THALES

Deep Learning versus Templates Attacks : experimental comparison

https://eprint.iacr.org/2018/1213

Francis OLIVIER, Eric BOURBAO, Yevhenii ZOTKIN

THALES-DIS *(formerly GEMALTO)*

CryptArchi 2019 (Pruhonice)

Preamble: origin of this study

Excerpt from ANSSI + CEA-LETI 2018 white paper

- Deep Learning Techniques for SCA and introduction to ASCAD data base. Benadjila, Prouff…

3.3.3 Comparison with Template Attacks

We compare MLP_{best} with standard Template Attacks (aka Quadratic Discriminant Analysis, or QDA in the Machine Learning community). We first perform an unsupervised dimension reduction to extract meaningful features. For this task we use a classical PCA which is parametrized by the number of components to extract. Then the classification task is performed with a QDA (*i.e.* Template Attacks). Note that, contrary to QDA, neural networks do not require the preprocessing feature extraction step since this task is realized by the first layers of the networks. Figure 15 shows the results obtained with different numbers of components extracted from the PCA.

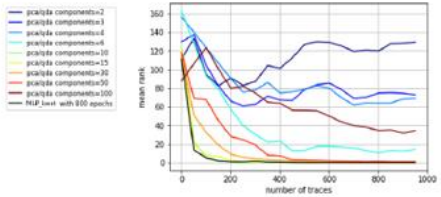


Figure 15: Mean ranks of a PCA on n components followed by a QDA.

- ASCAD data base : set of traces on a basically masked AES 1st round (mask included)
- This implies that Template Attacks (QDA) work on masked implementations ! **THAT'S A SCOOP !**

Agenda

DL versus TA in 4 use cases

- Quick reminder on TA, DL, Multi Layer Perceptron, Convolutional Neural Network
- Use cases A, B, C, D
- **Static targets/dynamic targets: cross matching**

DL versus TA results (comparison on ranking convergence criterion)

- Use cases A and B (dynamic)
- Use case C (static, 8 bit)
- Use case D (static, 8 bit and 16 bit words)
- Back to use cases A and B seen as static

Take away : 4 lessons

Conclusion : risks and protections

Quick reminder : Side-channel analysis as Machine Learning

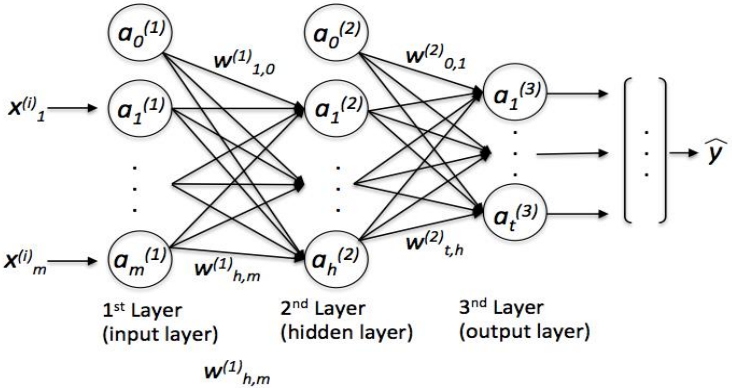
Machine Learning encompasses many supervised classification attacks involving

1. A first profiling/learning step on an open device
2. A second testing/matching step on a locked device hosting the secret K*

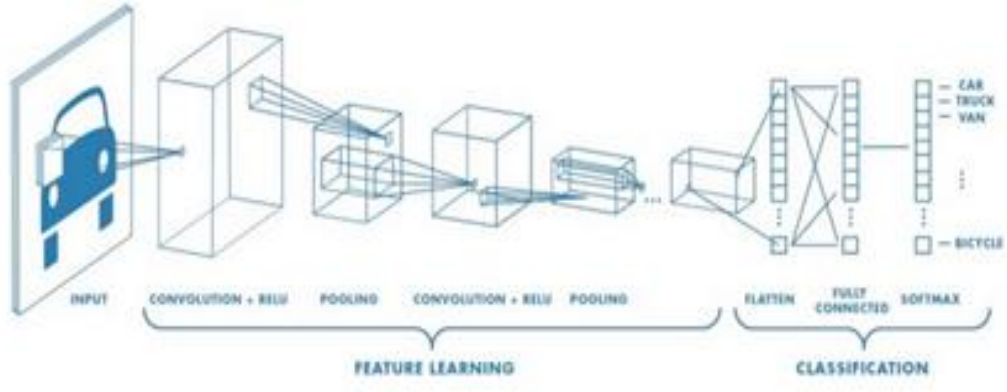
- Template based key inference (Bayes, Gauss): $$p(k/\{C_i\}) = \prod_i p(k/C_i) = \frac{1}{\sqrt{(2\pi)^n |R_k|}} e^{-\frac{1}{2}\sum_i (C_i - T_k)^t R_k^{-1} (C_i - T_k)}$$

- Deep Learning does the same with **Neural Networks** such as:

Multi Layer Perceptron (MLP)



Convolutional Neural Network (CNN)



THALES

Experimental comparative study on 4 use cases

	Target	Leakage	Traces (profile)	Traces (test)
A	Masked SubByte	High (EM)	45,000	5,000
B	Masked SubByte	Low (EM)	900,000	100,000
C	8 bits transfer	High (power)	900,000	100,000
D	8, 16, 32 bits transfer	Low (EM)	500,000	500,000

- A: Masked AES substitution ASCAD (high leakage)
- B: Masked AES substitution on a realistic chip (low leakage EM)

$$SBox'(K_i \oplus M_i \oplus u) = SBox(K_i \oplus M_i) \oplus v$$

- C: 8 bit data transfer (old chip with strong leakage)
- D: 8, 16, 32 bit transfer (recent « smart card » chip with low leakage)

❖ *Remark: all signals are perfectly aligned or desynchronized on purpose*

Static target / dynamic target (test phase)

❖ ***Warning : pattern recognition of PK cryptography calculation units not considered here !***

Static targets handle a constant secret word K_i^*(cases C&D)

- Secret transfer, key loading, key schedule, PIN verification…
- Unvarying data in test phase (no message)
- Possibly masked (e.g key schedule)

Dynamic targets involve a varying message (encryption cases A&B)!

- Learning/profiling based upon the leakages of

$$K_i \oplus M_i \text{ (input) and } \mathrm{SBox}(K_i \oplus M_i) \text{ (output)}$$

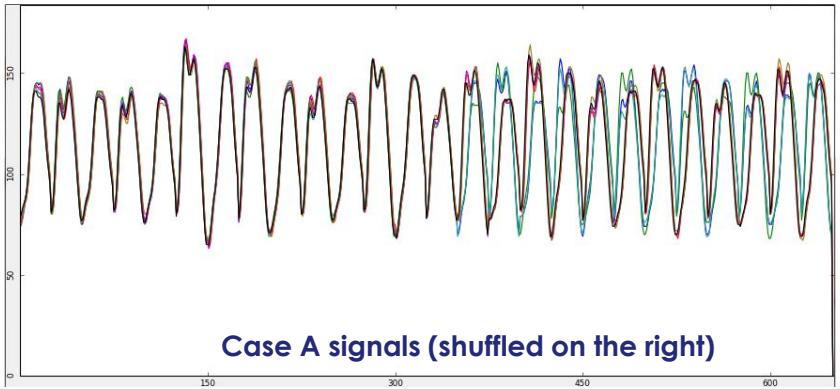
- **Cross matching** : classes index redirection with message *M** (test phase)

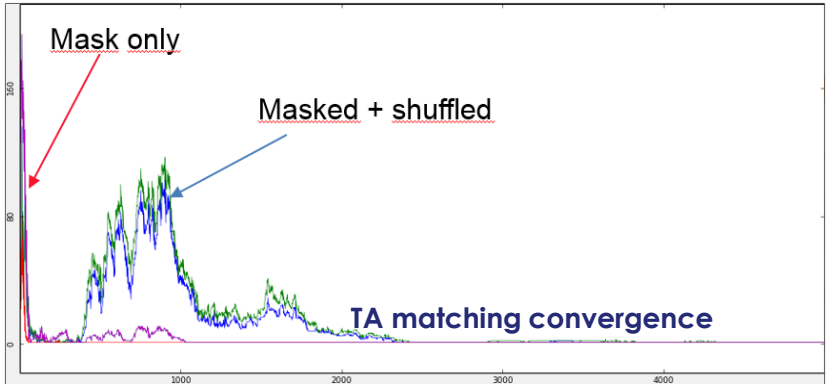
$$K^* = M^* \oplus K \oplus M$$

Matching performance comparison criteria

Attacks (test phase) are played in white box for all possible key words

- The expected value is **ranked** 1-256 amongst all hypotheses according to the distinguisher score (max likelihood)
  - Sum then rank: **aggregation** (noise reduction)
  - Rank then sum: **single trace** matching
  - Average rank over ***all values***: residual entropy = $\log_2(E[Rank])$
- **Convergence** : number of challenge traces to be aggregated for rank minimization (hopefully rank 1)





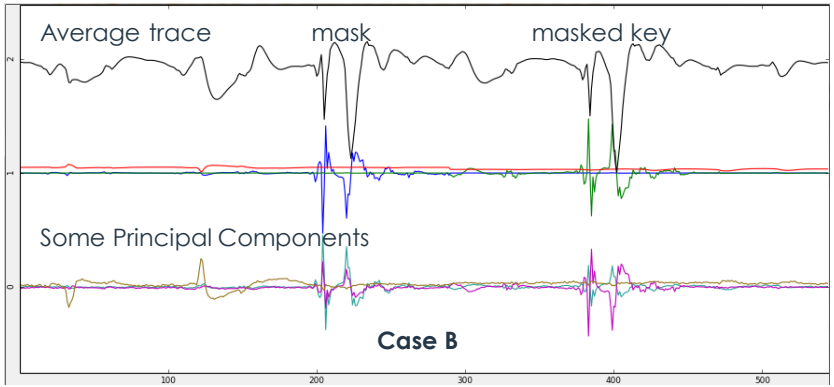
Many optional optimizations blur the picture

Optimization options

- MLP/CNN DL notions:
  - Hyperparameters, loss function, gradient, epochs, batch, accuracy, overfitting…
  - Learning involves many algorithmic tricks : regularization, early stopping…
- Normalization $x \rightarrow \frac{x-T}{\sigma}$
- Outliers discarding (rejection of « abnormal » candidates)
- TA: Principal Components Analysis (Projections onto covariance Eigen vectors subspace POIs)

Issues

- Optimality/robustness trade-off
- Loss of genericity
  - Lot of parameters to be set
  - Work during attack calibration
  - Portability on a different locked device?
- Neural networks are black boxes

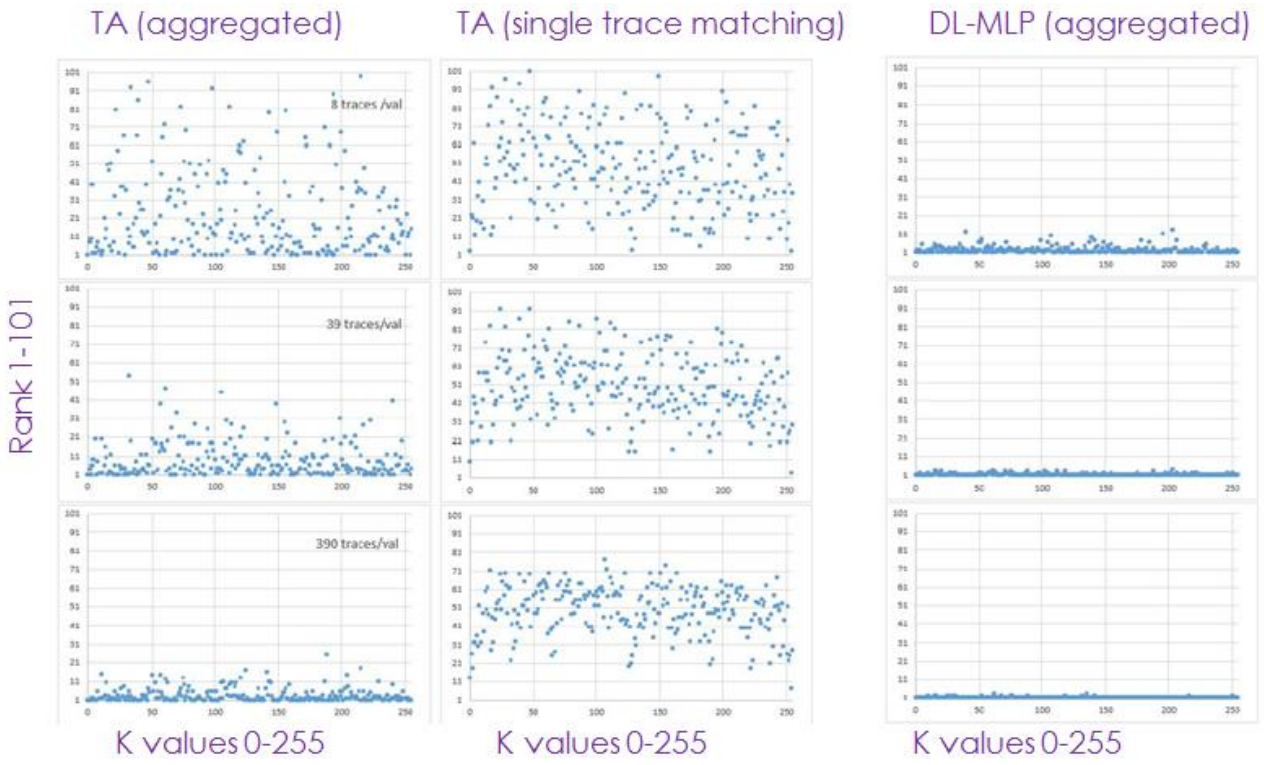


Targets A & B : DL ranks faster than TA ; CNN resist desynchronization

Results in number of aggregated challenges to reach rank 1

Use case	DL	DL optimized	TA	TA optimized
A (aligned)	42 (MLP)		50	45
A (slightly shuffled)	250 (CNN)	46 (MLP!)	2000	200
B (aligned)	150 (CNN)		4500	1000
B (slightly desynchronized)	150 (CNN)	150 (CNN)	Fail	Fail

Static target C : MLP ranks faster that TA



Nb of traces /val	Avg rank (TA)	Avg rank (MLP)
8 (2000)	24	2.16
39 10000	8.9	1.24
390 100000	3.5	1.07

THALES

Static target D (8 bit) : tough case for both TA and MLP

- 500.000 traces for profiling (TA) and learning (MLP)
- TA

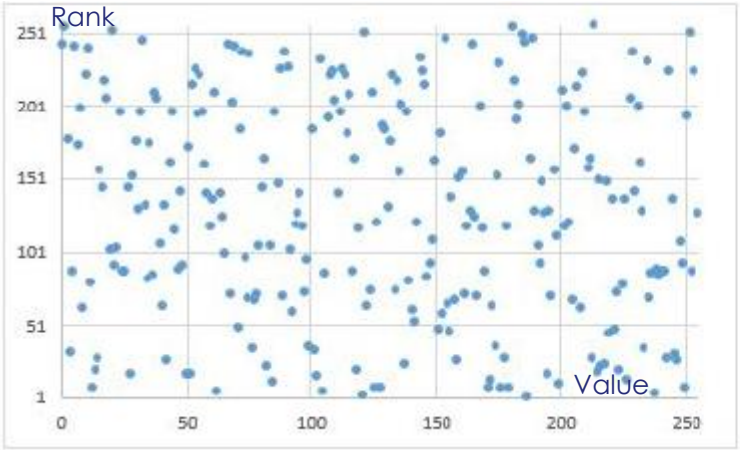
Nb of traces	Nb of traces / val	Avg aggregated rank
1000	4	92
10000	39	52
50000	195	33
100000	390	28
500000	1950	26

- MLP performs barely better
  - Average rank = 28
  - After 50.000 traces aggregation (*TA needs twice*)
- **Both fail to reach rank 1 !!!**

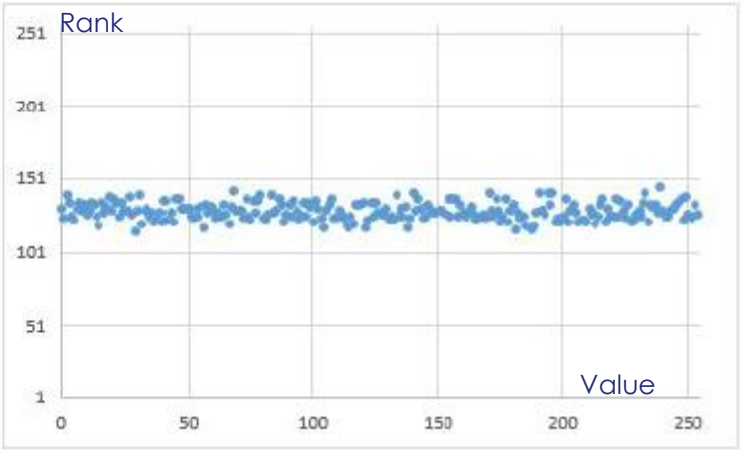
Static target D (16 bit case) : both TA and MLP fail !

Target D in the 16 bit case… (partitioning still on 8 bit)

- Profiling/learning still over 8 bits while 16 bits are handled
- TA matches « at random » and **FAILS** !



Aggregating 1500 candidates/value



1500 « single trace matching » per value

- DL-MLP **FAILS** the same !

THALES

Back to targets A and B seen as “masked static targets”: a failure

… e.g. as in a key schedule

Leakage observation	Type of problem	Resistance
$\mathcal{L}(SBox[K \oplus M])$	Encryption	Very low
$\mathcal{L}(u), \mathcal{L}(SBox[K \oplus M] \oplus u)$	Masked encryption	Low
$\mathcal{L}(K)$	Secrecy transfer	Possibly high
$\mathcal{L}(u), \mathcal{L}(K)$	Masked key schedule	Very high

Table 7: *Typical targets of supervised attacks and their intrinsic resistance.*

- Considering « K⊕M » as the masked secret in cases A and B
- Same results as masked AES SBox output without cross-matching
- Results on 8 bit : **all methods fail !**
  - Average rank converged around 55 (≈1 bit of information)
  - Not far from « matching at random »

Lesson #1: Supervised attacks can defeat masking

Observed on dynamic targets only !

- Provided the Points Of Interest are given ! (mask and masked variable)
- TA exploits non diagonal elements in the classes covariance R_k
  - Contain the interdependences between the mask and the masked value
  - Extracted by PCA
  - Optionnally enhanced by normalization
  - Global « pooled » covariance cannot and does not work
- Confirmed with DL methods even better:
  - How does DL proceed to retrieve the « mask/masked » interdependences ?
  - Open question !

Lesson #2 : Comparisons DL vs TA, MLP vs CNN

Deep Learning performs better than Template Attack…

- With less candidates aggregation
- With more calculation

… but TA performs not so bad!

- Even against masking! (**the scoop**!)
- With aggregation (too)
- Fails against traces misalignment

Target	DL- MLP	DL-CNN	Opt TA
Masked encryption (aligned)	+++	++	++
Masked encryption (slightly shuffled)	+	++	-
Masked encryption (slightly misaligned)	-	++	-
Data transfers (8 bit)	0+	N.A	0-
Data transfers (16 bit)	- -	- -	- -

MLP is more effective than CNN against aligned targets *(or fails like TA !)*

But CNN can still perform against slight misalignment (time invariance)

- Very large neural networks needed with more computational ressources
- CNN fails as misalignment increases

Lesson #3: Beware of toy problems !

« Good leaking targets » make « good looking attacks »

- Cross matching strongly helps the ranking
- Successfull attacks found in literature require
  - Strong leakage (low noise)
  - And **cross matching** (encryption)
  - And aggregation (of candidate traces)
- **BTW: single trace matching is an old fantasy !**
- But is encryption a relevant target?
- State of the art cryptographic libraries
  - Are more complex
  - Implement tougher countermeasures
  - Are protected at least against 2nd order DPA, CPA, MIA…

Lesson #4: Static targets make more sense and resist quite well

- Difficulties encountered by the supervised attacks against static targets
  - Low leakage (weak SNR)
  - Some values work better than others: **all should be tested**!
  - **Surjective** mapping from data set to signal space (e.g. Hamming weight model)
  - No variability ⇒ no **cross matching** to solve this surjectivity limit and distinguish hypotheses

- Large words ≥16 bit raise many problems
  - Many classes (>65536)
  - Statistical estimations issues (TA)
  - Hard to test all values many times each
  - BTW Schindler's stochastic approach (CHES'05)
    - Templates model $T_k(t) = C_0(t) + \sum_i C_i(t) b_{ki}$ *(multilinear bit decomposition of k: b_{ki})*
    - No significant examples in literature (to our best knowledge)
    - Our 16 bit experience (on target D) : hard to rank a short word below 1000 (on 65536!)
    - Only one global covariance matrix : hard limitation (against masking)
    - What about DL : new formalization, new networks?

	Byte 1	Byte 2	Byte 3	Byte 4
Learning	SEL	RND	RND	RND
	Enhanced by aggregation	Generate data variance only		
Testing	Fixed Secret	Fixed Secret	Fixed Secret	Fixed Secret
	Aggregation is powerless	No data variability : generates only **BIAS** !		

THALES

Conclusion: Status on risks and protections

DL techniques improve supervised attacks without major breakthrough

- But we had not seen it all about TA (masking)!
- CNN is definitely the new threat against misalignement (and masking)

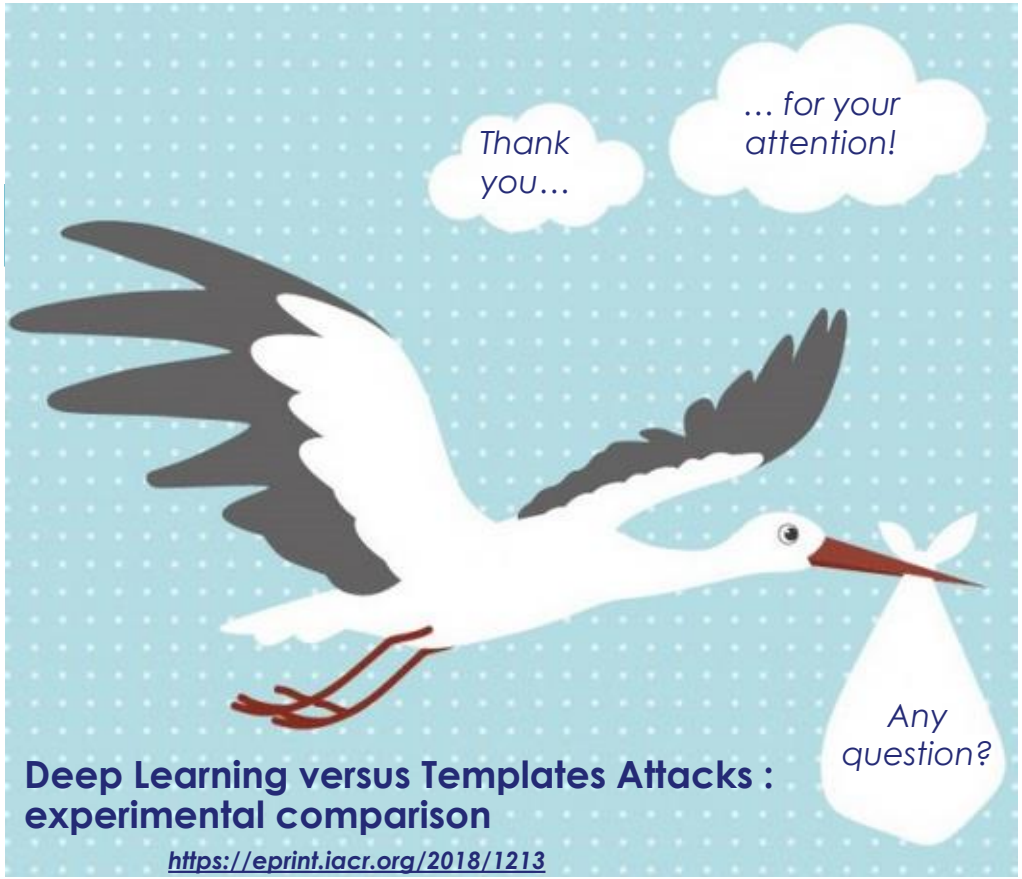
Protections against this potential new risk

- DL has heavy computational cost (huge memory, GPU…)
- Large words resist quite well (static target)
- Classical combined protections still work in synergy (desynchronization, shuffling, masking, limited exposure): need more against CNN?

Perspectives

- Normalization should be regarded as a plus towards portability (in test phase)
- Keep vigilance on old static/dynamic targets (e.g. CBC-MAC)
- Extension beyond 8 bit demands new neural networks architectures

THALES



Deep Learning versus Templates Attacks : experimental comparison

https://eprint.iacr.org/2018/1213