Processor Extensions for Security

Arnaud Tisserand : Chercheur au CNRS

# I. Introduction

On s’intéresse à la sécurité des systèmes embarqués (clé de voiture, carte pass, puce téléphone, robot). Le but est de se protéger contre de nombreux malfaisants.

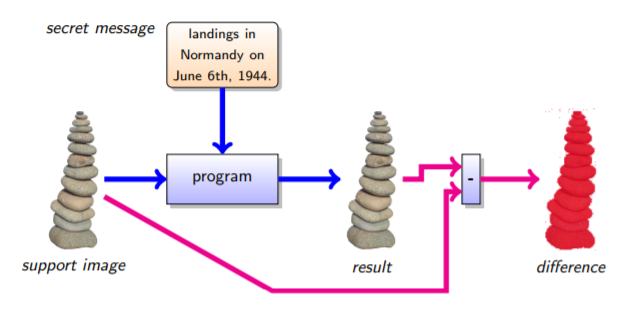
## Aspects sécurité

Cryptanalyse :

2 types :

Cryptanalyse physique : On va espionner des puces.

Stéganographie : C’est l’art de la dissimulation



Les images paraissent similaires mais on voit sur l’image rouge qu’il y a des différences sur toute l’image. Cette technique est utilisée par les terroristes.

**Les caractéristiques de la cryptographie :**

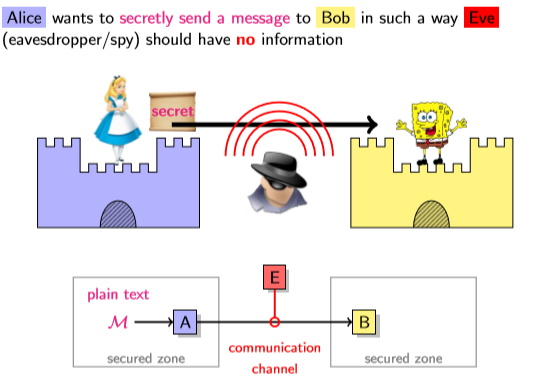
Le plus important est l’authenticité et l’intégrité.

Il y a aussi la confidentialité et la non répudiation.

DSP : filtrage sur le réseau

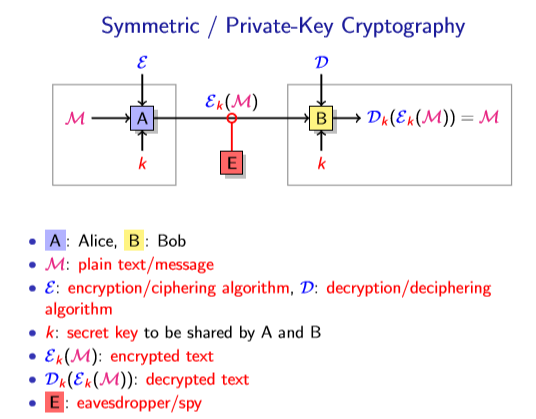
GPU : processeur fait pour accélérer les calculs sur les ordinateurs.

# II. Security Background



À partir du moment où physiquement ça sort de l’enceinte sécurisée, les données sont connues. C’est pourquoi il faut les protéger.

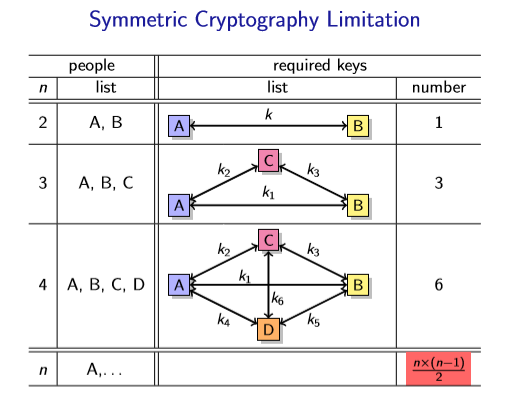
Plain text : Message en clair



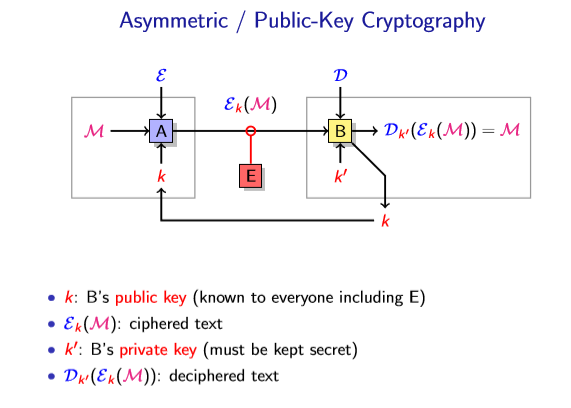
Alice et Bob se mettent d’accord sur des algos de chiffrement et de déchiffrement :

Alice a une clé **k** (entier d’un gros nombre à dizaine de bits jusqu’à plus d’une centaine de bits).

Donc si Eve n’a pas la clé elle ne peut pas décrypter. Il faut impérativement protéger la clé.



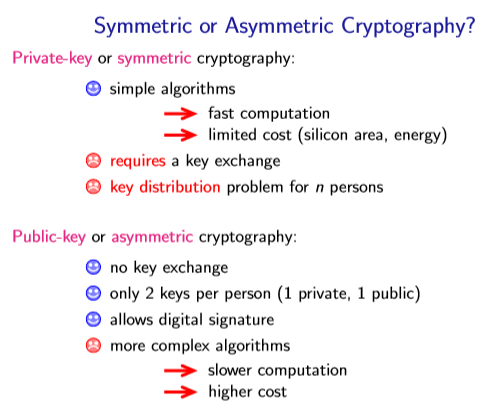
Cette solution ne passe pas à l’echelle.

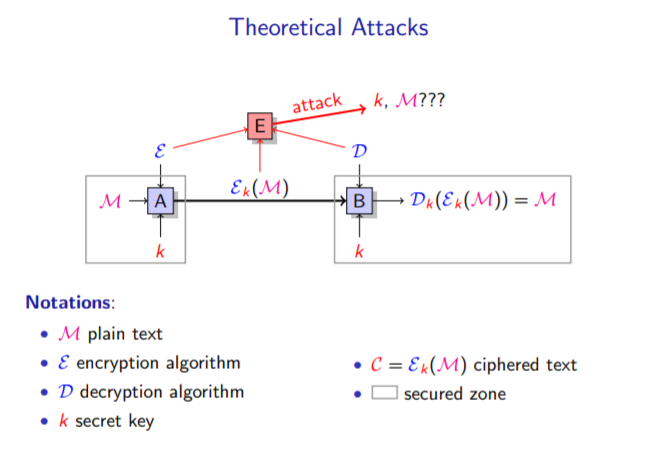


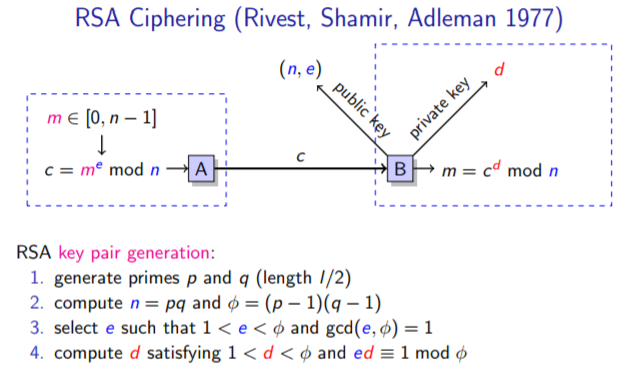
Alice va prendre la clé publique de Bob pour chiffrer. (Bob a préalablement afficher sa clé publique). Toute personne qui veut envoyer un message à Bob va passer par cette publique. Bob déchiffre grâce à sa clé privée.

**On chiffre avec la clé publique.**

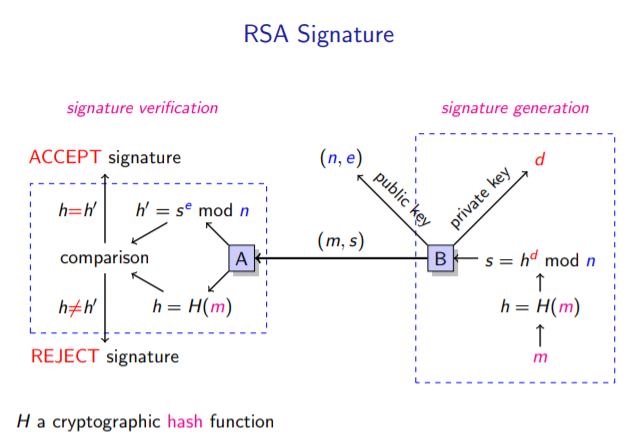
Tout le monde est capable d’envoyer à Bob grâce à la clé **k**. En revanche, seul Bob peut déchiffrer car personne ne peut décrypter le message tant qu’il n’a pas la clé privée (**k’**).







Gcd(e,phi) =1 🡪 On veut que e et phi soient premiers entre eux



Fonction de hachage : Opérations mathématiques qui font un résumé du message

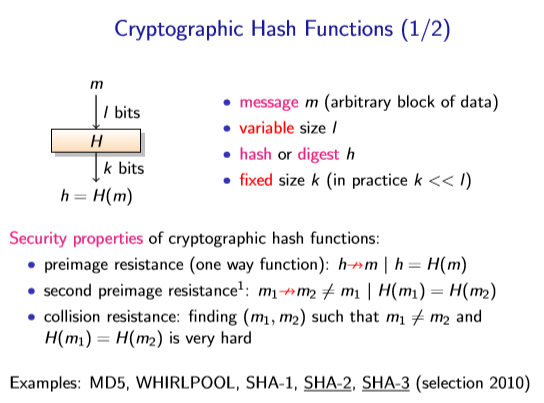
Quand je signe, j’utilise ma clé privée. Quand les gens reçoivent, ils utilisent la clé publique pour vérifier l’authentification.

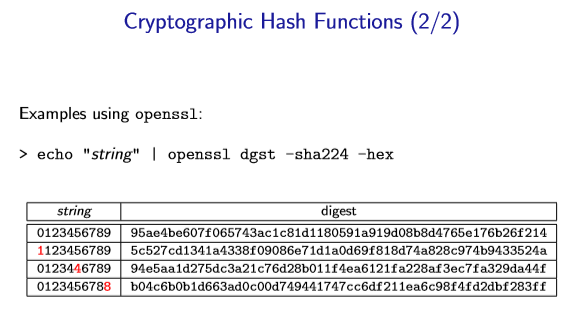
Alice veut envoyer un message chiffré à Bob. Et bob veut vérifier que c’est bien Alice qui envoie.

Alice chiffre le message avec la clé publique de Bob et en même temps elle veut pouvoir expliquer à Bob que c’est bien elle qui envoie le message.

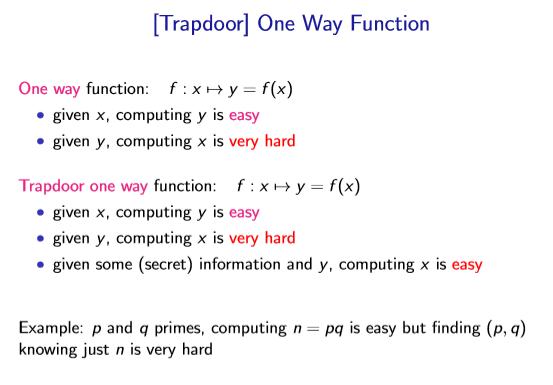
Bob sait bien déchiffrer le message grâce à sa clé privée.

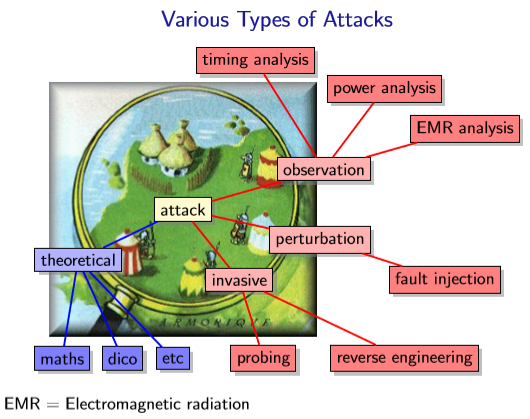
Pour s’identifier, Alice signe avec sa clé privée hachée. Bob vérifie la signature grâce à la clé publique d’Alice.



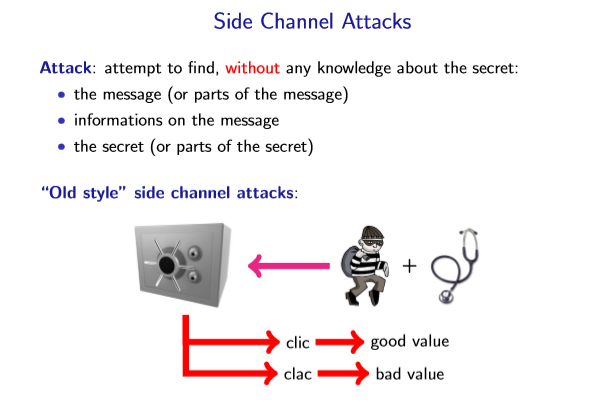


Avec seulement une variation dans la String, on peut voir que le hash est totalement différent.



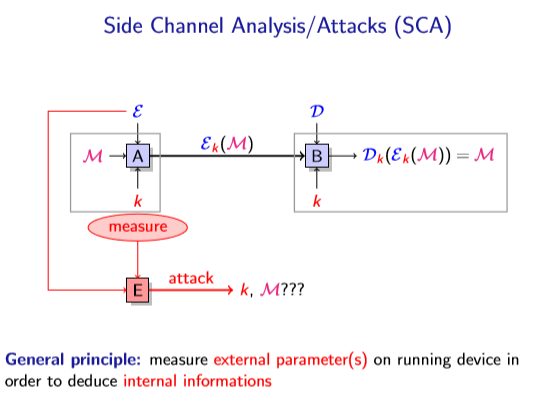


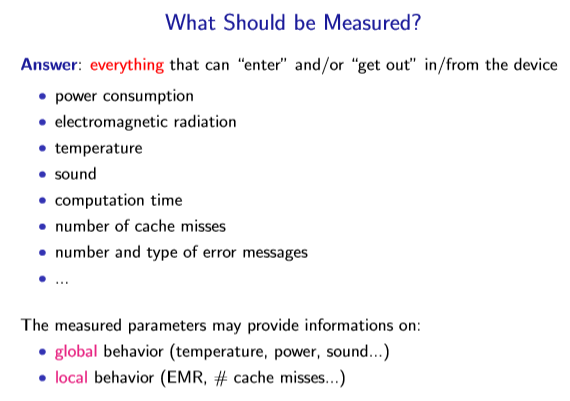
Observation : Ce qu’on peut observer dans une puce électronique.



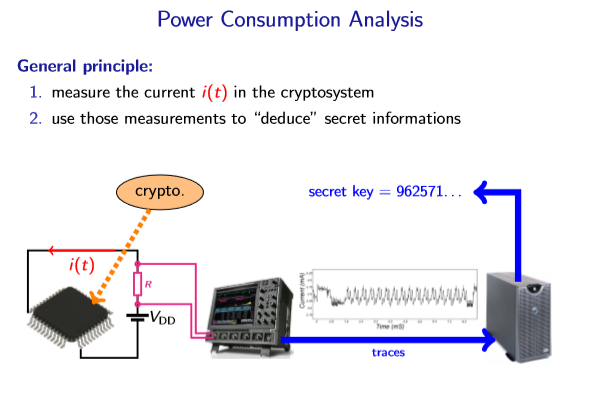
Attaque par canaux cachés : Par exemple, écoute au stéthoscope pour ouvrir un coffre-fort. Le canal caché est le son.

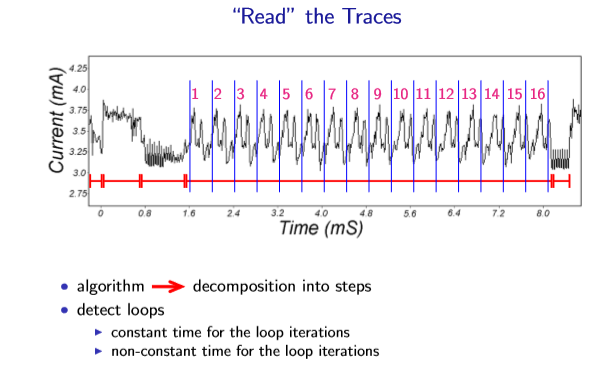
En électronique on écoute la puce à l’aide d’un oscilloscope.





On peut mesurer le courant, la température, le bruit (vibration des fils dans la puce).

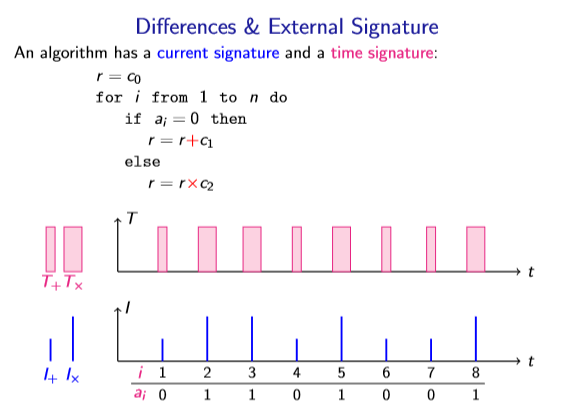




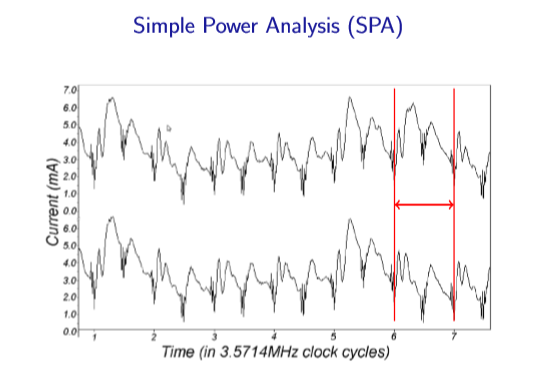
On envoie cette trace au serveur qui analyse et décrypte la clé.

On envoie un truc au début, ensuite, on voit un signal qui se répète. On a ici une boucle, avec 16 tours. L’algorithme qui possède 16 tours de boucles est DES.

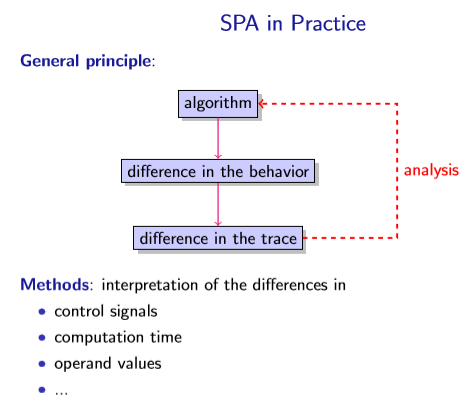
Faux algorithme de chiffrement simple :

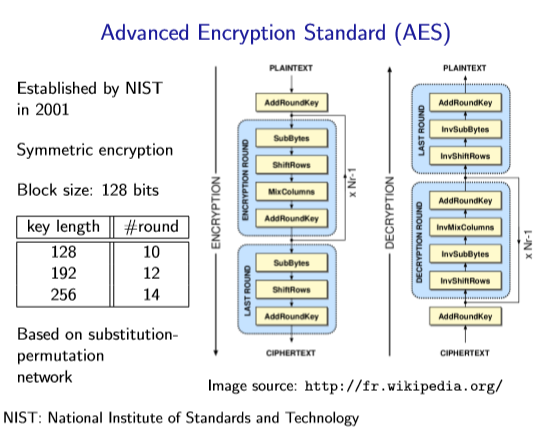


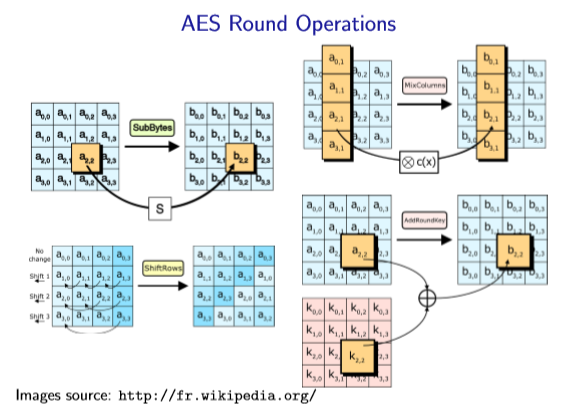
Quand j’aurai une multiplication (temps long et courant plus grand) et inverse pour l’addition.



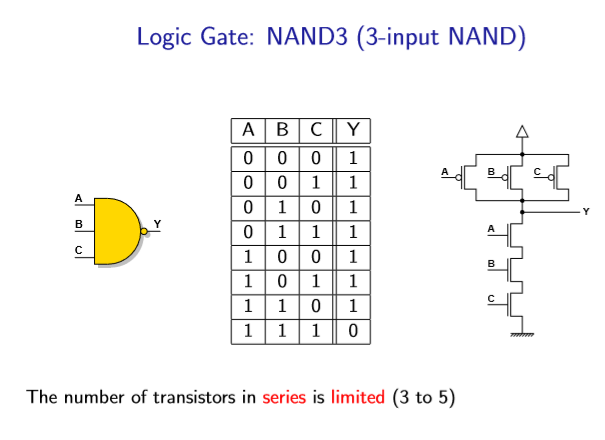
Dans le cadre rouge, il y a une modification. On peut détecter lequel est l’addition et lequel est la multiplication. En haut multiplication (bit secret = 1) et en bas addition (bit secret = 0).



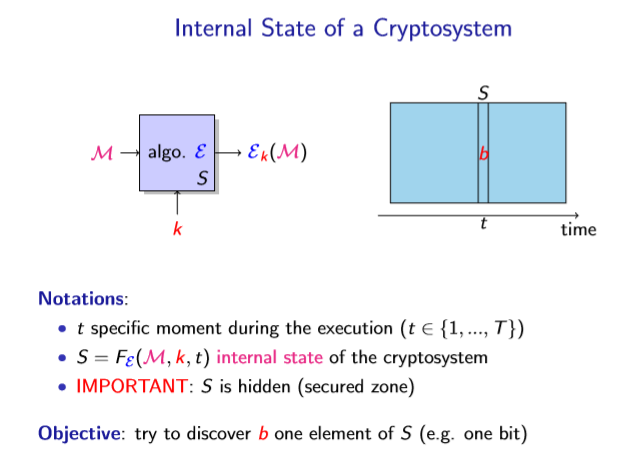


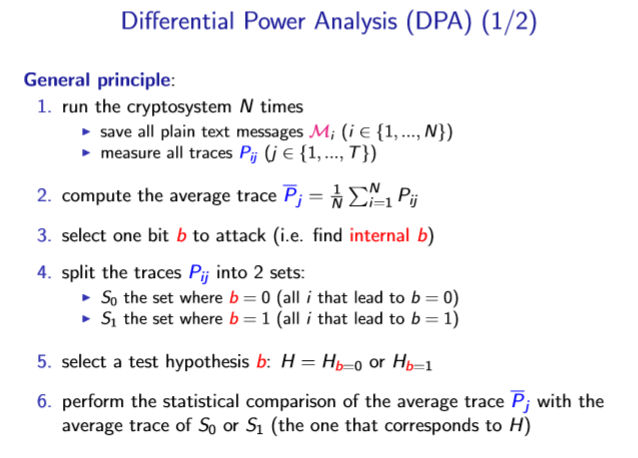


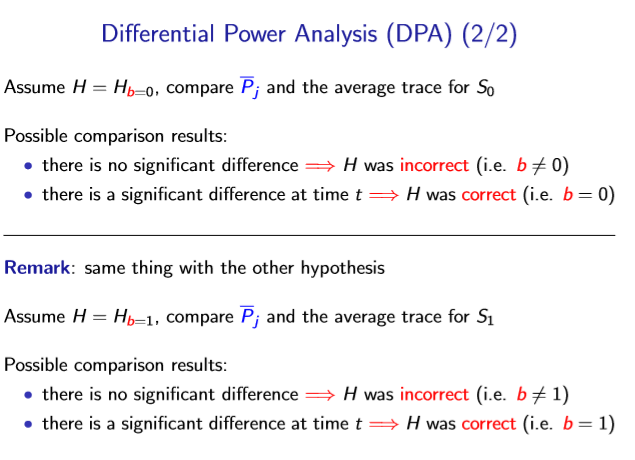
# Power Analysis and Cryptosystem Security: Attacks and Countermeasures

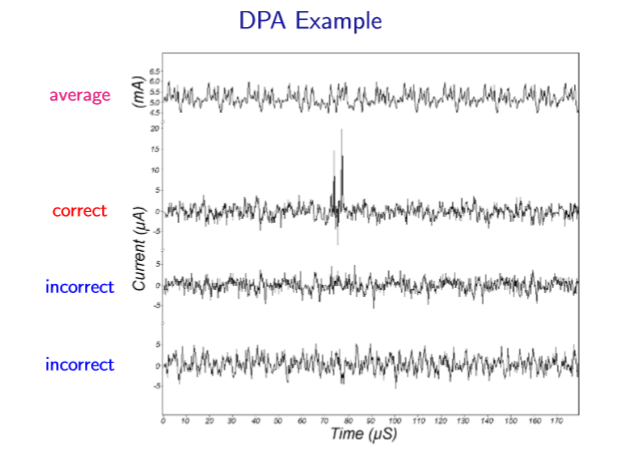


Le nombre est limité car on est limité en termes de tension de seuil.









Installer python

Installer numpy

Installer sage