

# Apprentissage profond pour les attaques par analyse de canaux auxiliaires des implémentations de fonctions cryptographiques

Brice Colombier, Damien Robissout, Gabriel Zaid,  
Lilian Bossuet, Amaury Habrard

Univ Lyon, UJM-Saint-Etienne, CNRS, Laboratoire Hubert Curien UMR 5516

FETCH 2020



Brice Colombier

Chercheur post-doctorant à  
l'Université Jean Monnet,  
Saint-Étienne, France



Domaines de recherche :

- Sécurité matérielle,
- Contrefaçon de circuits intégrés,
- Attaques et évaluation,
- Génération de nombres aléatoires.



© les contributeurs d'OpenStreetMap

# Contexte

---

“Apprentissage profond pour les attaques par analyse de canaux auxiliaires des implémentations de fonctions cryptographiques”.

*“implémentations de fonctions cryptographiques”*

🔒 Sécurité → Confidentialité → Chiffrement par **bloc** : AES

“Apprentissage profond pour les attaques par analyse de canaux auxiliaires des implémentations de fonctions cryptographiques”.

*“implémentations de fonctions cryptographiques”*

🔒 Sécurité → Confidentialité → Chiffrement par **bloc** : AES

Message découpé en **blocs** de 128 bits, **clé** de 128 bits.

État courant : 16 octets

0,0	0,1	0,2	0,3
1,0	1,1	1,2	1,3
2,0	2,1	2,2	2,3
3,0	3,1	3,2	3,3

Transformations successives (10×) :

- **AddRoundKey** : OU exclusif
- **SubBytes** : substitution
- **ShiftRows**
- **MixColumns**

**Important**

Opérations réalisées sur les **octets**

“Apprentissage profond pour les attaques par analyse de canaux auxiliaires des implémentations de fonctions cryptographiques”.

*“implémentations de fonctions cryptographiques”*

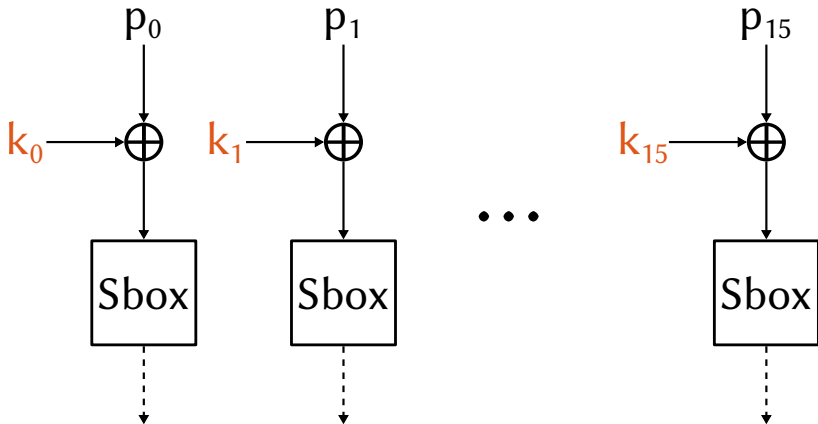
🔒 Sécurité → Confidentialité → Chiffrement par bloc : AES

	-0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-A	-B	-C	-D	-E	-F
0-	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1-	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2-	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3-	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4-	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5-	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6-	DO	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7-	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8-	CD	OC	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9-	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	OB	DB
A-	EO	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B-	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C-	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D-	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E-	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F-	8C	A1	89	OD	BF	E6	42	68	41	99	2D	OF	BO	54	BB	16

“Apprentissage profond pour les attaques par analyse de canaux auxiliaires des implémentations de fonctions cryptographiques”.

*“implémentations de fonctions cryptographiques”*



🔒 Sécurité → Confidentialité → Chiffrement par bloc : AES



“Apprentissage profond pour les attaques par analyse de canaux auxiliaires des implémentations de fonctions cryptographiques”.

*“attaques par analyse de canaux auxiliaires”*

Canaux (non prévus) donnant de l'**information** sur le système

-  Consommation de puissance
-  Rayonnement électromagnétique

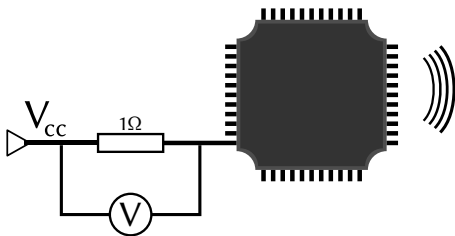


“Apprentissage profond pour les attaques par analyse de canaux auxiliaires des implémentations de fonctions cryptographiques”.

*“attaques par analyse de canaux auxiliaires”*

Canaux (non prévus) donnant de l'**information** sur le système

- ⚡ Consommation de puissance
- 📶 Rayonnement électromagnétique



Dans la plupart de cas, la **grandeur physique** mesurée est **proportionnelle au poids de Hamming** de la donnée manipulée.

“Apprentissage profond pour les attaques par analyse de canaux auxiliaires des implémentations de fonctions cryptographiques”.

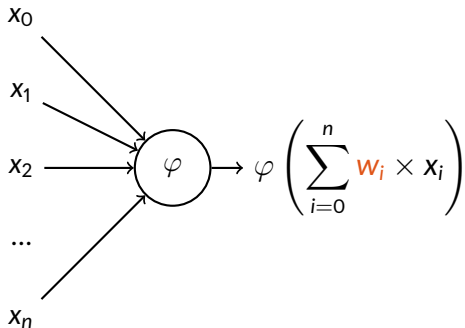
*“Apprentissage profond”*

Utilisation de réseaux de neurones.

“Apprentissage profond pour les attaques par analyse de canaux auxiliaires des implémentations de fonctions cryptographiques”.

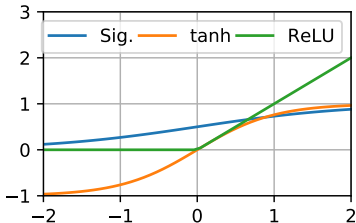
### “Apprentissage profond”

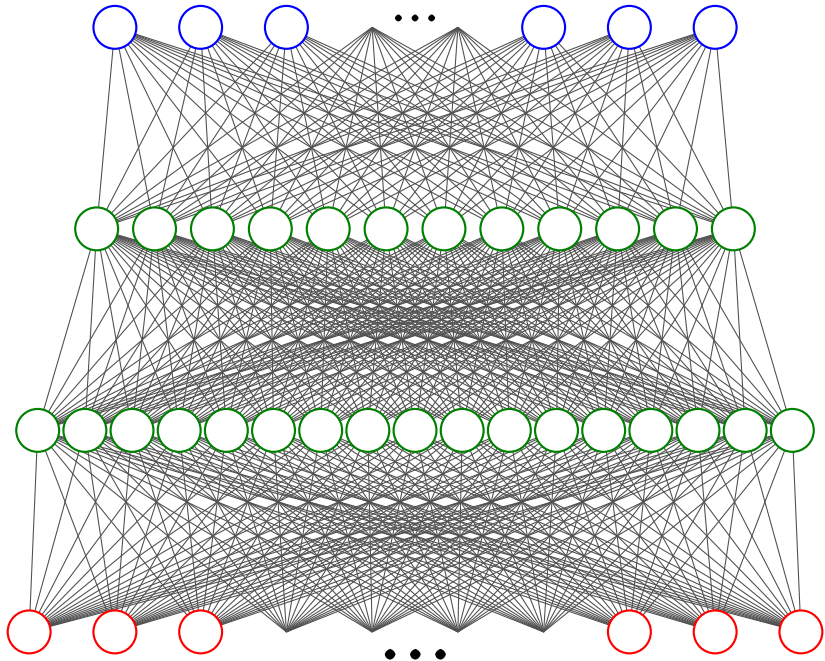
Utilisation de réseaux de neurones.



Fonction d'activation  $\varphi$  :

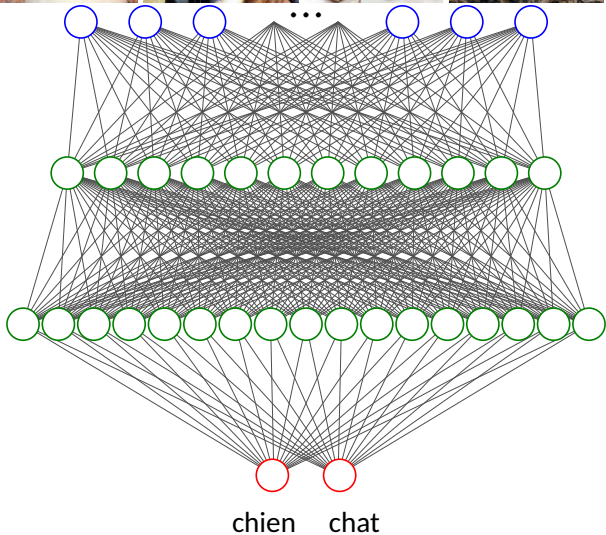
- ▶ Sigmoide :  $\frac{e^x}{e^x+1}$
- ▶ Tan hyperbolique :  $\tanh(x)$
- ▶ ReLU :  $\max(0, x)$

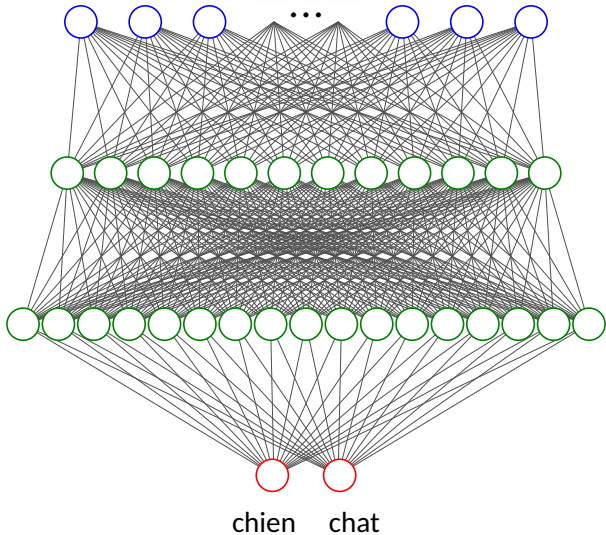


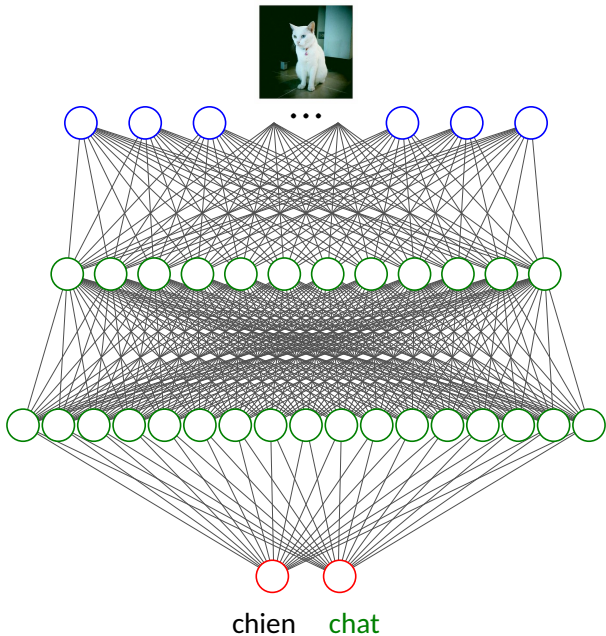


# Déroulement d'une attaque

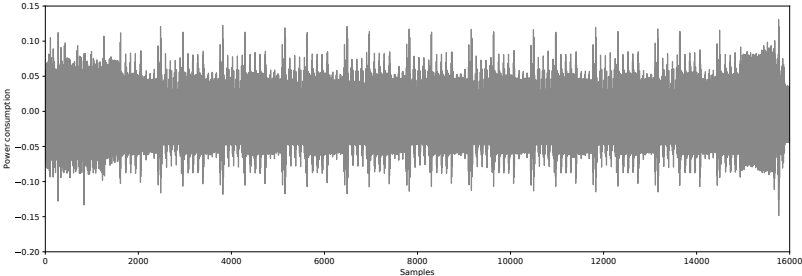
---

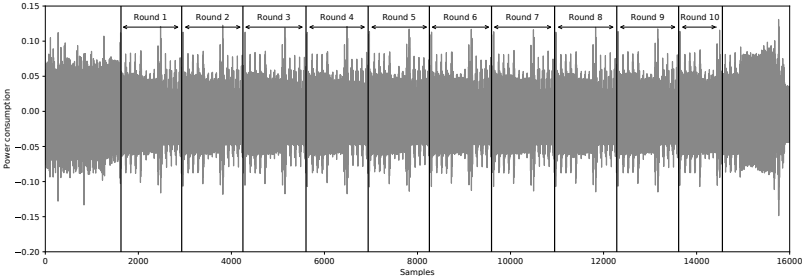


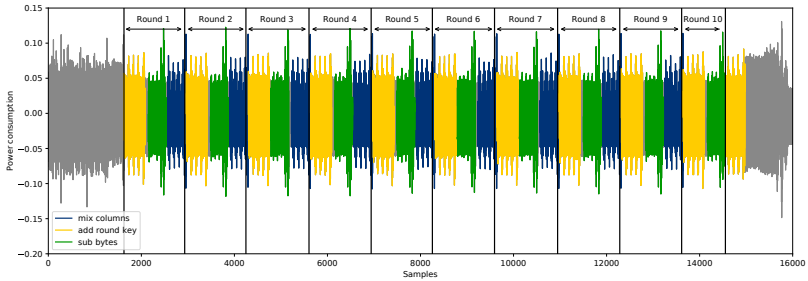


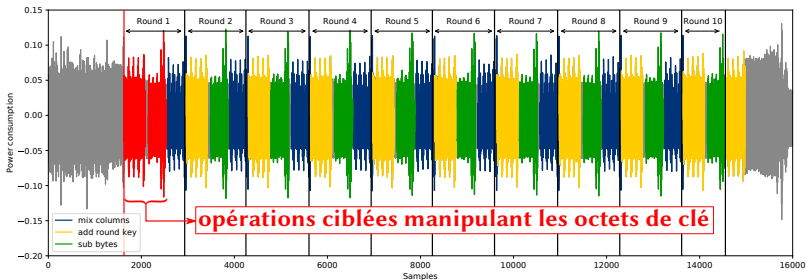












Différences avec l'utilisation **classique** des réseaux de neurones :

- Information extrêmement **localisée**,
- Nombreux échantillons **non informatifs**,
- Nécessité d'**accumulation statistique** des prédictions,

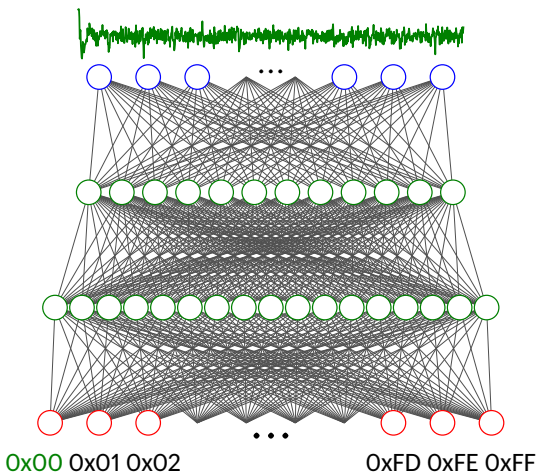
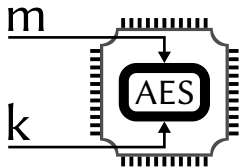
## Phase 1 : apprentissage

Système **maîtrisé** :

- Message **aléatoire**,
- Clé **aléatoire**.

**Entrée** : mesure physique

**Labels** : octet secret connu



## Principe

Mise à jour des **poids** du réseau ( $w_i$ ) pour **minimiser l'erreur**.

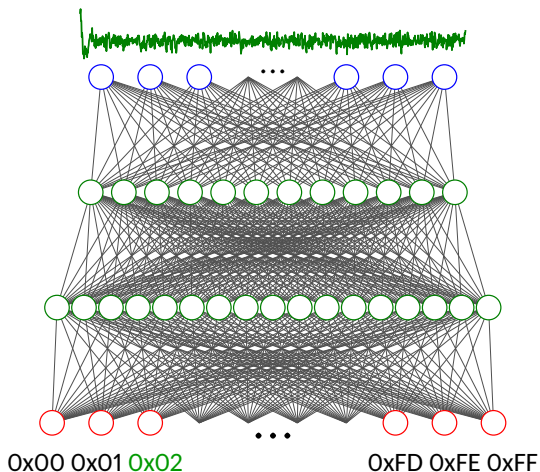
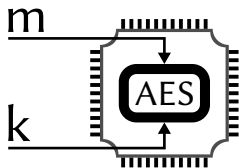
## Phase 1 : apprentissage

Système maîtrisé :

- Message **aléatoire**,
- Clé **aléatoire**.

Entrée : mesure physique

Labels : octet secret connu



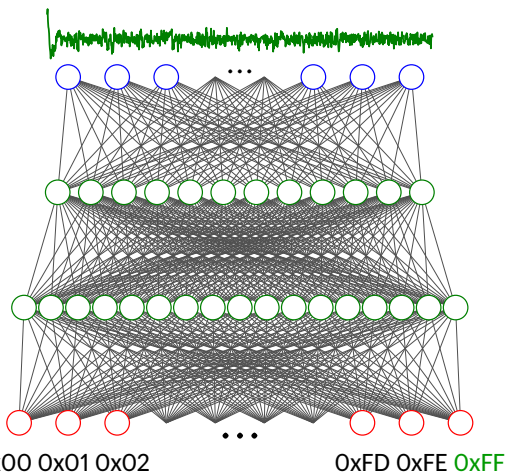
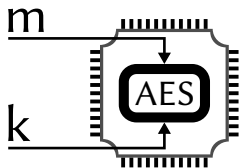
## Principe

Mise à jour des **poids** du réseau ( $w_i$ ) pour **minimiser l'erreur**.

## Phase 1 : apprentissage

Système **maîtrisé** :

- Message **aléatoire**,
- Clé **aléatoire**.

**Entrée** : mesure physique**Labels** : octet secret connu

## Principe

Mise à jour des **poids** du réseau ( $w_i$ ) pour **minimiser l'erreur**.

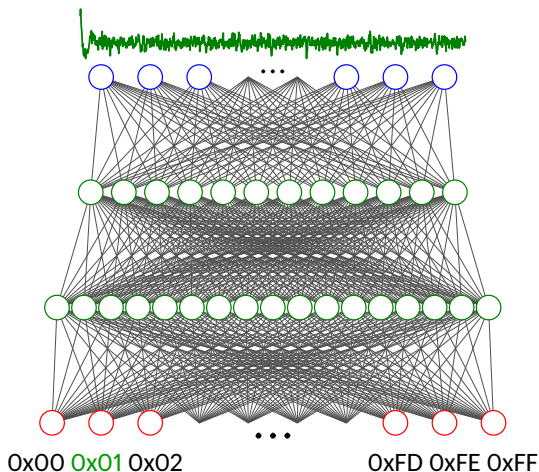
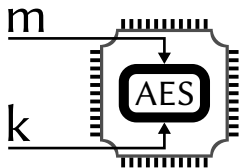
## Phase 1 : apprentissage

Système maîtrisé :

- Message **aléatoire**,
- Clé **aléatoire**.

Entrée : mesure physique

Labels : octet secret connu



## Principe

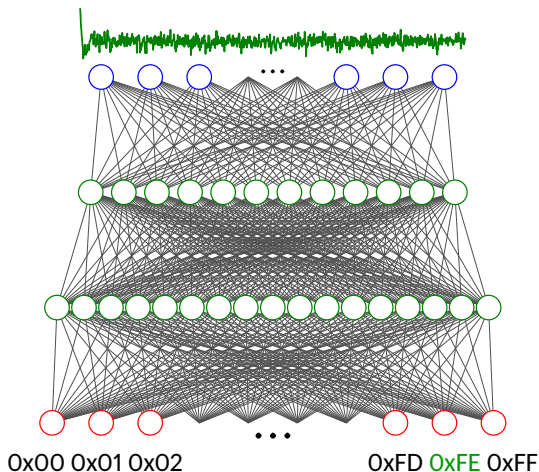
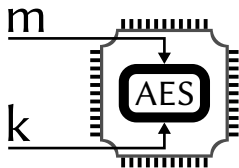
Mise à jour des **poids** du réseau ( $w_i$ ) pour **minimiser l'erreur**.



## Phase 1 : apprentissage

Système **maîtrisé** :

- Message **aléatoire**,
- Clé **aléatoire**.

**Entrée** : mesure physique**Labels** : octet secret connu

## Principe

Mise à jour des **poids** du réseau ( $w_i$ ) pour **minimiser l'erreur**.

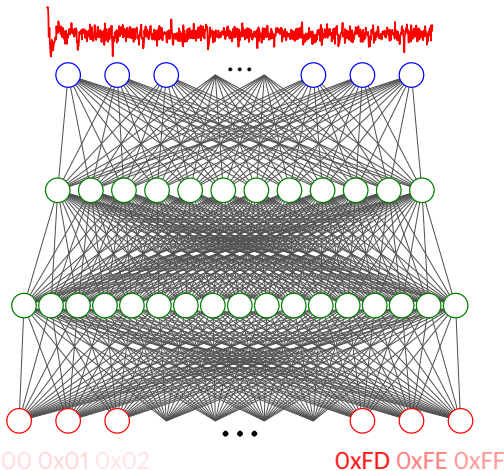
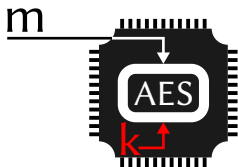
## Phase 2 : inférence

Système inconnu :

- Message **aléatoire**,
- Clé **fixe (inconnue)**.

Entrée : mesure physique

Sorties : prédictions



## Résultat

Obtention d'une **probabilité** pour chaque octet secret.

**Problème** : une seule inférence ne suffit souvent pas.

- Information trop **bruitée**,
- Confiance trop **faible**.

**Problème** : une seule inférence ne suffit souvent pas.

- Information trop **bruitée**,
- Confiance trop **faible**.

### Solution

**Accumuler** l'information des inférences successives.

$$P_{finale}(\text{octet}) = \sum_{i=0}^{\#inférences-1} \log(P_i(\text{octet}))$$

**Problème** : une seule inférence ne suffit souvent pas.

- Information trop **bruitée**,
- Confiance trop **faible**.

### Solution

**Accumuler** l'information des inférences successives.

$$P_{finale}(\text{octet}) = \sum_{i=0}^{\#inférences-1} \log(P_i(\text{octet}))$$

Octet	$P_0$
0x00	0.1
0x01	0.2
...	...
0xFE	0.4
0xFF	0.3

**Problème** : une seule inférence ne suffit souvent pas.

- Information trop **bruitée**,
- Confiance trop **faible**.

### Solution

**Accumuler** l'information des inférences successives.

$$P_{finale}(\text{octet}) = \sum_{i=0}^{\#inférences-1} \log(P_i(\text{octet}))$$

Octet	$P_0$	$P_1$
0x00	0.1	0.4
0x01	0.2	0.2
...	...	...
0xFE	0.4	0.3
0xFF	0.3	0.1

**Problème** : une seule inférence ne suffit souvent pas.

- Information trop **bruitée**,
- Confiance trop **faible**.

### Solution

**Accumuler** l'information des inférences successives.

$$P_{finale}(\text{octet}) = \sum_{i=0}^{\#inférences-1} \log(P_i(\text{octet}))$$

Octet	$P_0$	$P_1$	$P_2$
0x00	0.1	0.4	0.2
0x01	0.2	0.2	0.3
...	...	...	...
0xFE	0.4	0.3	0.4
0xFF	0.3	0.1	0.1

**Problème** : une seule inférence ne suffit souvent pas.

- Information trop **bruitée**,
- Confiance trop **faible**.

### Solution

**Accumuler** l'information des inférences successives.

$$P_{finale}(\text{octet}) = \sum_{i=0}^{\#inférences-1} \log(P_i(\text{octet}))$$

Octet	$P_0$	$P_1$	$P_2$	$P_3$
0x00	0.1	0.4	0.2	0.4
0x01	0.2	0.2	0.3	0.1
...	...	...	...	...
0xFE	0.4	0.3	0.4	0.3
0xFF	0.3	0.1	0.1	0.2



**Problème** : une seule inférence ne suffit souvent pas.

- Information trop **bruitée**,
- Confiance trop **faible**.

### Solution

**Accumuler** l'information des inférences successives.

$$P_{finale}(\text{octet}) = \sum_{i=0}^{\#inférences-1} \log(P_i(\text{octet}))$$

Octet	$P_0$	$P_1$	$P_2$	$P_3$	$P_{finale}$
0x00	0.1	0.4	0.2	0.4	-2.5
0x01	0.2	0.2	0.3	0.1	-2.9
...	...	...	...	...	...
0xFE	0.4	0.3	0.4	0.3	-1.8
0xFF	0.3	0.1	0.1	0.2	-3.2

**Problème** : une seule inférence ne suffit souvent pas.

- Information trop **bruitée**,
- Confiance trop **faible**.

### Solution

**Accumuler** l'information des inférences successives.

$$P_{finale}(\text{octet}) = \sum_{i=0}^{\#inférences-1} \log(P_i(\text{octet}))$$

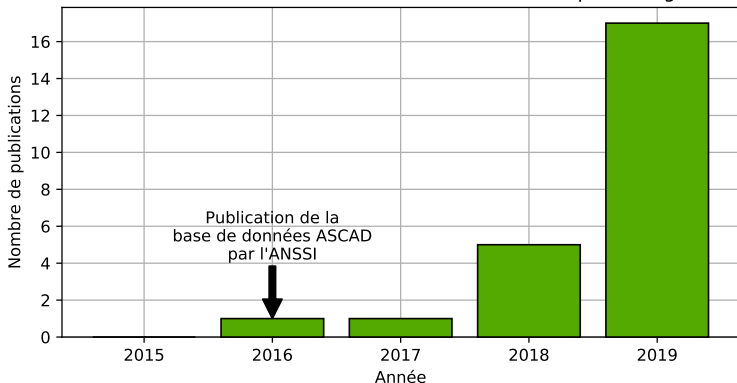
Octet	$P_0$	$P_1$	$P_2$	$P_3$	$P_{finale}$	
0x00	0.1	0.4	0.2	0.4	-2.5	✗
0x01	0.2	0.2	0.3	0.1	-2.9	✗
...	...	...	...	...	...	
0xFE	0.4	0.3	0.4	0.3	-1.8	✓
0xFF	0.3	0.1	0.1	0.2	-3.2	✗

Avantages et inconvénients

Tendances et challenges

---

Nombre de publications par an étudiant les attaques par canaux auxiliaires dont le titre ou les mots-clés mentionnent "deep learning"



source : Cryptology ePrint Archive

<https://eprint.iacr.org>

R. Benadjila et al. "Deep learning for side-channel analysis and introduction to ASCAD database". *Journal of Cryptographic Engineering* (2019)

En comparaison des attaques par *template*, les **avantages** sont :

- **Bons** résultats obtenus relativement **facilement** sur des cibles non protégées,
- Pas de sélection des **points d'intérêt** nécessaire,
- Insensible à la **désynchronisation** des traces (invariance temporelle),
- Applicable aux implémentations **protégées** par des contre-mesures.

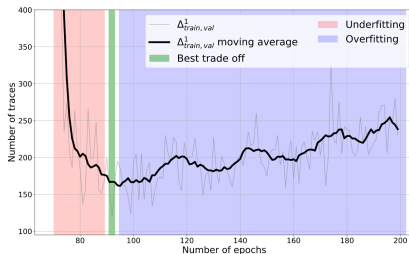


Néanmoins, des **inconvénients** subsistent...

- Choix des **hyper-paramètres** difficile,
- **Performances** inférieures aux *templates* dans certains cas,
- Utilisation **non optimale** des réseaux,
- **Explicabilité** des résultats limitée.



## Utilisation : optimisation du processus d'apprentissage [1]



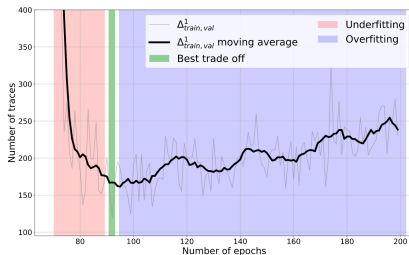
Évaluer le réseau lors de l'apprentissage et l'**arrêter** au bon moment :

- Meilleures **performances** lors de l'attaque (moins de traces),
- **Coût** d'entraînement moindre,
- **Exploration** des configurations possibles plus rapide.

---

[1] D. Robissout et al. "Online Performance Evaluation of Deep Learning Networks for Side-Channel Analysis". *International Workshop on Constructive Side-Channel Analysis and Secure Design (to be published)*. 2020.

## Utilisation : optimisation du processus d'apprentissage [1]



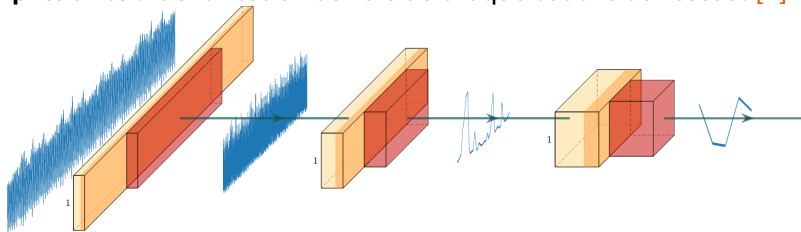
Évaluer le réseau lors de l'apprentissage et l'arrêter au bon moment :

	ASCAD	Travaux LabHC
Temps d'apprentissage	1 h	40 min
Traces d'attaque requises	1151	802

[1] D. Robissout et al. "Online Performance Evaluation of Deep Learning Networks for Side-Channel Analysis". *International Workshop on Constructive Side-Channel Analysis and Secure Design (to be published)*. 2020.



**Explicabilité** : identification du rôle de chaque couche du réseau. [2]

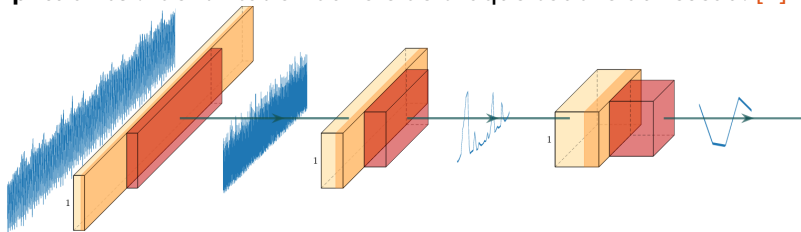


- Couche 1 : détection et combinaison des points d'intérêt,
- Couche 2 : détection et correction de la désynchronisation,
- Couche 3 : réduction de la dimension de la trace.

---

[2] G. Zaid, L. Bossuet, A. Habrard, and A. Venelli. "Methodology for Efficient CNN Architectures in Profiling Attacks". *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2020.1 (2020), pp. 1-36.

**Explicabilité** : identification du rôle de chaque couche du réseau. [2]



	ASCAD	Travaux LabHC
Nombre de paramètres	66 652 444	16 960
Temps d'apprentissage	1 h 30 min	4 min
Traces d'attaque requises	1146	191

[2] G. Zaid, L. Bossuet, A. Habrard, and A. Venelli. "Methodology for Efficient CNN Architectures in Profiling Attacks". *IACR Transactions on Cryptographic Hardware and Embedded Systems 2020.1* (2020), pp. 1–36.

## Tendances :

- Mise en place de *benchmarks* : base de données **ASCAD** [3]
- Peu de travaux en apprentissage **non supervisé**
- Très bons résultats sur d'autres fonctions cryptographiques, en particulier la cryptographie **asymétrique**
  - Attaque réussie en **une trace** sur RSA [4]
- Utilisation pour l'**évaluation** de sécurité [5]
  - "Si le réseau trouve quelque chose, il y a quelque chose"
  - Ne dit rien de l'**exploitabilité** des fuites.

---

[3] R. Benadjila et al. "Deep learning for side-channel analysis and introduction to ASCAD database". *Journal of Cryptographic Engineering* (2019).

[4] M. Carbone et al. "Deep Learning to Evaluate Secure RSA Implementations". *IACR Transactions on Cryptographic Hardware and Embedded Systems 2019.2* (2019), pp. 132–161.

[5] F. Wegener, T. Moos, and A. Moradi. "DL-LA: Deep Learning Leakage Assessment: A modern roadmap for SCA evaluations". *IACR Cryptology ePrint Archive 2019* (2019), p. 505.

Nouvelles méthodes d'attaque basées sur l'apprentissage profond :

- Efficaces sur de nombreuses cibles,
- Beaucoup de recherche sur le sujet.

Nouvelles méthodes d'attaque basées sur l'apprentissage profond :

- Efficaces sur de nombreuses cibles,
- Beaucoup de recherche sur le sujet.

**Défis** associés non négligeables :

- Dépassement l'utilisation en "boîte noire",
- Compréhension du succès de l'attaque,
- Exploitabilité des résultats,

Nouvelles méthodes d'attaque basées sur l'apprentissage profond :

- Efficaces sur de nombreuses cibles,
- Beaucoup de recherche sur le sujet.

**Défis** associés non négligeables :

- Dépassement l'utilisation en "boîte noire",
- Compréhension du succès de l'attaque,
- Exploitabilité des résultats,
- **Conception de contre-mesures.**

Nouvelles méthodes d'attaque basées sur l'apprentissage profond :

- Efficaces sur de nombreuses cibles,
- Beaucoup de recherche sur le sujet.

**Défis** associés non négligeables :

- Dépassement l'utilisation en "boîte noire",
- Compréhension du succès de l'attaque,
- Exploitabilité des résultats,
- **Conception de contre-mesures.**

— Questions? —