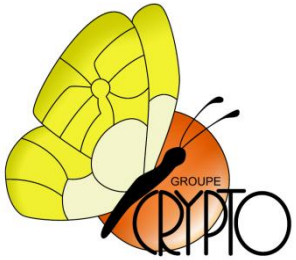
Side channel attacks evaluation : Template VS Machine Learning

Romain Poussier, François-Xavier Standaert: Université catholique de Louvain
Liran Lerman: Université libre de Bruxelles



Outline

- Introduction to SCA
- Standard tools
- Machine learning
- General Intuition
- Framework and observations
- Future work

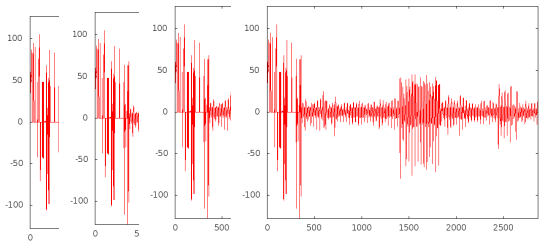
1. Overall Functioning

Target device

Consumption model

Power consumption traces

Prior hypothesis	Model estimation : profiling
Hamming weight Single bit	*Gaussian template Linear regression*



Statistical distinguisher

$K_j \neq K$

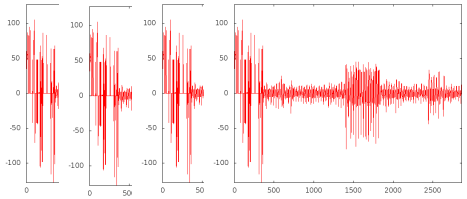
$K_j = K$

1. Profiling: worst case scenario

Device copy of the attacker

Different messages

Different keys



Profile for all target values (e.g. M ⊕ K)

- Motivation :
  - worst case scenario
  - Kerckhoffs's principle

2. Template attack

$$f_{\mu,\Sigma}(x) = \frac{1}{(2\pi)^{N/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)}$$

- Profiling: Two parameters to estimate
  - For each M $\oplus$ K value, the associated mean μ
  - An overall noise covariance matrix Σ
- Attack:
  - For an attack trace l, for each key candidate k_i, compute:

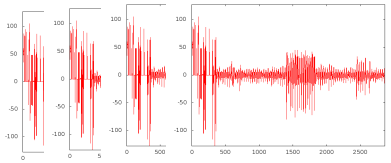
$$P(K_i|\, l) = \frac{P(l \mid K_i)\, P(K_i)}{P(l)} = \frac{f_{\mu_i,\Sigma}(l)}{\Sigma_j f_{\mu_j,\Sigma}(l)}$$

2. Linear regression

- Assume that each physical observables come from a deterministic function H and random noise R:

$$I(x,k) = H(x,k) + R$$

- Given the basis $(g_1, g_2, \dots, g_n)$ and some observations, find the best a_i which approximate H:



Traces from different known inputs x, k

$$h(x,k) = a_0 + \sum_{i=1}^{n} g_1 a_1$$

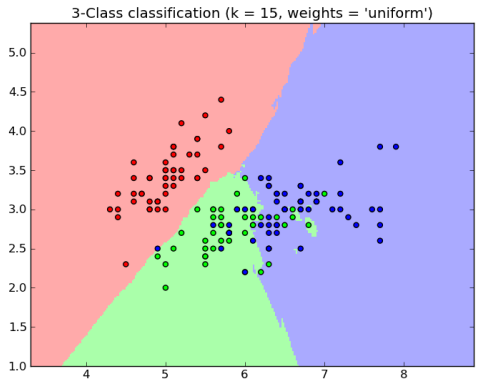
Example

- g_i represent the bits of $x \oplus k$ (LR9)
- All bit combination (LR256)

3. Machine learning

Supervised

- Two steps
  - Learning with labeled data
  - Classification

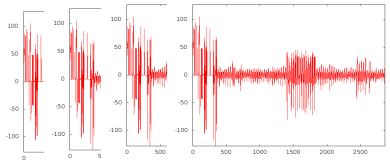


- Examples :
  - SVM, Random Forest

Unsupervised

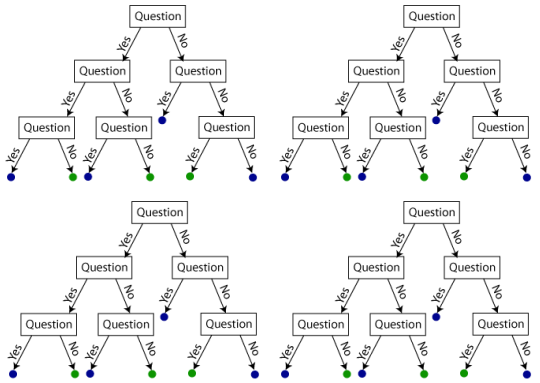
- One step
- Often probabilistic
- Weaker than supervised
- No need of labeled data
- Examples :
  - Kmeans
  - Kmedoids

3. Random forest

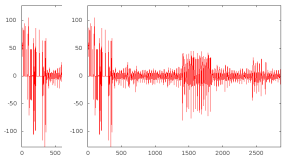


Traces from different known inputs x, k

Creates different decision trees with random subset of traces



Asks for classification



Attack traces with unknown key K

Collects the number of votes for each tree

Recover K

4. General intuition

Device with a real leakage function L

Simple L

- L is explained by a low number of explicative variables:

LR9 is the most powerful tool

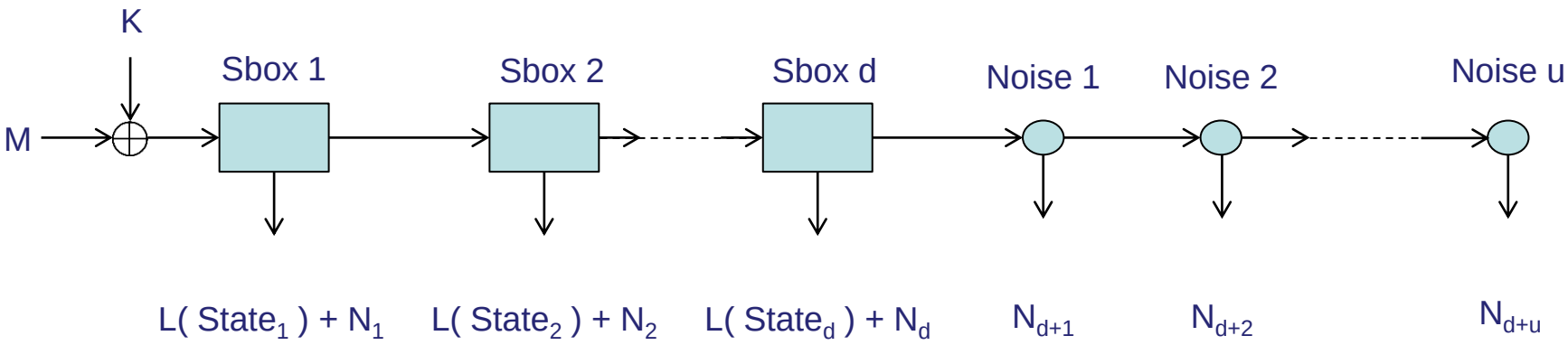
Complex L

- High number of explicative variables
- Black box

TA (≃LR256) is needed

5. Framework

Leakage simulation:

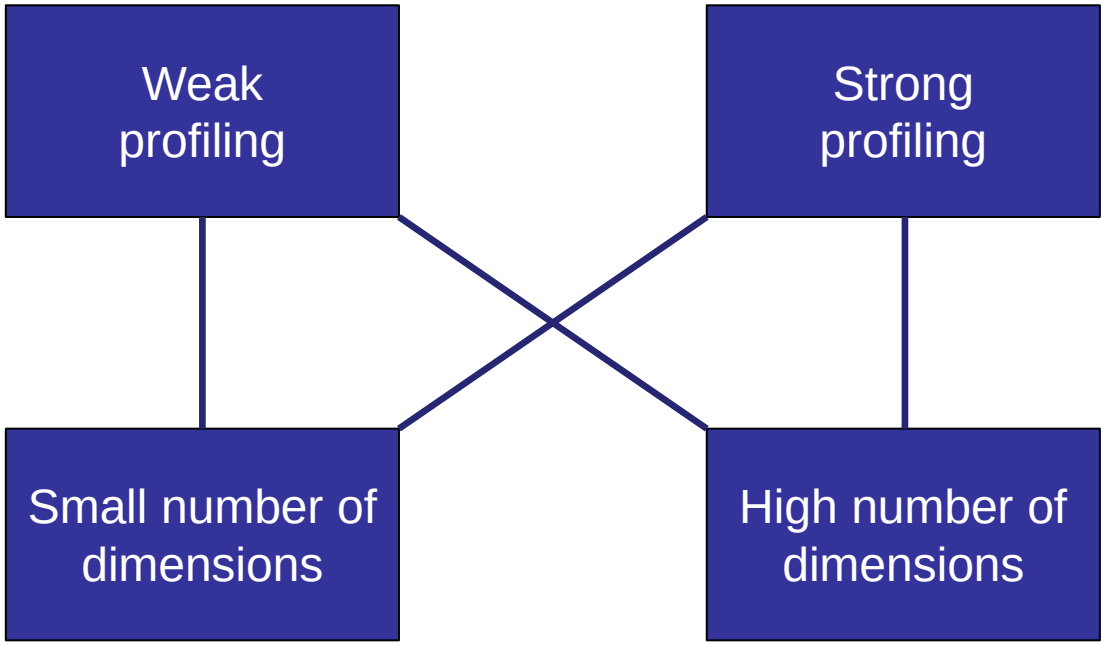


L : Linear leakage function (normalized)

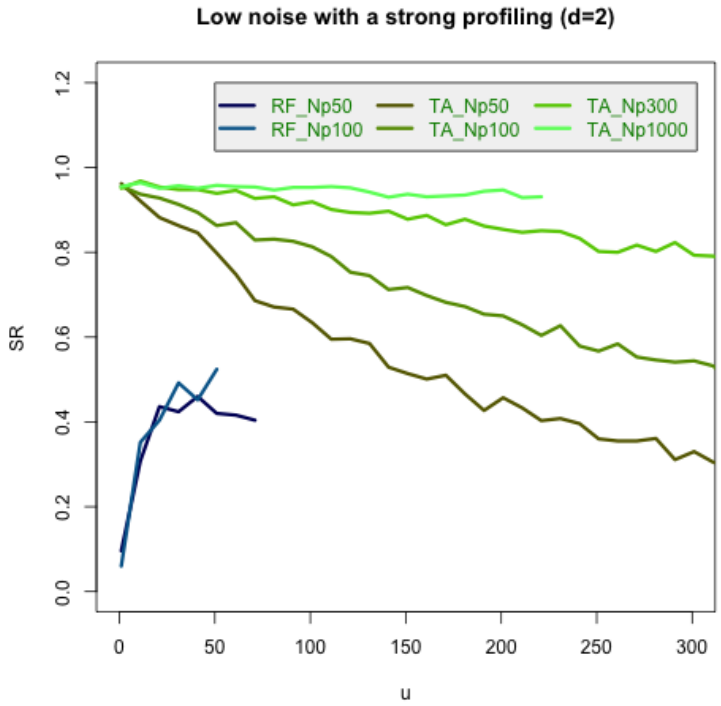
N_i : Random gaussian variable at time i
(variance σ^2, mean 0)

5. Framework

Template versus Random Forest



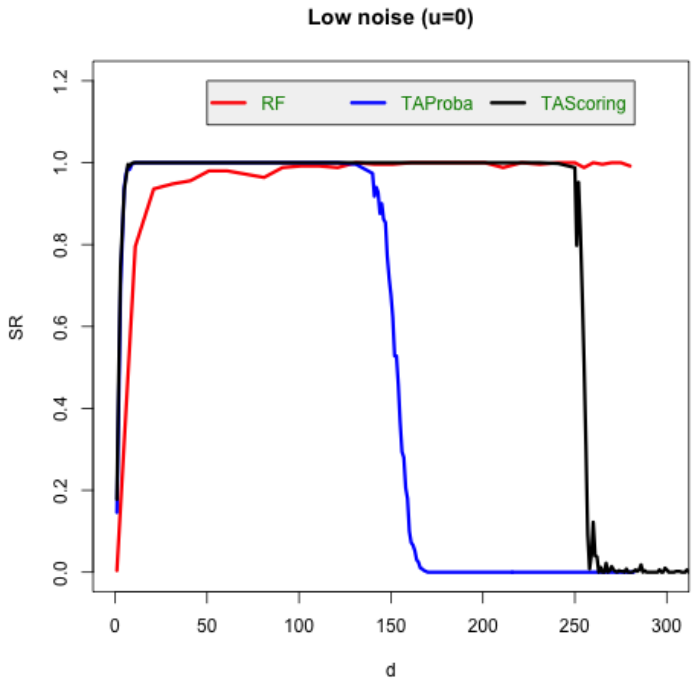
3. Strong profiling

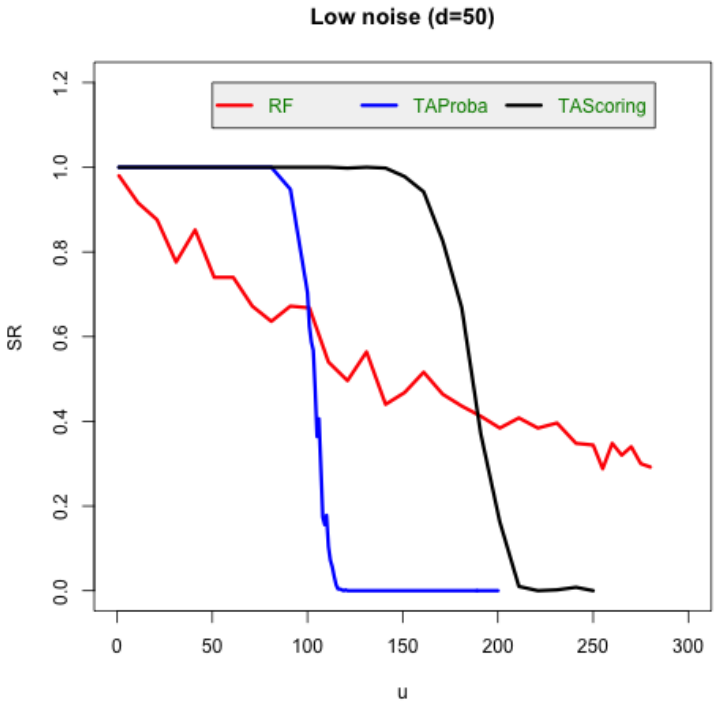


d = 2, u on x axis

50 attack traces, $\sigma^2 = 1$.

3. Low profiling



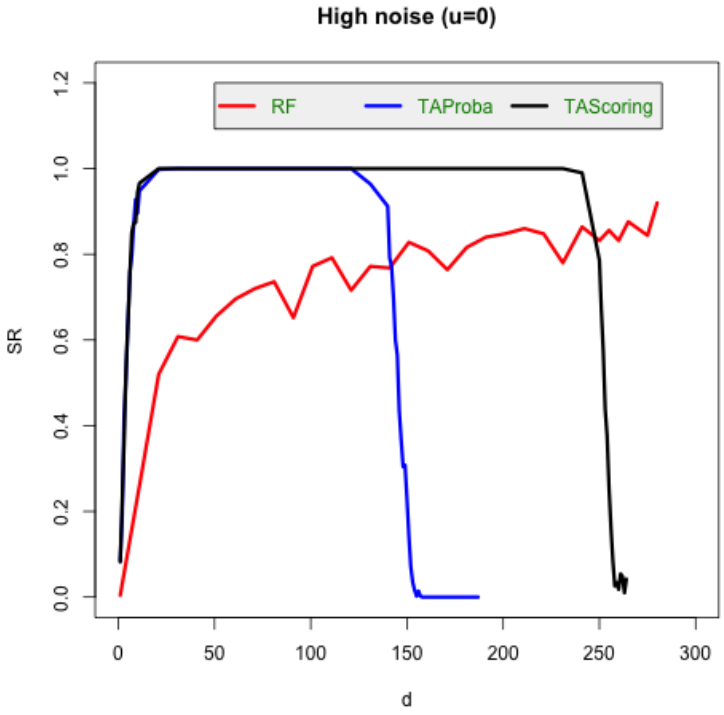


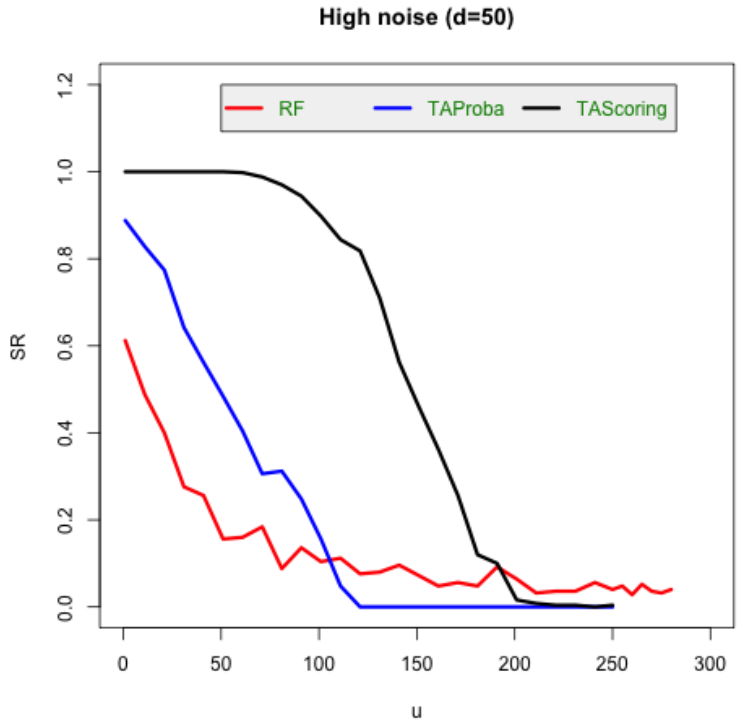
Left : u = 0, d on x axis

Right : d = 50, u on x axis

10 attack traces, 2 profiling traces per value, $\sigma^2 = 1$.

3. Low profiling





Left : u = 0, d on x axis

Right : d = 50, u on x axis

50 attack traces, 2 profiling traces per value, $\sigma^2 = 1$.

6. Future work

- Extend the analysis to masking
  - Profiling needs more traces
- Give a fair comparison in terms of compexity (time/memory)
- See how the use of scoring impacts the master key enumeration

Thank you for your attention

Any questions ?