SCAs against Embedded Crypto Devices

F.-X. Standaert

UCL Crypto Group, Université catholique de Louvain

Lecture 3 - Side-Channel Attacks (II)

Outline

- How to evaluate cryptographic implementations?
- IT metric: conditional entropy
- Main theorem (informal)
- Security metric: success rate
- First-order DPA
- Paper & pencil estimations
- Second-order DPA

A motivating example

- Goal: fair evaluation and comparison of two implementations (AES-CMOS and AES-WDDL)
- Tool: adversary A $:= \{$ correlation, $\mathrm{H_W}$, 8-bit target $\}$
  - Key recovered after $q = 10$ traces for AES-CMOS
  - ...and after $q = 10\,000$ traces for AES-WDDL

AES-WDDL 1000 times more "secure" than AES-CMOS?

A motivating example

- Goal: fair evaluation and comparison of two implementations (AES-CMOS and AES-WDDL)
- Tool: adversary A $:= \{$ correlation, $\mathrm{H_W}$, 8-bit target $\}$
  - Key recovered after $q = 10$ traces for AES-CMOS
  - ...and after $q = 10\,000$ traces for AES-WDDL

AES-WDDL 1000 times more “secure” than AES-CMOS?

NO !

Possible issues

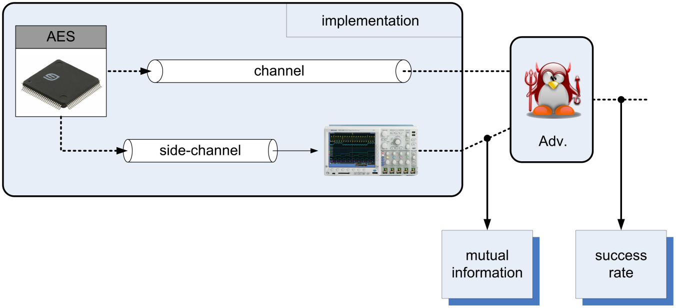
Possible issues

- We may be lucky (only 1 attack performed)
- Distinguisher issue
  - Correlation suboptimal
  - Maybe other distinguishers work better
- Most important: model issue !
  - Hamming weight model suboptimal for CMOS
  - ...and completely meaningless for WDDL

• Consequence: we may perform an evaluation of the adversary rather than a comparison of the implementations

Fair(er) evaluation

- Requires to separate implementations and adversaries



Implementations evaluated with “optimal” profiled attacks

Information theoretic metric

- Conditional entropy and mutual information
  - $\mathrm{MI}(Z;L)$ = information leakage
  - $\mathrm{H}[Z|L]$ = remaining "secrecy" in Z:

$$\mathrm{H}[Z|L] = \mathrm{H}[Z] - \mathrm{MI}(Z;L)$$

- More precisely:

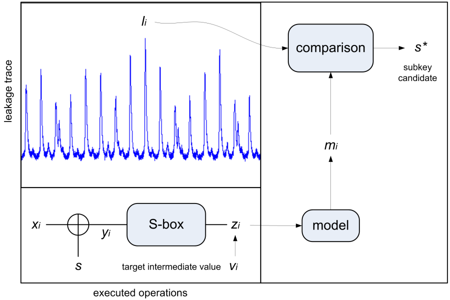
$$\begin{aligned}
\mathrm{H}[Z] &= -\sum_{z\in\mathcal{Z}} \Pr[Z=z]\cdot\log_2 \Pr[Z=z] \\
\mathrm{H}[Z|L] &= -\sum_{l\in\mathcal{L}} \Pr[L=l] \sum_{z\in\mathcal{Z}} \mathrm{H}[Z|L=l] \\
\mathrm{H}[Z|L] &\overset{short}{=} -\sum_{l\in\mathcal{L}} \Pr[l] \sum_{z\in\mathcal{Z}} \mathrm{H}[Z|l]
\end{aligned}$$

Information theoretic metric (II)

$$\begin{aligned}
\mathrm{H}[Z|L] &= -\sum_{l\in\mathcal{L}} \Pr[l] \sum_{z\in\mathcal{Z}} \Pr[z|l].\log_2 \Pr[z|l] \\
&= \{\ldots\} \\
\mathrm{H}[Z|L] &= -\sum_{z\in\mathcal{Z}} \Pr[z] \sum_{l\in\mathcal{L}} \Pr[l|z].\log_2 \Pr[z|l]
\end{aligned}$$

- Second representation closer to actual evaluations (fix one secret, generate all leakages)

Hamming weight example



- Assume $l = \mathrm{HW}(z)$, with z n-bit wide
- Compute $\Pr[Z, L]$, $\Pr[Z]$, $\Pr[L]$, $\Pr[Z|L]$, $\Pr[L|Z]$, $\mathrm{H}[Z|L]$, $\mathrm{I}(Z; L)$, $\{\dots\}$ HW_example_noiseless.m

Noisy Hamming weight example

- Assume $l = \mathrm{HW}(z) + n$ with $n \overset{R}{\leftarrow} \mathcal{N}(0, \sigma_n)$
- Implies using probability density functions:

$$\Pr[l|z] \overset{def}{\equiv} \mathcal{N}(l|\mathrm{HW}(z), \sigma_n)$$

- ... and differential entropies:

$$\mathrm{H}[Z|L] = -\sum_{z\in\mathcal{Z}} \Pr[z] \int_{l\in\mathcal{L}} \Pr[l|z].\log_2 \Pr[z|l]\ dl$$

- HW_example_noise.m, HW_example_noise_fast.m

DPA setting

1. Known plaintext attack scenario:

$$\mathrm{I}(K;X,L) = \mathrm{H}[K] + \sum_{k\in\mathcal{K}} \Pr[k] \sum_{x\in\mathcal{X}} \Pr[x|k] \sum_{l\in\mathcal{L}} \Pr[l|k,x] \cdot \log_2 \Pr[k|x,l]$$

2. X is independent of K:

$$\mathrm{I}(K;X,L) = \mathrm{H}[K] + \sum_{k\in\mathcal{K}} \Pr[k] \sum_{x\in\mathcal{X}} \Pr[x] \sum_{l\in\mathcal{L}} \Pr[l|k,x] \cdot \log_2 \Pr[k|x,l]$$

DPA setting (II)

3. Sampling: adversary's model may be unperfect:

$$\mathrm{PI}(K;X,L) = \mathrm{H}[K] + \sum_{k\in\mathcal{K}} \Pr[k] \sum_{x\in\mathcal{X}} \Pr[x] \sum_{l\in\mathcal{L}} \Pr_{chip}[l|k,x] \cdot \log_2 \Pr_{model}[k|x,l]$$

- i.e. the perceived information can be negative
- $\mathrm{PI}(K;X,L) = \mathrm{I}(K;X,L)$ if $\Pr_{chip} = \Pr_{model}$

4. $\sum_k \sum_x$ is redundant in case of key equivalence
    - It can be sufficient to compute $\mathrm{PI}(K=k;X,L)$
    - sampling_1D.m

Security metric (I)

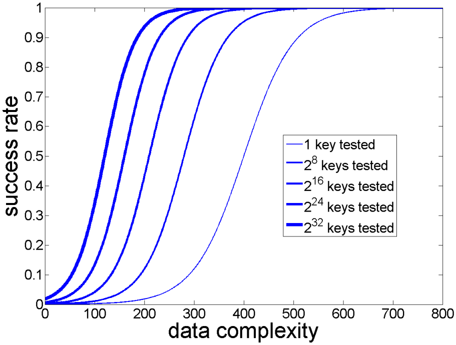
- Perceived information $\approx$ a worst case analysis
- But independent of time complexity (e.g. enumeration)
- $+$ practical adversaries may be suboptimal (e.g. because profiling of the chip is not possible)

- Evaluating how actual distinguishers take advantage of the leakage is the goal of security analysis

- Success rate $= \mathrm{Pr}[\mathrm{Adv}(X, \mathrm{L}(X, k)) = k]$
- (in practice, also estimated from sampling, by launching N_e independent experiments)

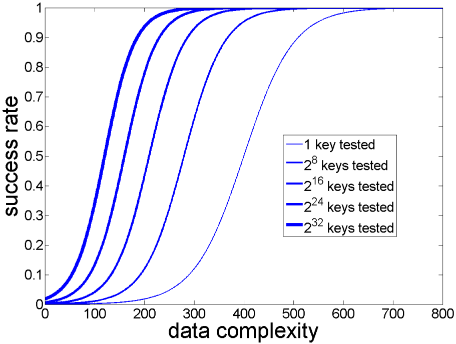
Security metric (II)

- Success rate against a 128-bit master key



Security metric (II)

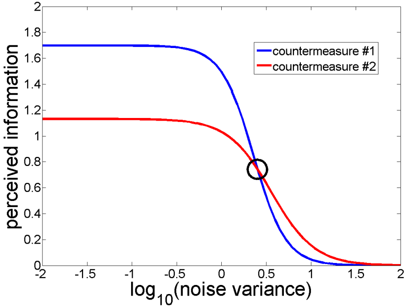
- Success rate against a 128-bit master key



- Optimal enumeration requires probabilities {. . . }

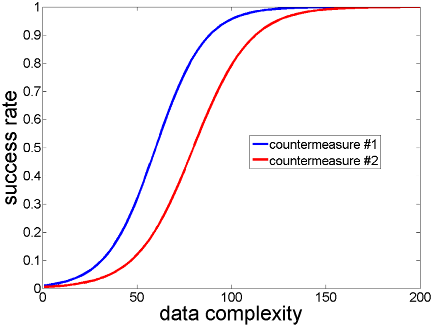
Main theorem (informal)

- $\mathrm{PI}(K; X, L)$ is directly proportional to the success rate of an adversary using $\hat{\Pr}_{\texttt{model}}[k|l]$ as template
- e.g. $\mathrm{PI}(K; X, L)$ in function of the noise variance



As a result

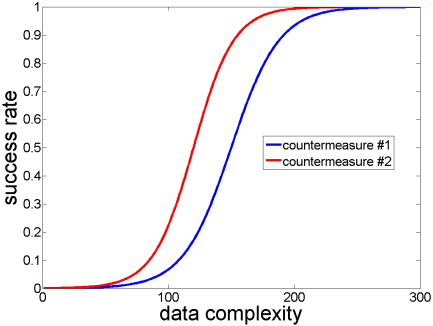
- Left of the intersection



- Countermeasure #2 more secure than first one

As a result

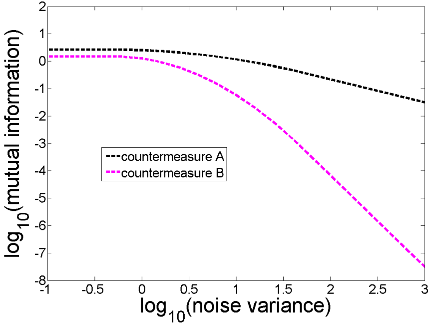
- Right of the intersection



- Countermeasure #1 more secure than first one

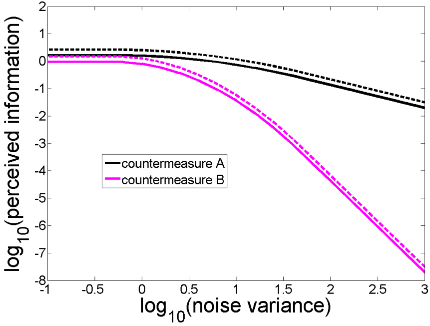
In other words

- $\mathrm{MI}(K; L)$ measures the worst case data complexity

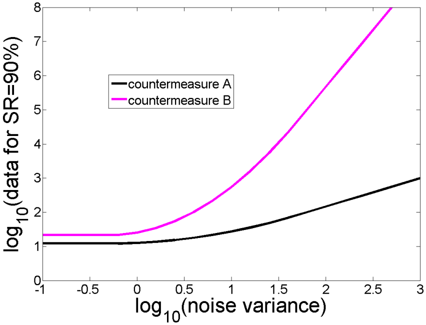


In other words

- $\mathrm{PI}(K; L)$ is the evaluator's best estimate

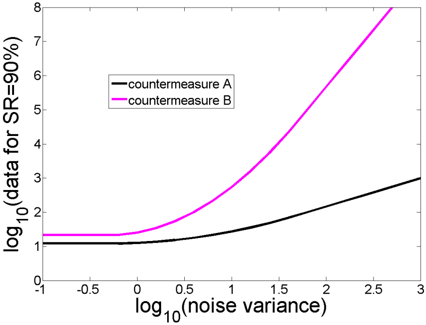


Relation with data complexity



- Theorem only proven in very specific cases

Relation with data complexity



- Theorem only proven in very specific cases
- But holds surprisingly well in all real-world settings

Summary

In theory:

- $\mathrm{H}[K|X, L]$ captures any leakage dependency
- It relates to the asymptotic success rate of the (strongest possible) Bayesian adversary

In practice:

- Computing $\mathrm{H}[K|X, L]$ requires to approximate the leakage pdf $\Pr[K|X, L]$ (not straightforward)
- Multivariate extension $(\mathrm{H}[K|X, L_1, L_2, \ldots, L_d])$ becomes even harder to estimate for large d's
- sampling_2D.m

Summary (II)

- The perceived information depends on:
  - The information leakage provided by the target chip
  - The difficulty to estimate the leakage distributions
- Good security evaluations should rely on the "best available" estimators for the distributions

First-order DPA

Theorem. The mutual information between two normally distributed random variables X, Y, with means μ_X, μ_Y and variances σ_X^2, σ_Y^2 can be expressed as:

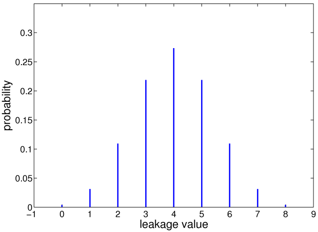
$$\mathrm{I}(X;Y) = -\frac{1}{2} \cdot \log_2 \left(1 - \rho(X,Y)^2\right)$$

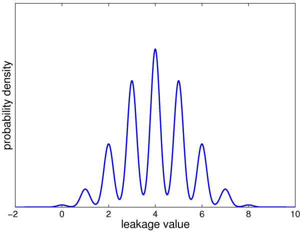
- Previously: template attack $\approx$ correlation attack
- Here: mutual information metric $\approx$ correlation coef.

- Only holds for univariate distributions
- If the same leakage model is used !

First-order DPA (II)

- Are leakage functions Gaussian?





- e.g. for Hamming weights, not exactly
- Approximation better holds for "large enough" noise
- sampling_1D_bis.m

Paper & pencil estimations

Lemma. 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, we have:

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

Lemma. 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 $\sqrt{m/n}$

Paper & pencil estimations (II)

- Assume $\rho(M_k, L)$ follows a normal distribution
- Assume Hamming weight leakage function
- Assume $\rho(M_{k^*}, L) = 0$ for wrong key candidates
- Assume that the number of samples needed to distinguish the key can be approximated with:

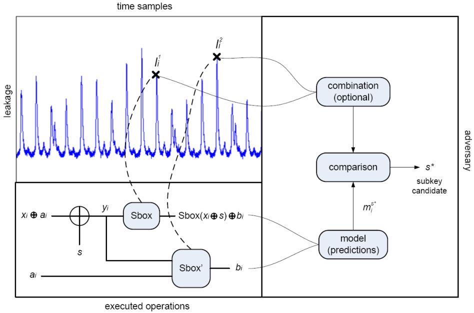
$$n = c \cdot \frac{1}{\rho(M_k, L)^2}$$

Example

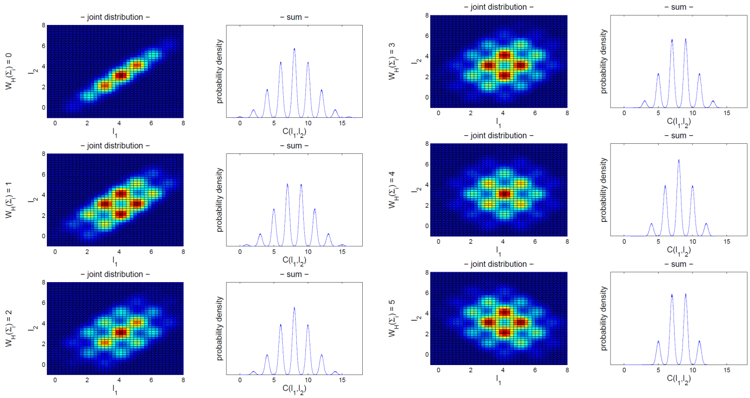
- FPGA implementation of the AES
- 8-bit loop architecture is broken in 10 traces
- How does the complexity of the attack scales?
  - for a 32-bit architecture?
  - for a 128-bit architecture?
- How does it depend on the adversarial capabilities?
- What if the leakage function is not Hamming weight?

Second-order DPA

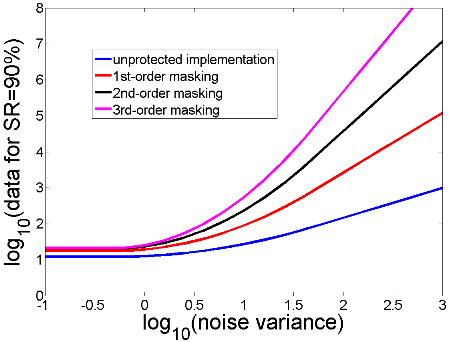
- Against a masked implementation, e.g. with 2 shares



Distribution plots

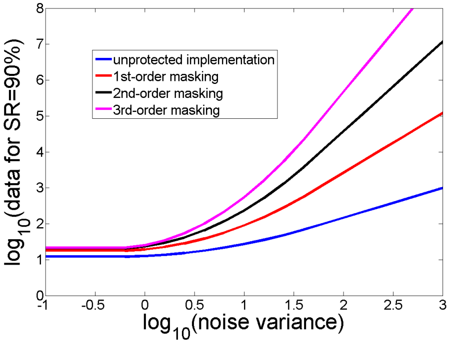


IT analysis



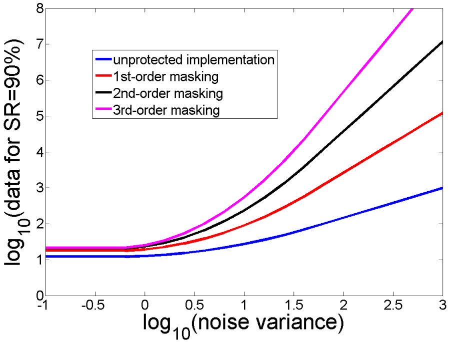
- How does the attacks complexity evolve with N_m?
- $N_{sr=90\%} \approx$

IT analysis



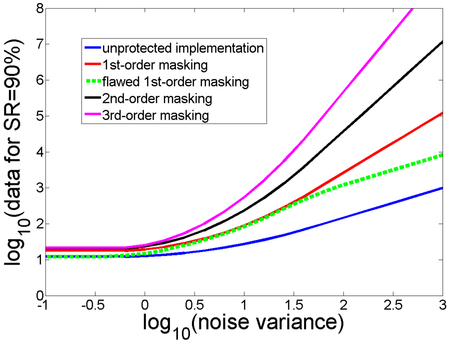
- How does the attacks complexity evolve with N_m?
- $N_{sr=90\%} \approx (\sigma_n^2)$

IT analysis



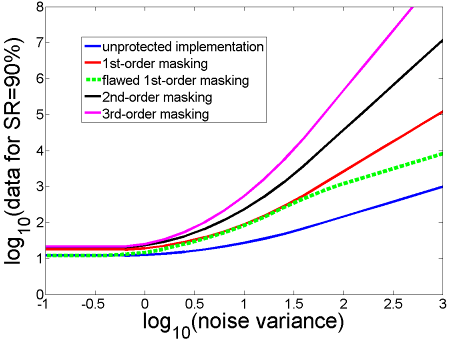
- How does the attacks complexity evolve with N_m?
- $N_{sr=90\%} \approx (\sigma_n^2)^{N_m}$ - Why? $\{\dots\}$

IT analysis (II)



- Flaws due to physical defaults can be detected
  - Examples:

IT analysis (II)



- Flaws due to physical defaults can be detected
  - Examples: glitches, early propagation, ...

Conclusion

- Security evaluations of leaking devices in 2 steps
  - Information theoretic analysis (profiled)
  - Security analysis (profiled or not)
- Usually rely on heuristics
  - Because of practical limitations
  - e.g. estimating an d-dimensional distribution can be hard (i.e. require too many measurements)

Conclusion (II)

- There are "easy" contexts
  - e.g. univariate SCAs with additive Gaussian noise
- Protected implementations are harder to analyze
  - e.g. masking implies "mixture" distributions
- Cryptographer's goal: design efficient algorithms and implementations with bounded information leakage

Further readings

- S. Mangard, E. Oswald, T. Popp, *Power Analysis Attacks (DPA book)*, Springer, 2007
- Recent results on side-channel attacks can be found in the proceedings of the CHES conference: http://www.sigmod.org/dblp/db/conf/ches/index.html
- *e.g.* correlation attacks, template attacks, collision attacks, masking schemes, higher-order attacks ...

Thanks