# Étude critique d'une méthode de Machine Learning appliquée à l'analyse par canaux auxiliaires

Sana Boussam, Julien Eynard, Guénaël Renault et Gabriel Zaid

THALES

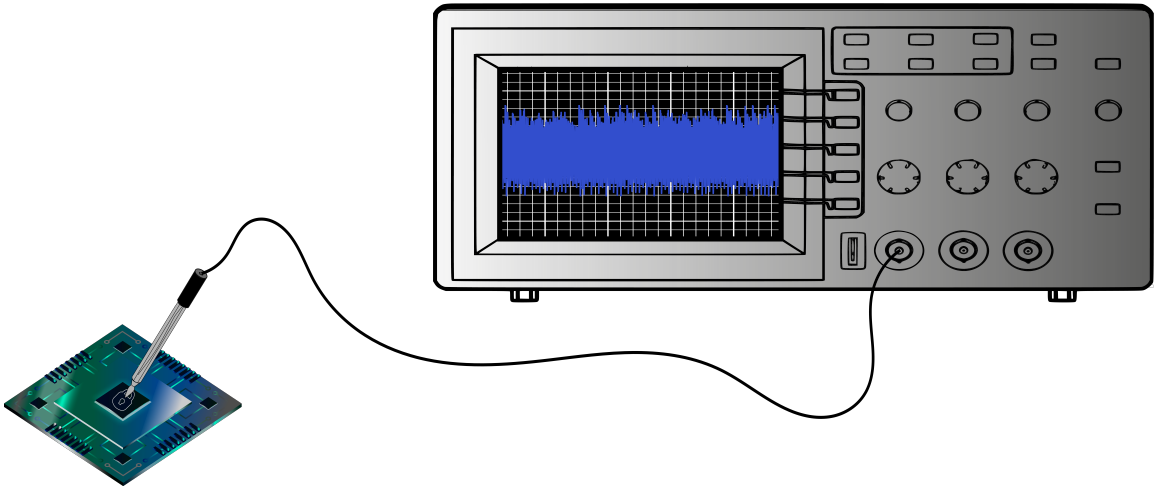
AGENCE NATIONALE DE LA SÉCURITÉ DES SYSTÈMES D'INFORMATION
ANSSI

INSTITUT POLYTECHNIQUE DE PARIS

Inria
INVENTEURS DU MONDE NUMÉRIQUE

# Sommaire

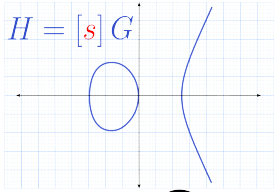
# Introduction



Attaques par canaux auxiliaires (SCA)

# Introduction

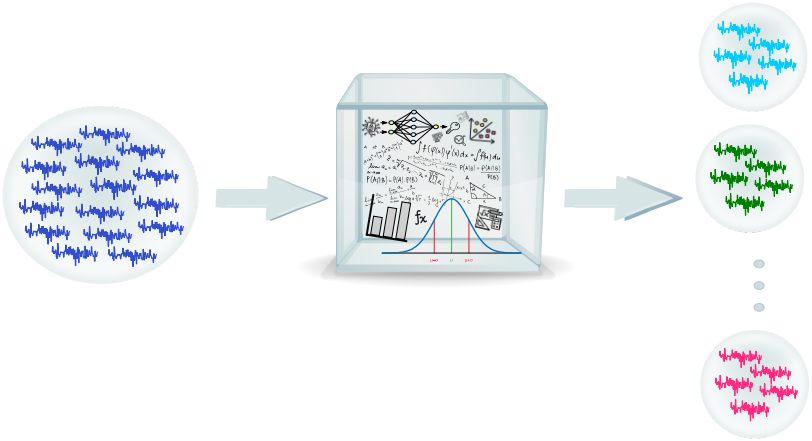
EC-DH (Elliptic Curve Diffie Hellman)





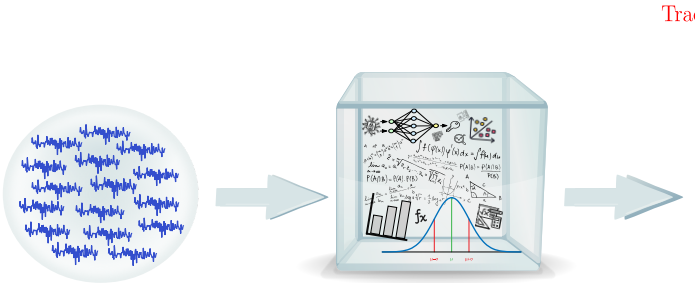

Attaques par canaux auxiliaires (SCA)

# Introduction



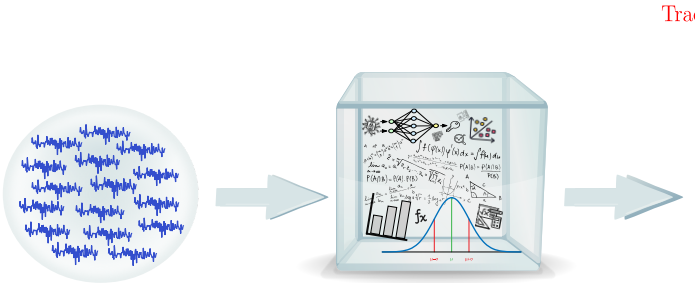
Problème d'étiquetage - Application aux attaques par canaux auxiliaires (SCA)

# Introduction




Problème d'étiquetage - Application aux attaques par canaux auxiliaires (SCA)

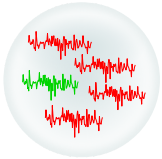
# Introduction

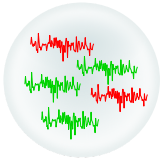



Problème d'étiquetage - Application aux attaques par canaux auxiliaires (SCA)

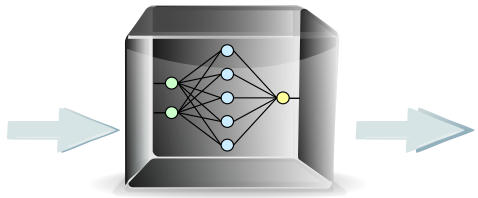
# Introduction

Traces manipulant un bit 0



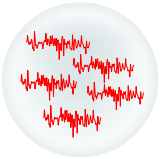


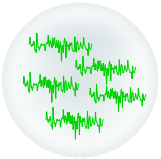
Traces manipulant un bit 1



Keep It Unsupervised : Horizontal Attacks Meet Deep Learning [PCBP20]
*présenté à CHES en 2021*

Traces manipulant un bit 0





Traces manipulant un bit 1

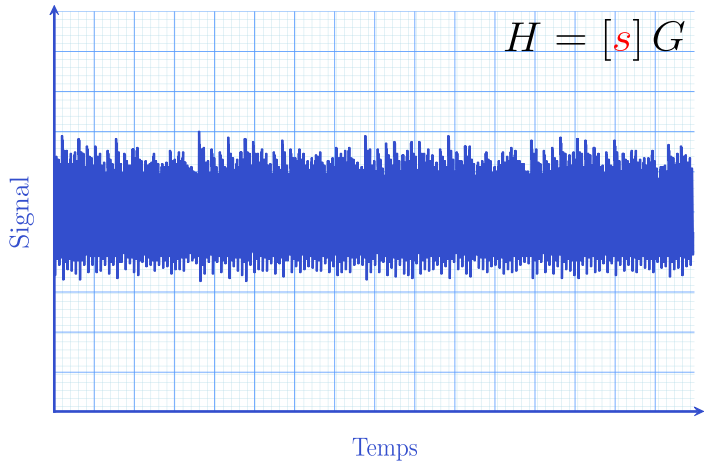
Problème d'étiquetage - Proposition d'une méthode corrective appliquée aux SCA

# Introduction


Problème d'étiquetage - Proposition d'une méthode corrective appliquée aux SCA

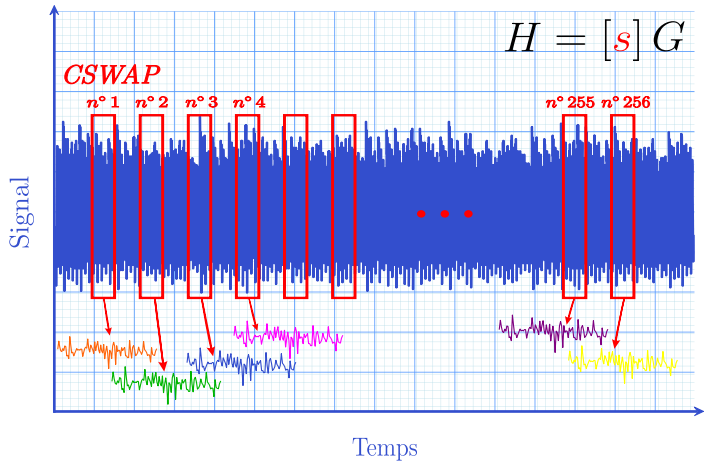
# Sommaire

# Identification de la cible



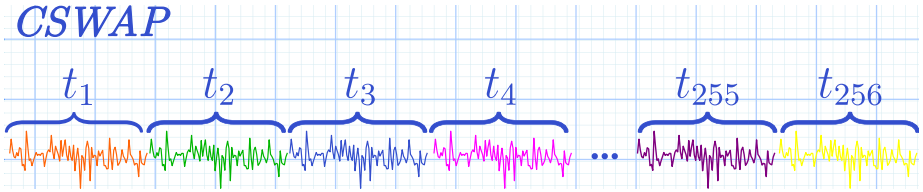

Trace d'une exécution complète d'une multiplication scalaire (courbe elliptique)

# Identification de la cible




Récupération des traces correspondant à l'exécution d'un *CSWAP*

# Identification de la cible



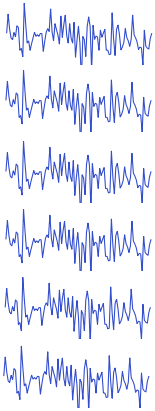

→ Une trace correspond à la concaténation des 256 *CSWAP*s

→ Chaque portion de trace $t_i$ est échantillonnée sur 1000 points et correspond à la manipulation du $i$-ème bit de s

→ Attaque bit par bit pour retrouver le scalaire s de 256 bits

# Vue d'ensemble de l'attaque

*CSWAP*



Applying Horizontal Clustering Side-Channel Attacks on Embedded ECC Implementations
*Nascimento, Chmielewski* [NC17]

Keep It Unsupervised : Horizontal Attacks Meet Deep Learning
*Perin, Chmielewski, Batina, Picek* [PCBP20]

Etape 1 : Labellisation des traces

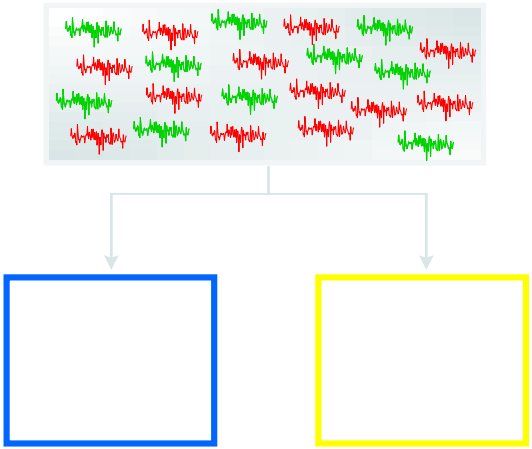
Etape 2 : Correction des labels

Traces manipulant un bit 0

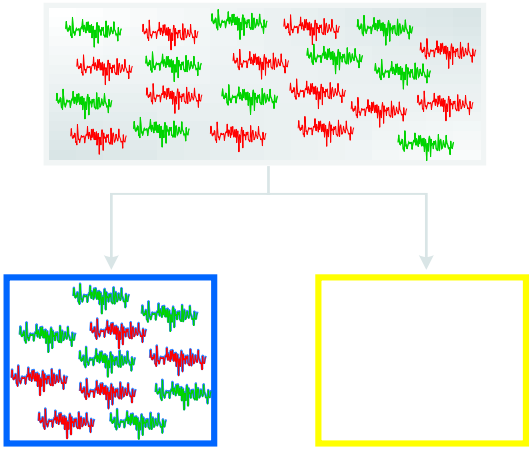
Traces manipulant un bit 1

Attaque Complète - IFSCA

# Présentation IFSCA - Étape 1

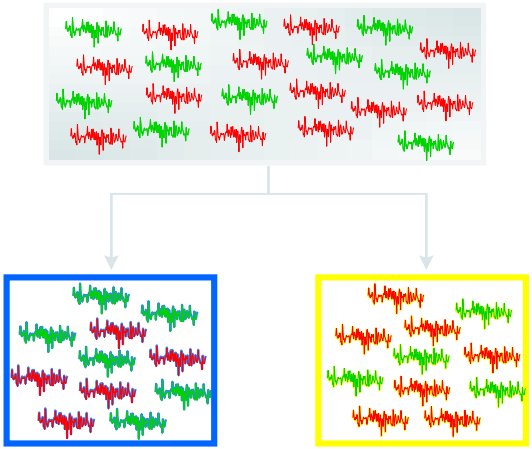


Étape 1 - Division du dataset $\mathcal{D}$ en deux sous-ensembles $\mathcal{D}_1$ (bleu) et $\mathcal{D}_2$ (jaune)



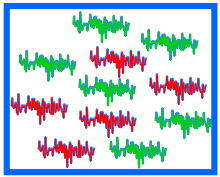
Étape 1 - Division du dataset $\mathcal{D}$ en deux sous-ensembles $\mathcal{D}_1$ (bleu) et $\mathcal{D}_2$ (jaune)

# Présentation IFSCA - Étape 1



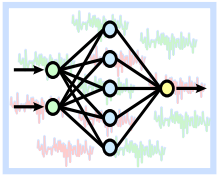
Étape 1 - Division du dataset $\mathcal{D}$ en deux sous-ensembles $\mathcal{D}_1$ (bleu) et $\mathcal{D}_2$ (jaune)

# Présentation IFSCA - Étape 2



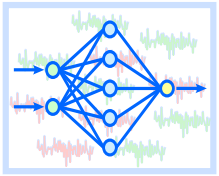
Étape 2 - Entraînement sur le dataset $\mathcal{D}_1$ (bleu) puis relabellisation du dataset $\mathcal{D}_2$ (jaune)

# Présentation IFSCA - Étape 2

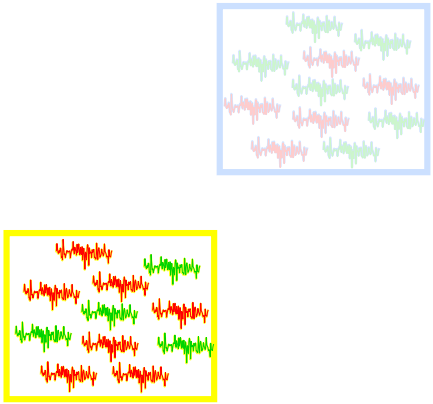


Étape 2 - Entraînement sur le dataset $\mathcal{D}_1$ (bleu) puis relabellisation du dataset $\mathcal{D}_2$ (jaune)

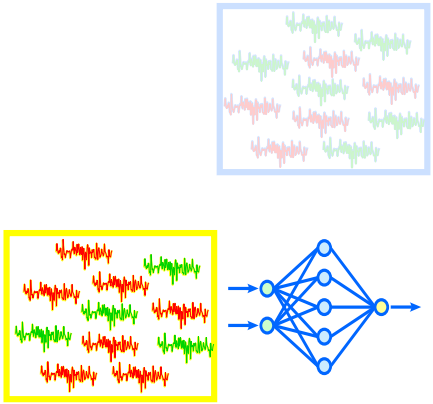
# Présentation IFSCA - Étape 2



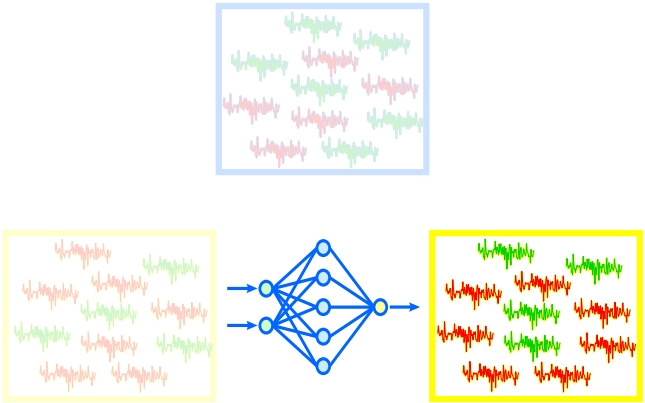
Étape 2 - Entraînement sur le dataset $\mathcal{D}_1$ (bleu) puis relabellisation du dataset $\mathcal{D}_2$ (jaune)



Étape 2 - Entraînement sur le dataset $\mathcal{D}_1$ (bleu) puis relabellisation du dataset $\mathcal{D}_2$ (jaune)

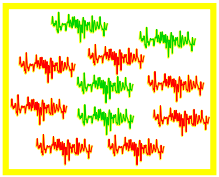


Étape 2 - Entraînement sur le dataset $\mathcal{D}_1$ (bleu) puis relabellisation du dataset $\mathcal{D}_2$ (jaune)



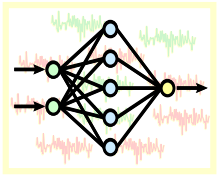
Étape 2 - Entraînement sur le dataset $\mathcal{D}_1$ (bleu) puis relabellisation du dataset $\mathcal{D}_2$ (jaune)

# Présentation IFSCA - Étape 3



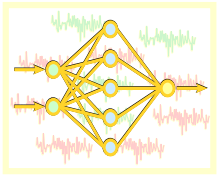
Étape 3 - Entraînement sur le dataset $\mathcal{D}_2$ (jaune) puis relabellisation du dataset $\mathcal{D}_1$ (bleu)

# Présentation IFSCA - Étape 3

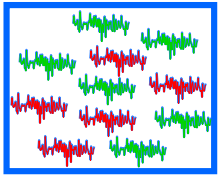


Étape 3 - Entraînement sur le dataset $\mathcal{D}_2$ (jaune) puis relabellisation du dataset $\mathcal{D}_1$ (bleu)

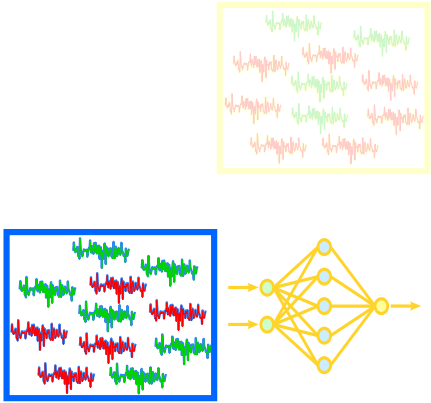
# Présentation IFSCA - Étape 3



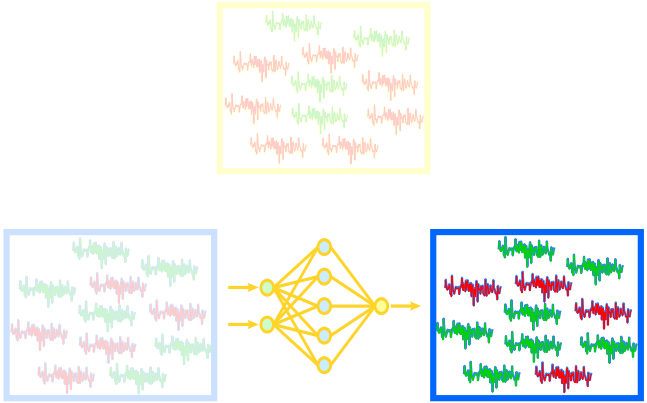
Étape 3 - Entraînement sur le dataset $\mathcal{D}_2$ (jaune) puis relabellisation du dataset $\mathcal{D}_1$ (bleu)



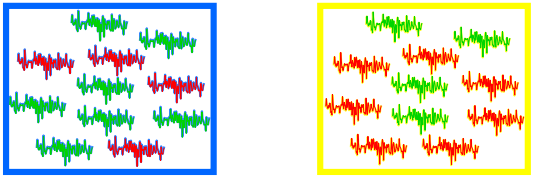
Étape 3 - Entraînement sur le dataset $\mathcal{D}_2$ (jaune) puis relabellisation du dataset $\mathcal{D}_1$ (bleu)



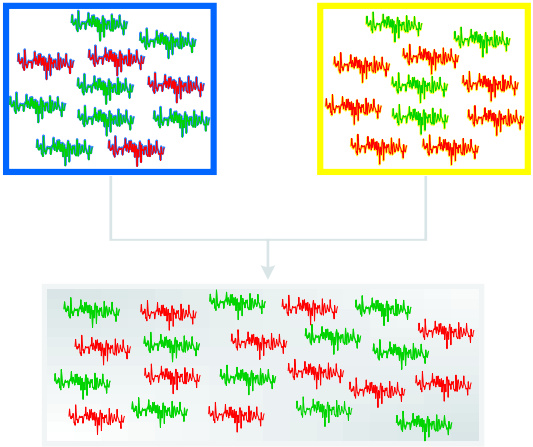
Étape 3 - Entraînement sur le dataset $\mathcal{D}_2$ (jaune) puis relabellisation du dataset $\mathcal{D}_1$ (bleu)



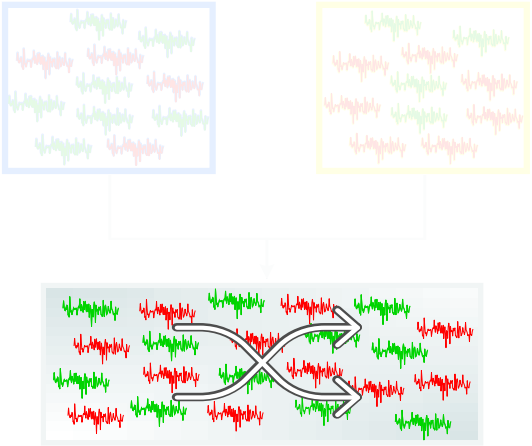
Étape 3 - Entraînement sur le dataset $\mathcal{D}_2$ (jaune) puis relabellisation du dataset $\mathcal{D}_1$ (bleu)



Étape 4 - Fusion des deux sous-ensembles $\mathcal{D}_1$ (bleu) et $\mathcal{D}_2$ (jaune)



Étape 4 - Fusion des deux sous-ensembles $\mathcal{D}_1$ (bleu) et $\mathcal{D}_2$ (jaune)



Étape 4 - Fusion des deux sous-ensembles $\mathcal{D}_1$ (bleu) et $\mathcal{D}_2$ (jaune)

# Résultats obtenus

| | Précision moyenne | Précision maximale |
|---|---|---|
| Sortie de la phase de pré-labellisation | $52.24\%$ | $59.22\%$ |
| Application IFSCA | $91\%$ | $100\%$ |

# Résultats obtenus

- $\sim 55\% \rightarrow \sim 100\%$ ?
- Approche aussi générique que prétendue ?

# Sommaire

# Étude critique de l'IFSCA

| | Précision moyenne | Précision maximale |
|---|---|---|
| Sortie de la phase de pré-labellisation | $52.24\%$ | $59.22\%$ |
| IFSCA sans régularisation | $85\%$ | $97.64\%$ |
| IFSCA avec régularisation | $91\%$ | $100\%$ |

# Étude critique de l'IFSCA

| | Précision moyenne | Précision maximale |
|---|---|---|
| Sortie de la phase de pré-labellisation | $52.24\%$ | $59.22\%$ |
| IFSCA sans régularisation | $85\%$ | $97.64\%$ |
| IFSCA avec régularisation | $91\%$ | $100\%$ |

# Étude critique de l'IFSCA

| | Précision moyenne | Précision maximale |
|---|---|---|
| Sortie de la phase de pré-labellisation | $52.24\%$ | $59.22\%$ |
| IFSCA sans régularisation | $85\%$ | $97.64\%$ |
| IFSCA avec régularisation | $91\%$ | $100\%$ |

# Étude critique de l'IFSCA

| | Précision moyenne | Précision maximale |
|---|---|---|
| Sortie de la phase de pré-labellisation | $52.24\%$ | $59.22\%$ |
| IFSCA sans régularisation | $85\%$ | $97.64\%$ |
| IFSCA avec régularisation | $91\%$ | $100\%$ |

Notre hypothèse : Le réseau utilisé dans l'IFSCA est trop complexe.

# Étude critique de l'IFSCA

| Couches | `cswap_pointer` |
|---|---|
| Entrées | traces de 1000 points |
| Conv1D_1 | 8 filtres |
| Conv1D_2 | 16 filtres |
| Conv1D_3 | 32 filtres |
| Dense_1 | 100 neurones |
| Dense_2 | 100 neurones |
| *Softmax* | 2 neurones |

Architecture du CNN utilisée dans l'IFSCA

# Étude critique de l'IFSCA

| Couches | `cswap_pointer` |
|---|---|
| Entrées | traces de 1000 points |
| Conv1D_1 | 8 filtres |
| Conv1D_2 | 16 filtres |
| Conv1D_3 | 32 filtres |
| Dense_1 | 100 neurones |
| Dense_2 | 100 neurones |
| *Softmax* | 2 neurones |

Architecture du CNN utilisée dans l'IFSCA

Ce modèle semble beaucoup trop complexe

# Étude critique de l'IFSCA

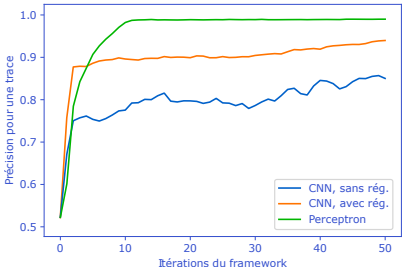
| Couches | `cswap_pointer` |
|---|---|
| Entrées | traces de 1000 points |
| Conv1D_1 | 8 filtres |
| Conv1D_2 | 16 filtres |
| Conv1D_3 | 32 filtres |
| Dense_1 | 100 neurones |
| Dense_2 | 100 neurones |
| *Softmax* | 2 neurones |

Architecture du CNN utilisée dans l'IFSCA

| Couches | `cswap_pointer` |
|---|---|
| Entrées | traces de 1000 points |
| *Sigmoid* | 1 neurone |

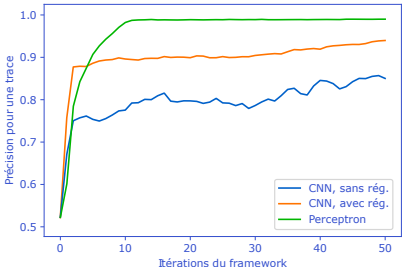
Architecture proposée - Perceptron

# Étude critique de l'IFSCA




Comparaison des performances des réseaux de neurones

# Étude critique de l'IFSCA




Comparaison des performances des réseaux de neurones

| | CNN (IFSCA) | Perceptron |
|---|---|---|
| Nb. de paramètres | 45 978 | 1 001 |
| Temps de l'attaque | 75 min. | 5 min. |

# Proposition d'une approche alternative

**Algorithme :** Attaque alternative proposée

**Entrées :** Traces d'exécution

**Sorties :** Un scalaire entièrement reconstitué

**pour chaque** *instant temporel* $p = 1...1000$ **faire**

  Application d'une $k$-moyenne sur les traces au point $p$ pour les labelliser

  **pour chaque** $j = 0...9$ **faire** // 9 bits à corriger par trace maximum (recherche exhaustive calibrée)

    **pour chaque** *trace complète* **faire**

      Correction des $j$ bits

      **si** *Requête à l'Oracle* $=$ *SUCCES* **alors**

        **retourner** `Scalaire trouvé`

      **fin**

    **fin**

  **fin**

**fin**

# Proposition d'une approche alternative

→ Complexité totale de l'attaque inférieure à $\mathbf{2^{70}}$

→ Au point temporel $643$, précision maximale de $\mathbf{97.68\%}$ soit **6 bits à corriger** (environ $\mathbf{2^{39}}$ tests)

⇒ Données biaisées

# Sommaire

# Conclusion

→ L'approche présentée dans [PCBP20] est trop compliquée pour le jeu de données considéré

→ Une **utilisation réfléchie** des outils de *machine learning* est nécessaire pour obtenir de meilleurs résultats, qui soient plus interprétables [Cag18, Mas20, Zai21]

# Conclusion

→ Pour plus d'informations concernant cette présentation : voir l'article associé

→ Pour aller encore plus loin : étude plus poussée de l'IFSCA faite en collaboration avec le CEA-Leti en cours de parution

# Merci de votre attention !

Contact: sana.boussam@inria.fr

# Bibliographie I




Eleonora Cagli.
*Feature Extraction for Side-Channel Attacks. (Extraction de caractéristiques pour les attaques par canaux auxiliaires)*.
PhD thesis, Sorbonne University, France, 2018.



Loïc Masure.
*Towards a better comprehension of deep learning for side-channel analysis. (Vers une meilleure compréhension de l'apprentissage profond appliqué aux attaques par observations)*.
PhD thesis, Sorbonne University, Paris, France, 2020.

# Bibliographie II



Erick Nascimento and Łukasz Chmielewski.
Applying Horizontal Clustering Side-Channel Attacks on Embedded ECC Implementations.
In Thomas Eisenbarth and Yannick Teglia, editors, *Smart Card Research and Advanced Applications*, Lecture Notes in Computer Science, pages 213–231, Cham, 2017. Springer International Publishing.


Guilherme Perin, Łukasz Chmielewski, Lejla Batina, and Stjepan Picek.
Keep it Unsupervised : Horizontal Attacks Meet Deep Learning.
*IACR Transactions on Cryptographic Hardware and Embedded Systems*, pages 343–372, December 2020.


Gabriel Zaid.
*Bridging Deep Learning and Classical Profiled Side-Channel Attacks. (Rapprochement de l'apprentissage profond et des attaques par canaux auxiliaires)*.
PhD thesis, University of Lyon, France, 2021.