y_1 \oplus y_2 \oplus \cdots \oplus y_{d-1} \oplus y_d \]
Private circuits / probing security [ISW03]

\[ y = y_1 \oplus y_2 \oplus \cdots \oplus y_{d-1} \oplus y_d \]

\[ d - 1 \] probes do not reveal anything on \( y \)
• Private circuits / probing security [ISW03]

\[ y = y_1 \oplus y_2 \oplus \cdots \oplus y_{d-1} \oplus y_d \]

• But $d$ probes completely reveal $y$
Masking (concrete view)

- Private circuits / probing security [ISW03]

\[ y = y_1 \oplus y_2 \oplus \cdots \oplus y_{d-1} \oplus y_d \]

- Noisy leakage security [PR13]
• **Private circuits / probing security [ISW03]**

\[ y = y_1 \oplus y_2 \oplus \cdots \oplus y_{d-1} \oplus y_d \]

Masking (concrete view)

- **Bounded information** \( \text{MI}(Y; L) < \text{MI}(Y_i; L_{Y_i})^d \)
• Private circuits / probing security [ISW03]

\[ y = y_1 \oplus y_2 \oplus \cdots \oplus y_{d-1} \oplus y_d \]

• Bounded information \( \text{MI}(Y; L) < \text{MI}(Y_i; L_{Y_i})^d \)
• Linear operations: $f(a) = f(a_1) \oplus f(a_2) \oplus \cdots \oplus f(a_d)$
• 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
Masked operations [ISW03]

- **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

\[
\begin{bmatrix}
a_1 b_1 & a_1 b_2 & a_1 b_3 \\
a_2 b_1 & a_2 b_2 & a_2 b_3 \\
a_3 b_1 & a_3 b_2 & a_3 b_3 \\
\end{bmatrix}
\]

partial products
• **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
Masked operations [ISW03]

• 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 \quad \text{refreshing} \quad \text{compression}
Masked operations [ISW03]

- **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 \( a_1 b_1 \oplus a_1 b_2 \oplus a_1 b_3 = a_1 b \) leaks on \( b \)
• 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

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

⇒ Quadratic overheads & randomness
  • (Many published optimizations [R+15,Be+16,GM18])
• Leakage mean vector for $Y = 0, 1 = [0.5 \ 0.5]$
• Leakage mean value for $Y = 0, 1 = 1$
Case study: ARM Cortex M4 [JS17]
Case study: ARM Cortex M4 [JS17]

Graph 1: Security vs. SNR
- 31st-order security
- 15th-order security
- 7th-order security
- $2^{128}$-bit security
- $2^{64}$-bit security
- Measured SNR

Graph 2: Performance vs. Number of Shares
- Cycles per byte

10^5
Case study: ARM Cortex M4 [JS17]
• Sounds easy but implementation is complex
• Sounds easy but implementation is complex
  • *Independence issue*: physical defaults (e.g., glitches) can re-combine shares (e.g., [MPG05,NRS11,F+18])
  • Security against horizontal attacks require more noise/randomness as $d$ increases [BCPZ16,CS19]
  • Scalability/composition are challenging [Ba+15,Ba+16]
• Sounds easy but implementation is complex
  • *Independence issue*: physical defaults (e.g., glitches) can re-combine shares (e.g., [MPG05,NRS11,F+18])
  • Security against horizontal attacks require more *noise/randomness* as $d$ increases [BCPZ16,CS19]
  • Scalability/*composition* are challenging [Ba+15,Ba+16]

⇒ High security against DPA can be reached but
  • It implies large performance overheads
    • E.g., industry currently uses 2-4 shares (?)
  • It « only » protects the key (plaintexts are not shared)
Summarizing

- Sounds easy but implementation is complex
  - Independence issue: physical defaults (e.g., glitches) can re-combine shares (e.g., [MPG05,NRS11,F+18])
  - Security against horizontal attacks require more noise/randomness as $d$ increases [BCPZ16,CS19]
  - Scalability/composition are challenging [Ba+15,Ba+16]

⇒ High security against DPA can be reached but
  - It implies large performance overheads
    - E.g., industry currently uses 2-4 shares (?)
    - It « only » protects the key (plaintexts are not shared)

- SPA security expected to be (much) cheaper
Outline

• Introduction to side-channel analysis

• Masking (aka secret sharing) countermeasure

• Leakage evaluation and certification
  • Problem statement & first approach
  • Bounding the Perceived Information

• Conclusions: white box design & evaluation
1. Directly estimate the leakage PDF (or PMF)
2. Try to attack with this estimated model

   - Good if it works (but no guarantees of optimality)
   - Hard to interpret if it does not work:
     - either the leakages are sufficiently noisy, or
     - the model is not accurate (”false sense of security”)
1. Directly estimate the leakage PDF (or PMF)
2. Try to distinguish estimation & assumption errors
1. Directly estimate the leakage PDF (or PMF)
2. Try to distinguish estimation & assumption errors

- Example:
1. Directly estimate the leakage PDF (or PMF)
2. Try to distinguish estimation & assumption errors

- Example:
1. Directly estimate the leakage PDF (or PMF)
2. Try to distinguish estimation & assumption errors

- Example:

  estimation errors dominate

⇒ need to measure more
1. Directly estimate the leakage PDF (or PMF)
2. Try to distinguish estimation & assumption errors

• Example:
1. Directly estimate the leakage PDF (or PMF)
2. Try to distinguish estimation & assumption errors

- Example:

\[ assumption \text{ errors dominate} \]

⇒ need another statistical model
1. Directly estimate the leakage PDF (or PMF)
2. Try to distinguish estimation & assumption errors

• Example:

\[ \text{assumption errors dominate} \]

\[ \Rightarrow \text{need another statistical model} \]

\[ \Rightarrow \text{good enough model: } \text{ass. err} \ll \text{est. err. given } n \]
Information theoretic view [R+11]

\[
\text{PI}(Y_i; L_{Y_i}) = H(Y_i) + \sum_y p(y_i) \cdot \int f(l_{y_i}|y_i) \cdot \log_2 \tilde{m}_n(y_i|l_{y_i}) \, dl
\]

- Information extracted by a statistical model
- Possibly biased by estimation & assumption errors
Information theoretic view [R+11]

\[ \hat{I}(Y_i; L_{Y_i}) = H(Y_i) + \sum_y p(y_i) \cdot \sum_{j=1}^{n_t(y_i)} \frac{1}{n_t(y_i)} \cdot \log_2 \tilde{m}_n(y_i | l_{y_i}) \]

- Information extracted by a statistical model
- Possibly biased by estimation & assumption errors
- Computed in two 2-steps: (1) model estimation (2) integral by sampling (the true distribution)
\[ \hat{\Pi}(Y_i; L_{Y_i}) = H(Y_i) + \sum_y p(y_i) \cdot \sum_{j=1}^{n_t(y_i)} \frac{1}{n_t(y_i)} \cdot \log_2 \tilde{m}_n(y_i|l_{Y_i}) \]

- Information extracted by a statistical model
- Possibly biased by estimation & assumption errors
- Computed in two 2-steps: (1) model estimation (2) integral by sampling (the true distribution)
- \( \Pi = MI \) if the model is perfect (\( \Pi \neq MI \) otherwise)
- E.g., can be negative if the model is too incorrect
\[ \hat{I}(Y_i; L_{Y_i}) = H(Y_i) + \sum_y p(y_i) \cdot \sum_{j=1}^{n_t(y_i)} \frac{1}{n_t(y_i)} \cdot \log_2 \tilde{m}_n(y_i | l_{Y_i}) \]
Information theoretic view [R+11]

\[
\hat{\Pi}(Y_i; L_{Y_i}) = H(Y_i) + \sum_y p(y_i) \cdot \sum_{j=1}^{n_t(y_i)} \frac{1}{n_t(y_i)} \cdot \log_2 \tilde{m}_n(y_i | l_{y_i})
\]

- PI curve “saturates” too far from the MI
- Evaluator has to look for another statistical model

**certification fails**
Concrete limitation #1

We may lack samples to be conclusive
Because estimation errors decrease slowly

\[
\hat{\Pi}(Y_i; L_{Y_i}) = H(Y_i) + \sum_y p(y_i) \cdot \sum_{j=1}^{n_t(y_i)} \frac{1}{n_t(y_i)} \cdot \log_2 \tilde{m}_n(y_i | l_{Y_i})
\]

**certification did not fail (yet?)**
Concrete limitation #2

\[ \hat{\Pi}(Y_i; L_{Y_i}) = H(Y_i) + \sum_y p(y_i) \cdot \sum_{j=1}^{n_t(y_i)} \frac{1}{n_t(y_i)} \cdot \log_2 \tilde{m}_n(y_i | l_{y_i}) \]

- Such certification tests are only qualitative
- They give no indication about the security loss
• Introduction to side-channel analysis

• Masking (aka secret sharing) countermeasure

• Leakage evaluation and certification
  • Problem statement & first approach
  • Bounding the Perceived Information

• Conclusions: white box design & evaluation
Hypothetical Information [DSM16]

\[ HI(Y_i; L_{Y_i}) = H(Y_i) + \sum_y p(y_i) \cdot \sum_l \tilde{m}_n(y_i | l_{y_i}) \cdot \log_2 \tilde{m}_n(y_i | l_{y_i}) \]

- Information that would be extractable from the samples \textit{if} the true distribution was the model
Hypothetical Information [DSM16]  

\[
\text{HI}(Y_i; L_{Y_i}) = H(Y_i) + \sum_y p(y_i) \cdot \sum_l \tilde{m}_n(y_i|l_{y_i}) \cdot \log_2 \tilde{m}_n(y_i|l_{y_i})
\]

- Information that would be extractable from the samples \textit{if} the true distribution was the model

+ Easier/faster to compute (known distribution)
H(HI(Y_i; L_{Y_i}) = H(Y_i) + \sum_{y} p(y_i) \cdot \sum_{l} \tilde{m}_n(y_i | l_{y_i}) \cdot \log_2 \tilde{m}_n(y_i | l_{y_i})

- Information that would be extractable from the samples \textit{if} the true distribution was the model
  + Easier/faster to compute (known distribution)
  - Disconnected from the true distribution
  - Remains positive even if model is incorrect
Hypothetical Information [DSM16]

\[ \text{HI}(Y_i; L_{Y_i}) = H(Y_i) + \sum_{y} p(y_i) \cdot \sum_{l} \tilde{m}_n(y_i | l_{Y_i}) \cdot \log_2 \tilde{m}_n(y_i | l_{Y_i}) \]

- Information that would be extractable from the samples \textit{if} the true distribution was the model
  + Easier/faster to compute (known distribution)
  - Disconnected from the true distribution
    - Remains positive even if model is incorrect
    - Unless specific model families are considered
    - Next: empirical distribution \( \text{eHI}_n(Y_i; L_{Y_i}) \)
• Upper bound for the MI metric

\[
E_{\mathcal{M}} \left( eHI_n(Y_i; L_{Y_i}) \right) \geq \text{MI}(Y_i; L_{Y_i})
\]

\[
\lim_{n \to \infty} eHI_n(Y_i; L_{Y_i}) = \text{MI}(Y_i; L_{Y_i})
\]
Bounds for the Mutual Information

• Upper bound for the MI metric

\[ E_{\mathcal{M}} \left( eHI_n(Y_i; L_{Y_i}) \right) \geq MI(Y_i; L_{Y_i}) \]

\[ \lim_{n \to \infty} eHI_n(Y_i; L_{Y_i}) = MI(Y_i; L_{Y_i}) \]

• Uniform (constant) distribution for the secret \( Y_i \)
  \( \Rightarrow \) MI biased upwards everywhere (like the entropy)

• Monotonic convergence of the empirical distrib.
Bounds for the Mutual Information

• Upper bound for the MI metric

\[ E_M \left( eHI_n(Y_i; L_{Y_i}) \right) \geq \text{MI}(Y_i; L_{Y_i}) \]

\[ \lim_{n \to \infty} eHI_n(Y_i; L_{Y_i}) = \text{MI}(Y_i; L_{Y_i}) \]

• Uniform (constant) distribution for the secret \( Y_i \)
  \( \Rightarrow \) MI biased upwards everywhere (like the entropy)

• Monotonic convergence of the empirical distrib.

• Lower bound for the MI metric

\[ \text{PI}_n(Y_i; L_{Y_i}) \leq \text{MI}(Y_i; L_{Y_i}) \]

• We can only lose information if \( \tilde{m}_n(y_i|l_{y_i}) \neq p(y_i|l_{y_i}) \)
- **eHI converges faster than ePI (no cross-validation)**
• eHI converges faster than ePI (no cross-validation)
• Bound becomes tighter as \( n \) increases
• More eval. efforts lead to better sec. guarantees
• Quite similar results (but unknown MI & lower $n$)
• Quite similar results (but unknown MI & lower $n$)
• Gaussian HI/PI converge (much) faster
  • And are close to the eHI/ePI (in our case study!)
Multivariate analyzes

- Curse of dimensionality ⇒ need assumptions?
- (But then the connection with the MI is lost)
- Nice learning problem: multivariate & higher-order
- Link with statistical learning theory (Vapnik)
Outline

• Introduction to side-channel analysis

• Masking (aka secret sharing) countermeasure

• Leakage evaluation and certification
  • Problem statement & first approach
  • Bounding the Perceived Information

• Conclusions: white box design & evaluation
Evaluation challenge

standard practice

evidence-based evaluations
(assumptions tested per device!)

bounds

success probability

measurements

computation
Evaluation challenge

standard practice

evidence-based evaluations (assumptions tested per device!)

\[ \begin{align*}
\text{measurement bounds} & \geq 2^{30} \\
& = 2^{40}? \\
& = 2^{80}? 
\end{align*} \]
Evaluation challenge

standard practice

open design & evaluation

evidence-based evaluations on reduced versions

proof-based evaluations [DFS15, GS18]
Try leveraging « leveled implementations »
- Strongly protected BC: high-order masking
- Weakly protected permutation: low-latency

Raises many definitional challenges (leakage-resilience)

For such implementations, two different primitives are not an issue (since implementations are different)
• Performance gains of leveled implementations
• Performance gains of leveled implementations
• Block ciphers & symmetric encryption

Transparency (as a measure of maturity)
• Secure cryptographic implementations
THANKS

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