A machine learning approach against a masked AES

L. LERMAN, S. FERNANDES MEDEIROS, G. BONTEMPI, and O. MARKOWITCH

Université Libre de Bruxelles
Faculty of Sciences
Department of Computer Sciences
Cryptography and Security Service & Machine Learning Group

CARDIS 2013

Context

Cryptography has been used for a long time for confidentiality purposes

- Mobile phones
- Banks
- Cars



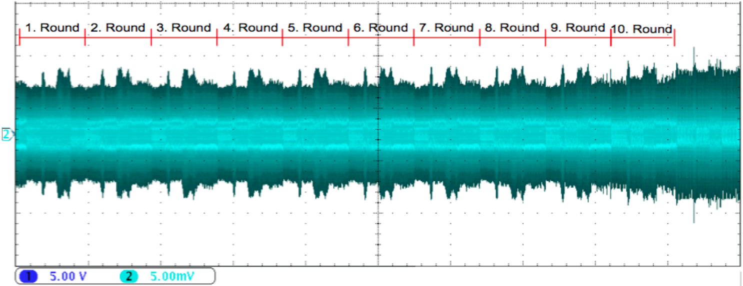
Side channel attacks

Reduction in cryptography security in real situation
Possibility to find the secret key when we focalize on a side channel

- Timing attack (Kocher - 1996)
- Electromagnetic attack (Gandolfi, Mourtel & Olivier - 2001)
- Power monitoring attack (Kocher, Jaffe & Jun - 1999)

Power monitoring attack

EM leakage



[1]

$$T^{(Q)} = \left\{ T^{(Q)}_{(t)} \in \mathbb{R} \middle| t \in [1; n] \right\}$$

[1] MARTINASEK, Z., ZEMAN, V., TRASY, K.. Simple Electromagnetic Analysis in Cryptography. International Journal of Advances in Telecommunications, Electrotechnics, Signals and Systems, North America, 1, sep. 2012.

Non-profiling attacks

- f is the target function (e.g. SBox) using P and Q
- L is the leakage model (e.g. HW)
- D is the distinguisher (e.g. Pearson correlation)

$$\hat{Q} = \arg\max_{Q \in \mathcal{Q}} \left| D\left(L\left(f\left(P, Q\right)\right), T\right) \right|$$

Profiling attacks

$$\hat{Q} = \arg\max_{Q} P(Q|T)$$

$$\hat{Q} = \arg\max_{Q} \frac{P(T|Q) \times P(Q)}{P(T)}$$

$$\hat{Q} = \arg\max_{Q} \hat{P}(T|Q) \times \hat{P}(Q)$$

How to estimate $P(T|Q)$?

Profiling attacks

- Parametric methods
  - TA (i.e. $P(T|Q_i) \sim N(\mu_i, \Sigma_i)$) **[S. Chari et al. 2002]**
  - SA (i.e. $P(T|Q_i) \sim N(\mu_i, \Sigma)$) **[W. Schindler et al. 2005]**
- Non-parametric methods **[L. Lerman et al. 2011 & 2013, G. Hospodar et al. 2011, A. Heuser et al. 2012, T. Bartkewitz et al. 2012]**
  - SVM
  - RF
  - KNN
- Results in unprotected contexts
  - A ML model is as efficient (and often better) than TA

Countermeasures

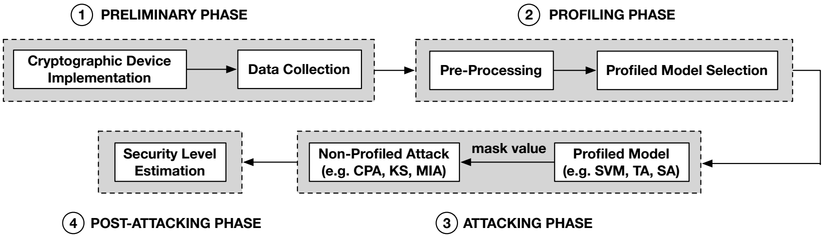
- Several countermeasures
  - Masking
  - Hiding
- Several algorithms of masking schemes
  - Boolean, multiplicative, affine masking schemes

Issues

Are the results of the previous ML works still the same in a protected environment?

1. How many traces are required
   1. against a protected device with a ML model compared to a strategy based on TA or SA?
   2. by a ML model attacking a protected device compared to an unprotected device?
2. What is the impact of the number of traces used in the profiling step by a ML model attacking a protected device?

Framework



Lower the error between the correct and the estimated mask values, higher the correlation between the real and the predicted traces for the correct key

Target

- AES-128 protected by the Rotating Sbox (Boolean) Masking scheme (based on table look-up)
- Atmel ATMega-163 smart card
- According to its authors (in a hardware context):
  - Performances and complexity close to unprotected scheme
  - Resistant against several side-channel attacks

Models

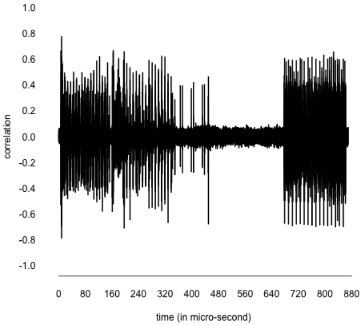
- Profiling attacks
  - TA
  - SA
  - SVM
  - RF
- Non-profiling attack
  - CPA on $\mathrm{HW}(\mathrm{maskedSBox}(\mathrm{plaintext} \oplus \mathrm{mask} \oplus \mathrm{key}))$

Dataset

- Public dataset of the DPAContest V4 (updated in October)
- Electromagnetic emission leakages
- First round of AES
- Each trace has 435,002 samples

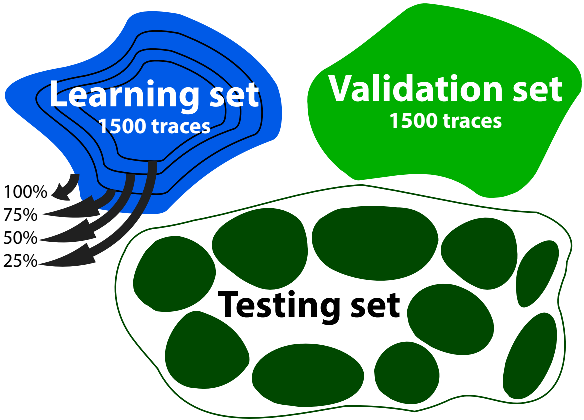
Finding the offset value on traces

$\rho({}_t T, \text{offset})$ on 1500 traces



Feature selection step:
50 instants highest linearly correlated with the offset value

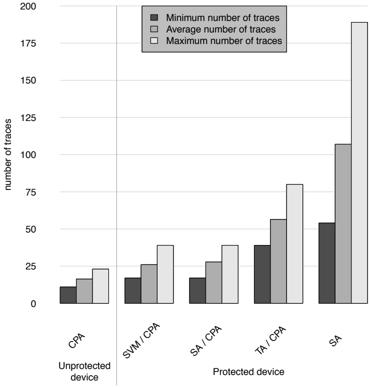
Model estimation



Model selection results

- Higher the number of traces in the learning set, higher the accuracy
- Higher the number of features, higher the success rates for SVM, RF and SA (except TA)
- The success rates of
  - ML models
    - SVM: 0.88
    - RF: 0.81
  - SA: 0.90
  - TA: 0.66

Attacking step



- Unmasked implementation
  - CPA: 16.3 traces in average (5s)
- Masked implementation
  - SVM / CPA: 26 traces in average (20s)
  - SA / CPA: 27.8 traces in average (80s)
  - TA / CPA: 56.4 traces in average (45s)
  - SA: 107 traces in average (180s)

Discussion & Conclusion

- (Unprotected) implementation of the Rotating Sbox Masking
  - 26 traces with 20s during the attacking phase
- ML approach outperforms TA in data complexity
- Original SA is less efficient than the new strategy based on SA
- SVM outperforms SA in time complexity
- How to improve the attack ?
  - Increasing the number of points selected in each trace
  - Optimizing the model's parameters

Last but not least ...

Official result in the DPAContest V4 :

22 traces with 0.528 seconds

in order to retrieve the secret key of AES-128