Contents

SIDE CHANNEL ATTACKS-TER

TER encadré par Mr Mokrane

September 6, 2021

1 Introduction

Cette étude se porte sur les attaques par canaux cachés qu'on appelle aussi "side channel attacks (SCA)".

Actuellement, la notion de sécurité de l'information est très répandue. La cryptologie est le département des mathématiques qui traite les procédés cryptographiques mais aussi de la cryptanalyse. Grace à la participation des universitaires, de nombreux progrès sont faits à la fois au niveau algorithmique grâce à la cryptanalyse et au niveau matériel pour améliorer la sécurité des informations. La notion de cryptographie a largement fait évoluer le domaine de la protection des informations. Les données disponibles sur le réseau sont sécurisées de bout à bout. L'inviolabilité des algorithmes de chiffrement sur la sécurité des informations est assurée par un niveau de complexité qui garantit la structure mathématique solide et sure grâce aux études faites en cryptanalyse.
Tous les appareils connectés sur internet sont dotés d'un système qui assure la partie cryptographique des informations. On appelle par **crypto processeur**, un microprocesseur dédié aux taches cryptographiques de l'appareil où il est rattaché. Il garantit les implémentations des algorithmes cryptographiques comme celui du chiffrement RSA (Rivest Shamir Adleman Cryptosystem) ou du chiffrement AES (Advanced Encryption Standard). Les études en cybersécurité ont fait de grand progrès et ne cessent pas d'évoluer pour contrer les attaques susceptibles de passer par ce canal. De plus, grâce à la contribution des universitaires dans l'évolution des études en cryptanalyse, de nouvelles types attaques ont été détectés. L'information a été largement décrite dans l'article publié par Kocher.

A la fin des années 90, Paul Kocher et al. a développé pour la première fois une nouvelle forme d'attaque basée sur la consommation électrique du matériel lors d'une implémentation cryptographique. Depuis cet article, on définit par "attaques par canaux cachés", toutes attaques qui exploitent les fuites d'informations qui se révèlent au moment de l'implémentations des taches cryptographiques du crypto processeur. Ce type d'attaque ne se déroule donc pas sur le réseau mais est qualifié d'attaque sur le hardware ou attaque physique. Les attaques par canaux cachés sont des attaques qui exploitent la vulnérabilité de la partie physique (hardware) du système représenté par une fuite involontaire d'information cruciale lors des exécutions des microprocesseurs. Ces fuites sont caractérisées par la consommation de courant relevée et les émanations électromagnétiques recueillies. Ce type d'attaque est non-invasive et se contente de récupérer des mesures de l'activité du processeur pendant son activité.

2 Pré-requis:

2.1 Les attaques algorithmiques

Les attaques algorithmiques sont ceux qui se réfèrent à la cryptanalyse. Ce sont :

- attaque à chiffré seul : l'adversaire n'a que les entrées chiffrées pour essayer d'en déduire soit les entrées en clair soit la clé.
- attaque à clair connu : l'adversaire possède à la fois les chiffrés ainsi que les clairs correspondants. Il doit retrouver la clé pour le chiffrement. Utilisé et efficace sur le DES.
- attaque à clair choisi : l'attaquant choisit un nombre de texte clair et obtient les chiffrés correspondants. Cette attaque est plus efficace que les attaques à clair connus du fait que l'attaquant a la possibilité de choisir des clairs spécifiques, qui donnera plus d'informations sur la clé. L'algorithme AES est spécialement conçu pour résister à ce type d'attaque.
- attaque à chiffré choisi : l'attaquant à la possibilité de choisir différents chiffrés et dispose en plus d'une boite noir qui lui permet de les déchiffrer.

2.2 Les attaques physiques

Les attaques physiques sont les attaques qui exploitent la vulnérabilité d'un ou plusieurs composants électroniques et non la structure mathématique d'un algorithme. Les composants visés sont ceux qui opèrent directement avec l'algorithme. La sécurité apportée par l'architecture mathématique des

algorithmes ne suffit plus pour assurer la sécurité des informations lors du chiffrement. Ce sont les attaques communément appelés attaque par canaux cachés.
Ces attaques se basent essentiellement sur la connaissance de l'architecture du matériel et portent sur différent types de données :

- attaque temporelle : basée sur l'évaluation du temps mis pour certaines opérations.
- attaque par faute : consiste à l'introduction volontaire d'erreurs pour provoquer des comportements inhabituels du système.
- analyse par rayonnement électromagnétique : récupère et exploite le rayonnement émis par le système lors d'une opération cryptographique pour en déduire la clé.
- analyse de consommation : est le fondement de ce travail. Elle consiste à analyser la consommation de courant engendrés par les crypto processeurs lors d'une opération.

3 Notion de canaux cachés :

L'idée générale est de récupérer les courbes de consommation électriques d'un protocole cryptographique puis à l'aide d'un traitement statistique retrouver le secret utilisé pendant le calcul. Faire une hypothèse sur le secret utilisé puis vérifier si cette hypothèse est vraie ou non. Elle porte souvent sur la valeur d'un ou plusieurs bits du secret.
La partie physique ou hardware de notre ordinateur laisse fuir des informations lorsque le crypto processeur s'exécute c'est à dire lorsqu'il se met à implémenter les algorithmes cryptographiques comme le chiffrement/ déchiffrement à partir d'une clé. Un crypto processeur est un microprocesseur dédié aux taches cryptographiques comme le cas du calcul d'exponentiation dans un algorithme RSA.

On appelle par attaque physique, une attaque qui vise essentiellement un ou plusieurs composants hardwares dédiés aux taches cryptographiques du système. Ce type d'attaque va contourner les difficultés, du point de vue mathématiques ou algorithmique. Les attaquants ne passent plus par le réseau.

Les microprocesseurs, du fait de leur conception, donnent des informations sur leur comportement ainsi que sur les informations secrètes censés être protégés. Pour accroître le niveau de sécurité, les algorithmes de chiffrement utilisent une ASIC (Application Specific Integrated Circuit) ou une FPGA (Field Programmable Gate Array). Concrètement, ce sont des circuits intégrés composés de cellules programmables où chaque cellule est capable de réaliser une fonction. Les composants FPGA sont basés sur la technologie RAM qui consiste à stocker temporairement les fichiers que l'ordinateur exécute. Les crypto processeurs sont des semiconducteurs composés des portes logiques. Les électrons qui les traversent apportent ou libèrent une charge au niveau des portes des transistors en consommant de l'électricité et en produisant des radiations électromagnétiques. Par contre, la partie hardware sous-jacente reste vulnérable.
Pour mesurer la consommation électrique du circuit, il suffit de placer une résistance, petite de 50 ohms par exemple, en série avec le circuit.
On applique la formule:

$$I = U/R$$

où U est la tension du courant, R la résistance et I l'intensité du courant. C'est ce dernier qui nous intéresse.
L'exploitation des données électroniques nécessite l'utilisation d'un oscilloscope. On appelle "**trace**", le signal électrique mesuré au moment de l'enregistrement.

4 Métrique et technique de mesure

Le principe est le suivant, l'attaquant sollicite le crypto système plusieurs fois pour récupérer les informations que le crypto processeur va laisser fuir involontairement, à travers un support physique mesurable, puis l'attaquant récupère à chaque fois la variation des informations récoltées, c'est-ce qu'on appelle "**la fuite (leakage)**".
L'objectif est de récupérer un nombre important de traces pour pouvoir mener une attaque.

On a besoin:

- d'un oscilloscope avec une bande passante de plusieurs centaines de megahertz (MHz) pour couvrir tous les sous-systèmes,
- d'une résistance de quelques ohms (souvent 1 ohm pour faciliter le calcul) sur la branche d'alimentation du processeur pour récupérer la consommation de la puce.

4.1 Technique de mesures des canaux cachés par analyse de courant

Pour avoir une fuite plus pertinente, on utilise une sonde sensible au champ électromagnétique posé au plus près du silicium. L'enroulement d'un fil de cuivre de très faible section autour d'une fine pointe permet de récupérer des fuites pour remonter à une clé secrète.
La récupération de ces mesures par l'attaquant se déroule comme suit :

- l'attaquant propose le secret k* au dispositif physique
- il l'interroge N fois (en envoyant N messages) pour solliciter le crypto système et obtenir N fuites.
- obtient la trace
- procède à une attaque **probabiliste** sur la trace.

4.2 Cas concrets:

La fameuse attaque dite "pita" montre ainsi la récupération d'une clé rsa avec une simple radio logicielle (autour de 2 MHZ et avec une bande passante de 100 khz) posée à proximité du pc attaqué qui ne prend pas beaucoup de place et peut être camouflée par un pain pita.
Des attaques sur les ordinateurs ont été menées via le port USB ou même la prise électrique. Des attaques d'extraction de clés avec une très faible bande passante ont aussi été validées avec des canaux physiques comme le son et le potentiel du châssis.

De ces fuites ont été introduites plusieurs attaques.

4.3 Modèles de fuites

La notion de Side channel attacks SCA regroupe plusieurs méthodes d'approches dans sa réalisation. En effet, on a vu précédemment qu'il est possible de brancher un oscilloscope sur le circuit et on obtient la fuite "physique" de la cible. On exploite directement ces mesures pour en extraire la clé. C'est la méthode du **Simple Power Analysis SPA**".

Une autre approche consiste à extraire la valeur de la clé en comparant les enregistrements et les valeurs issue du modèle de prédiction en utilisant un outil statistique qu'on appelle "**distingueur**". Plusieurs distingueurs sont proposés. Un distingueur optimal représente celui qui donne un résultat maximal et défini la probabilité du succès.

Les distingueurs sont les éléments constitutifs principaux du SCA car ils permettent de comparer les mesures récupérées au modèle de prédiction établie.

4.4 Notation:

Soit la fuite X un ensemble de variables aléatoire de dimension Q le nombre de mesures, D le nombre d'échantillons mesurés, R un ensemble de variables aléatoires, t l'ensemble des inputs pour la mesure de X. Soit k^* la clé secrète. On a un mot de n bits qui appartient à $\mathbb{F}_2{}^n$, alors k^* et t $\in \mathrm{F}_2{}^n$.

4.5 Définition: variable sensible

Une variable sensible est la variable traitée par l'algorithme cryptographique dont dépend les entrées et n'est pas connu de l'attaquant. C'est le cas de la clé secrète.

4.6 Définition: la fonction de sélection

Soit g la fonction qui part des données introduit vers une variable sensible:

$$g : \mathrm{F}_2{}^n \times \mathbb{F}_2{}^n \times \mathbb{R} \longrightarrow (\mathbb{F}_2{}^n)^D$$

$$k*,T,R \longrightarrow g(k^*, T, R)$$

4.7 Définition: la fonction de fuite

La tendance de la fuite des valeurs sensibles dépend de la fonction de fuite qui est une des spécificités de la cible :

$$\Psi : (\mathbb{F}_2{}^n)^D \longrightarrow \mathbb{R}^D$$

$$V \longrightarrow \Psi(V)$$

tel que N est le bruit indépendant entre chaque mesures. La fonction g est fonction de l'algorithme utilisée et la fonction Ψ est une caractéristique du matériel ciblé. Elle n'est pas connu de l'attaquant, de ce fait, ce dernier va sélectionner les modèles de fuite pour faire la prédiction de la fonction de fuite. Soit $\widehat{\Psi}$ le modèle de la fuite tel que : $\widehat{\Psi} : (\mathrm{F}_2{}^n)^D \longrightarrow \mathbb{R}^D$.

On pose $y = \widehat{\Psi} \circ g$, où $\circ$ est la loi de composition. On défini le modèle de prédiction par :

$$Y_k = y(k, T, R) = \widehat{\Psi}(g(k, T, R))$$

Le modèle de prédiction défini la relation entre la modèle de fuite et la prédiction de la clé k pour avoir une estimation des valeurs possibles de la fuite.

4.8 Mesures et quantification des informations

- Signal Noise ratio SNR: Une fois les traces récupérées, pour comparer les données, on utilisera une métrique de comparaison des signaux qui est le rapport signal-bruit, noté **SNR signal noise ratio**. Elle représente le rapport entre l'information pertinente et l'information non significative. Elle permet de quantifier la fuite.

- Entropie : La notion d'**entropie** est aussi utilisée pour caractériser la fuite d'information à partir de dispositif cryptographique. La notion d'entropie a été introduit par Shannon pour mesurer la fiabilité de l'incertitude associé à une variable aléatoire. L'une des raisons pour la quantification de l'information est de mesurer la qualité du circuit face à une fonction de fuite de donnée à l'aide de l'entropie conditionnelle. Ensuite, il est nécessaire de mesurer comment cette information fuitée est utilisée pour réussir à retrouver la clé.

- Entropie estimée: On appelle **entropie estimée**, la moyenne du classement de la clé pendant les attaques par canaux cachés. En effet, comme les attaques sca se basent généralement sur des suppositions exhaustives sur les clés, c'est une métrique pertinente car elle permet de voir au fur et à mesure le comportement de l'attaquant et permet de savoir si la clé recherchée remonte dans le classement ou non face à des traces bruitées.

- Le taux de succès : Le **taux de succès** est estimé par un adversaire, en calculant le nombre de fois que l'attaque est réussie en fonction des données utilisées. Le taux de succès SR représente donc la force de l'adversaire et par conséquent définie la robustesse du dispositif cryptographique attaqué.

5 Différents types d'attaques SCA:

Étant donné que selon les valeurs manipulées par chaque attaque, tous agissent selon le même principe que chacun compare le modèle de prédiction fourni avec les mesures recueillies. Ils vérifient donc la propriété statistique qui les définissent en tant que **distingueurs**. Ils peuvent être classés en plusieurs catégories en fonction des informations dont ils auront besoins.

Les différents types d'attaques SCA que nous allons voir sont donc ceux que l'on peut retrouver selon une approche dite **probabiliste**. Ce sont:

1. le Simple Side Channel Analysis: Simple Power Analysis SPA,
2. les Profiled Attacks (attaques profilés) : les attaques Templates,
3. les "non-profiled attacks": Correlation Power Analysis CPA, Differential Power Analysis DPA.

5.1 La "Simple Power Analysis (SPA)"

La SPA fait partie des distingueurs qu'on appelle Simple Side Channel Analysis (SSCA) qui est capable de déduire la clé secrète en observant les traces recueillies. La SPA exploite la dépendance qui existe entre la clé et l'évolution de la consommation électrique à cause du comportement du processeur.
La consommation d'électricité dépend largement du type des données présentés en entrées (inputs). En cryptographie, les inputs considérés seront les "textes clairs" et les "textes chiffrés" selon le mode d'attaque choisi par l'attaquant. Les fuites sont dues aux différents composants du système qui sont les mémoires, les bus, ainsi que les processeurs. Le "simple power analysis" exploite la trace laissée par la consommation de courant lors d'une opération. A partir de cette trace , l'attaquant est capable d'identifier l'opération en cours d'exécution comme le cas d'une exponentiation par exemple. En combinant les informations recueillies avec la connaissance de l'algorithme, l'attaquant extrait la valeur de la clé secrète. En effet, lors d'une opération, si un chemin ("branch") de l'algorithme est parcouru à certain moment quand le bit de la clé secrète est égale à zéro 0, alors il suffit à l'attaquant d'interpréter les traces de consommation collectées obtenu lors du "simple power analysis " pour déduire la clé secrète.
La SPA est utilisée contre les algorithmes asymétriques comme le chiffrement RSA car il utilise une étape d'exponentiation modulaire distinguable sur les traces obtenues. Elle effectue une opération de mise au carré à chaque itération suivie d'une multiplication seulement si le bit d'exposant égal à 1. L'exposant d, qui est la *clé privée*, peut être compromis si le profil de consommation de l'opération de mise au carré est différente de celle incluant en plus la multiplication. On peut lire directement sur la trace la clé privée.
De manière générale, la SPA sera efficace à chaque fois que le calcul implique des opérations dépendantes de la clé et que l'on peut différencier la consommation de ces opérations.

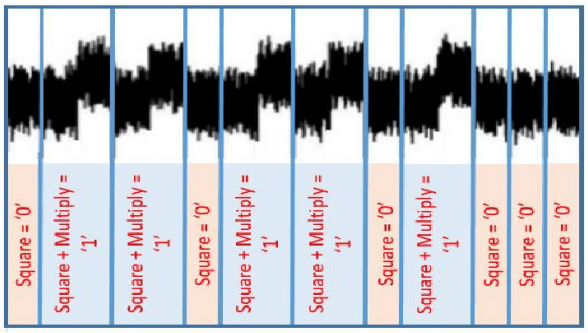

Figure 1: portion de trace d'une exponentiation RSA

Le **modèle de consommation** est basé sur un modèle de poids de Hamming (HW) qui est le nombre de bits 1 présents dans un mot donné. Dans un microprocesseur m bits, les données D codées en binaire sont représentées par

$$D = \sum d_j 2^j$$

où $d_j = 0$ ou 1.
Son poids de Hamming est $H(D) = \sum d_j$ qui est donc le nombre de "1" dans l'information. Sa valeur est comprise entre 0 et m. Si D contient m bits uniformément distribué et indépendant, alors tout le mot a :

- une moyenne de HW : $\mu_H = \frac{m}{2}$
- une variance de $(\sigma_H)^2 = \frac{m}{4}$

La fuite de données capturée par un SPA dépend du nombre de bits qui est passé d'un *état* à un autre en un temps donné. Cela se passe au niveau des portes logiques que l'on retrouve dans la technologie CMOS (Complementary Metal Oxide SemiConductor), des circuits logiques. Le courant consommé est l'énergie nécessaire pour passer le bit d'un état à un autre. On suppose que passer de "1" à "0" nécessite la même quantité dans les deux sens.
Soit R l'*état de référence* du bit qui n'est pas nécessairement zéro. On note, $H(D \bigoplus R)$ le nombre de bits qui est passé de l'état R à D. C'est la **distance de Hamming** entre D et R. Alors, si $R = 0$ et si D une variable aléatoire uniforme, $D \bigoplus R$ et $H(D \bigoplus R)$ ont la même moyenne $\mu_H = \frac{m}{2}$ et la variance $H(D) = \frac{m}{4}$.
On a alors la relation :

$$W = aH(D \bigoplus R) + b$$

où :

- a est un scalaire qui représente le gain défini par le poids de Hamming H et la consommation de courant W.
- b est une variable dépendante du temps et du bruit au moment de l'écoute.

5.1.1 Contre mesures

Pour dissimuler le comportement des traces en fonction de la position de l'implémentation au niveau de l'algorithme, on:

- peut agir au niveau du code: restructurer le code en éliminant les chemins des conditions qui révèlent les informations sur la clé.
- peut agir sur la trace: introduire aléatoirement des bruits dans les mesures de la consommation de courant.

Ces contre-mesures sont inefficaces contre d'autre type de SPA plus perfectionnées.

5.2 Differential Power Analysis DPA

Le "Differential power analysis DPA" est une approche de SPA d'efficacité plus élevée. Cette attaque exploite les courbes de consommation de plusieurs sollicitations de la part de l'attaquant et étudie par traitement statistique la variation de la dépendance entre les données et la consommation d'énergie résultante pour casser la clé utilisée pendant le calcul.
La DPA est une SPA qui récupère les informations à la sortie (output) et crée le différentiel en complément de l'observation des traces.
La DPA a été développé par Paul Kocher et al. dans son article publié en 1999. L'attaquant récupère des informations sur les données en entrées (inputs) et en sortie (outputs) lors de l'exécution du processeur en plus des informations sur la consommation de ce dernier en électricité ainsi que les radiations électromagnétiques qu'il dégage. La DPA est une attaque qui permet d'obtenir des informations sur la clé secrète par *analyse statistique* des données de consommation de courant mesuré grâce à un grand nombre de sollicitations avec la même clé. En effet, la dpa exploite le fait que la consommation en courant du processeur lors d'une opération dépend de la donnée exécutée.
Ces attaques sont appliquées en particulier lorsque les attaques simples ont échoué, c'est à dire lorsque les informations filtrées ne peuvent être directement liées au secret que l'attaquant veut déterminer. L'attaquant doit être en mesure de simuler plusieurs fois l'exécution d'un algorithme et si possible connaître la structure interne de l'algorithme (scare sur sécurité).
Les calculs exécutés par l'algorithme se traduisent matériellement par des changements d'états des portes logiques. Par exemple, en technologie CMOS, le changement d'état d'une porte entraîne la charge ou la décharge électrique des transistors pouvant être considérés comme des capacités. La variation de la consommation du circuit est donc la conséquence des changements d'états de l'ensemble des portes logiques pendant l'exécution de l'algorithme.
Pour effectuer une attaque DPA, l'attaquant:

- identifie une partie de la clé dite "gérable". L'analyse statistique devra être faite pour toute valeur que peut prendre cette partie gérable.
- pour chaque opération de chiffrement i, enregistrer la consommation instantanée du dispositif sous forme d'un vecteur V_i.

L'attaque se déroule généralement en deux phases :

- l'acquisition des données
- l'exploitation de ces données.

Etant donné que l'attaquant optimise au maximum son champs d'attaque, il agit uniquement sur les **points critiques** ou "target node". C'est l'instant où il prend en compte les variations de la courbe de consommation électrique. Il s'agit du moment où il y a une interaction entre l'opération avec la clé secrète, ou un morceau de celle-ci, et le message connu.
Concrètement, pour des entrées P et Q connues, l'attaquant sait quel sera le comportement du point critique. Il émet une hypothèse sur la valeur d'un bit de P, et fourni plusieurs entrées différentes connues Q.
En effet, il sollicite plusieurs fois l'algorithme avec les entrées choisies. Il récupère les courbes de consommation de chaque exécution de l'algorithme en supposant que le même secret P est utilisé à chaque fois. La consommation de courant dépend de la valeur de P et des entrées Q. En connaissant le protocole utilisé ainsi que les algorithmes de calcul, il est possible de prévoir un résultat à l'aide de l'entrée connue et de l'hypothèse faite sur le secret.
L'ensemble des courbes de consommation du circuit est divisé en deux sous ensemble S_a et S_b. D'un côté, nous mettons les courbes pour lesquelles les transitions consomment et de l'autre non. Ces sont les vecteurs de S_a et S_b. On représente la consommation électrique du circuit soumis à différentes entrées connues. Si l'hypothèse faite sur la clé secrète est la bonne, alors les simulations sont menées en utilisant le secret.

Il est important de noter que l'attaque DPA est qualifiée du plus impressionnante et du plus difficile à éviter. En effet, c'est une attaque impressionnante car l'attaquant n'a pas besoin de connaître ni la consommation en courant du processeur ni sa position en temps lors de l'implémentation de l'algorithme.

5.2.1 Hypothèse fondamentale du DPA

Il existe une valeur primordiale qu'on appelle **valeur intermédiaire** qui apparaît lors de l'implémentation de l'algorithme tel qu'en connaissant quelques bits de la clé, en générale moins de 32 bits, il est possible de savoir si deux inputs (ou deux outputs) donne ou non la même valeur pour ce variable. De ce fait, tout algorithme qui utilise une SBox est vulnérable à la DPA à cause de leur implémentation "naturelle" qui repose sur l'hypothèse ci-dessus.

5.2.2 Étude de cas sur un DES

Dans un DES, on sait que:

- le nombre de tours est 16.
- à chaque tour ou "round", il existe une transformation F sur 32 bits c'est-à-dire 8 transformations non linéaire de 6 bits vers 4 bits selon la table de la SBox
- au total, il y a 8 SBox.

L'attaquant va intervenir au niveau des SBox puis déterminer la clé entière de façon exhaustive.

1. Soient $E_1, ..., E_{1000}$ les 1000 entrées, $C_1, ..., C_{1000}$ les 1000 courbes relatives aux entrées et MC la moyenne des 1000 courbes.
2. l'attaquant analyse le premier bit à la sortie du premier SBox au premier tour. Soit b cette valeur sur 6 bits. On a alors $b \star \star \star \star \star$, soit $2^6 = 64$ combinaisons et donc 64 courbes. Il prend chaque bloc de 6 bits de b qu'il combine avec $E_1, ..., E_{1000}$. Puis créer deux ensembles pour $b = 0$ et $b = 1$. Les sorties obtenues seront classées selon la valeur de b.
3. il calcule la moyenne des $b = 0$ qui est MC'. Si MC et MC' représentent une différence appréciable de point de vue statistique, alors on a une valeur *correcte* des 6 bits. Sinon, il refait l'étape 2 avec un autre choix pour les 6 bits.
4. il répète l'étape 2 et l'étape 3 au deuxième SBox, puis troisième, ... jusqu'au huitième SBox pour obtenir les 48 bits de la clé secrète.
5. les 8 derniers bits seront retrouvés de façon exhaustive.

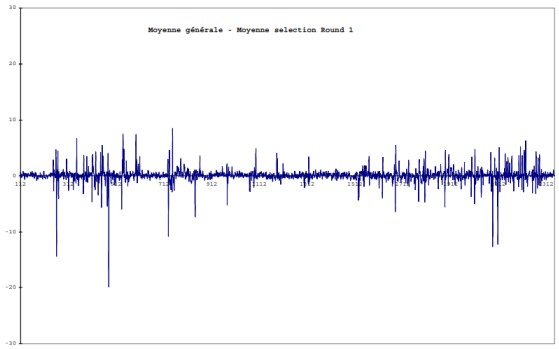

Figure 2: DPA avec différence entre MC et MC'non significative, 6 bits fausses

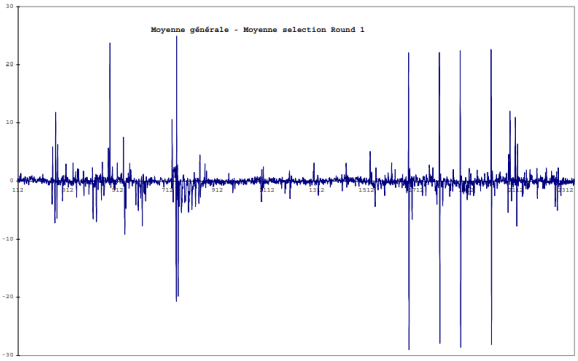

Figure 3: DPA avec différence entre MC et MC'significative, 6 bits corrects

5.2.3 Contre-mesures de la DPA:

Ce sont:

- introduire un "random timing shifts", de telle sorte que les moyennes ne correspondent pas à la consommation de la même instruction.
- on remplace les instructions crucial de l'algorithme (lecture, écriture,...) par des instructions dont les images de la consommation sera difficile à analyser.
- pour un algorithme donné, on propose une façon explicite de l'implémenter de tel sorte qu'il est possible de prouver qu'il est *DPArésistant* en agissant sur le calcul non-linéaire du SBOX.
- on rend l'hypothèse ci-dessus fausse. Car selon l'hypothèse:
  - une variable intermédiaire V qui apparaît lors du calcul et dépends des inputs (ou outputs) qu'il a en sa possession.
  - k variables $v_1, ..., v_k$ tel que $v_1, ..., v_k$ nous permet de retrouver V.

  Pour cela, il suffit à l'attaquant de choisir une fonction f qui satisfait l'identité

  $$V = f(v_1, ..., v_k)$$

  en respectant les deux conditions suivantes:

  1. A partir de la connaissance la valeur V et pour une valeur fixée i, $1 \leqq i \leqq k$, il est impossible de déduire une quelconque information sur l'ensemble des valeurs v_i, tel qu'il existe un $(k-1) - uple$ défini par $(v_1, ..., v_{i-1}, v_{i+1}, ..., v_k)$ qui satisfait $f(v_1, ..., v_k) = V$.
  2. La fonction f est telle que les transformations sur $v_1, ..., v_k$ durant l'implémentation est possible sans calculer V

Durant une attaque DPA, le point critique est choisi de telle sorte qu'il dépend d'une petite partie de la clé secrète seulement. Ainsi, les hypothèses sur le bout de clé peuvent être faites de façon exhaustive. Chacune des 8 courbes représente en fonction du temps la différence entre la moyenne de consommation électrique pour la transition cible du paquet Sa et Sb, pour une hypothèse de clé.

Il existe plusieurs façons de sélectionner la courbe correspondant à la bonne hypothèse de clé. Nous pouvons choisir la courbe présentant le plus grand pic de consommation ou encore celle dont l'aire sous la courbe est la plus grande.

5.3 L'analyse de courant par corrélation ou Correlation Power Analysis CPA

5.3.1 Généralités

En 2004, l'attaque DPA a été améliorée par des chercheurs, Micali et Reyzin, de la société Gemalto qui propose de calculer directement la corrélation entre la consommation de puissance et le poids de Hamming en sortie du modèle pour chacune des sous-clés. C'est l'attaque par corrélation qui utilise le coefficient de Pearson pour le calcul de la corrélation. Il est défini par :

$$C(M, P_j) = \frac{(\mu(M * P_j) - ((\mu(M) * \mu(P_j))}{\sqrt{(\sigma(M))^2 * (\sigma(P_j))^2}}$$

où M est un vecteur contenant N traces de puissance P_tri, P_j est une colonne de la matrice P (de taille $N * 256$) des poids de Hamming de la sortie du modèle pour chacun des N textes en clair et pour les 256 sous clés possibles K_j, $\mu(x)$ est la fonction moyenne et $\sigma(x)$ est la fonction variance.

$C(M, P_j)$ est donc la corrélation entre le vecteur de mesure M et le vecteur des poids de Hamming P_j calculé pour la sous−clé K_j. Le raisonnement par corrélation est fondé sur la dépendance entre la consommation en courant du circuit et la distance de Hamming des données manipulées.
L'attaque CPA consiste à analyser la corrélation qui existe en évaluant le **coefficient de Pearson**. C'est un outil statistique qui détermine le degré de dépendance linéaire entre ces deux variables. La dépendance est linéaire car on suppose que les fuites de données à travers un canal dépendent du nombre de bits de commutation d'un état à un autre à un instant t, et ainsi celle qui maximise les corrélations sera la clé correcte.
Les modèles sur la distance de Hamming est obtenue grâce à des hypothèses que l'attaquant fait sur la partie "gérable" de la clé qu'il a déterminé préalablement. Puis, on remarque que (w, h) vérifie l'inégalité de Cauchy-Schwarz : $-1 \leqq \rho \leqq 1$. La clé correcte est obtenue aux extremums +1 ou -1 pour la bonne hypothèse de la clé. Une valeur égale à zéro signifie que les variables ne sont dépendant linéairement entre elles. La relation entre la consommation en courant et la *distance de Hamming* est linéaire, par conséquent la clé correcte est celle qui maximise les corrélations entre elles. Le facteur de corrélation de Pearson en fonction de la distance de Hamming H est définie par:

$$\rho(W, H) = \frac{cov(W, H}{\rho(W) * \rho(H)}$$

avec

$$-1 \leqq \rho(W, H) \leqq +1$$

selon Cauchy-Schwarz. Les hypothèses pour l'utilisation de la distance de Hamming sont faites sur la "partie gérable" de la clé. La clé correcte est obtenue lorsque ce facteur tend vers +1 ou −1 pour la bonne *hypothèse* de clé. Une valeur égale à Zéro montre que les variables ne sont pas linéairement liées entre elles.

5.3.2 Approche spectrale pour accélérer la CPA

Une nouvelle méthode de calcul de l'attaque CPA qui repose sur l'analyse spectrale est introduit par Guillot et al. en 2017.

Son intérêt est la réduction de la complexité de calcul qui est égale à $O(ND)$ au lieu de $O(N2^nD)$ lorsque $N >> n2^n$, avec n la taille des registres en bits du dispositif cryptographique, N le nombre de messages requis et D le nombre d'échantillon temporelle par mesure.

Il est possible de baisser la complexité de calcul du coefficient de Pearson en utilisant la propriété d'**égalité des images sous les différentes sous clés (EIS equal images under different subkeys**) définie par:
"Soit V désignant un ensemble arbitraire et soit $G : \{0,1\}^n \times \{0,1\}^{n'} \longrightarrow V$ une fonction, telle que pour chaque sous clé k, toutes les images $G(\{0,1\}^n \times \{k\}) \subseteq V$ soient égales. Nous disons que la fonction F a la propriété "Images égales sous les différentes sous clé (EIS)", s'il existe une fonction $\widetilde{F} : V \to \mathbb{R}$ telle que $F = \widetilde{F} \circ G$".

C'est une approche qui combine l'utilisation du coefficient de Pearson avec la définition de l'EIS. Son intérêt, au lieu de calculer le coefficient de Pearson pour chaque sous clé k séparément, un seul calcul suffit, en tirant profit du produit de convolution. On appelle par **produit de convolution**, un produit de fonctions $L \times M$ des coordonnées. En complément de sa propriété par rapport à la transformée de *Walsh-Hadamard*, il est possible d'évaluer le produit de convolution pour toutes les valeurs possibles k en une seule fois, avec une complexité globale de $O(n2^n)$ additions au lieu de $O(2^{2n})$ multiplications. La propriété de **Walsh-Hadamard** par rapport au produit de convolution est définie par :

$$L \bigotimes M(k) = WHT^{-1}(WHT(L) \bigotimes WHT(M))(k)$$

où WHT est la transformée de Walsh-Hadamard, sui est la transformée de Fourier sur le groupe $(\mathbb{S}, \bigotimes)$ un groupe de taille 2^n, $k \in \mathbb{S}$, n la taille de la clé.

5.4 Les attaques "templates"

5.4.1 Généralités

C'est l'attaque par canaux cachés jugés le plus puissant publiés à ce jour. L'attaquant dispose d'un clone du dispositif attaqué, à la différence que ce clone ne contient pas de valeur secrète, dont il a la maîtrise totale. De ce fait, comme l'adversaire dispose d'un matériel programmable à sa disposition, d'apprentissage, il pourra obtenir plusieurs modèles de référence qu'on appelle des **Templates**.
L'attaquant procède en deux phases :

1. **Phase d'entraînement** : Le but est de recueillir des données de chiffrement de messages aléatoires combinés aux enregistrements des consommation électrique du dispositif lors du chiffrement. En effet, l'attaquant choisit la partie gérable de la clé et, pour chaque valeur possible K_i de cette partie gérable, il chiffre un grand nombre de messages aléatoires et récupère la consommation électrique en même temps. Une fois ces informations réunis, par un *traitement statistique*, il va identifier les caractéristiques de consommation dépendant de la valeur de K_i.

2. **Phase d'attaque en ligne** : Le but est de comparer les informations recueillies au préalable sur la machine clone avec des messages aléatoires chiffrés par la clé secrète K sur la machine à attaquer. L'attaquant va solliciter le dispositif en lui envoyant plusieurs messages à chiffrer et enregistre en même temps la consommation électrique. Une fois enregistré, il récupère les données transmis par le clone et les compare avec ceux issues du dispositif à attaquer pour en déterminer quel groupe de traces se *ressemblent*. Cela lui permet de déterminer une partie de la clé secrète K. Grâce à son clone, l'attaquant peut avoir autant de données qu'il le souhaite pour reconnaître les ensembles de sous clés qui recouvrent la totalité de K.

Lors de la première étape, sur le clone, l'attaquant peut exécuter une même information pour différentes valeurs de bits de la clé, et obtient ainsi un ensemble de possibilité. Puis, lors de la deuxième étape, il utilise ces templates pour classifier une trace T_k prise le dispositif ciblé en limitant le choix des bits de la clé K. Finalement, il peut ainsi en déduire quelle valeur correspondant à quelle opération exécutée. Il procède de la même façon au fur et à mesure pour combler les bits manquants qui constituent la clé secrète K.

5.4.2 Un modèle probabiliste

Grâce à la méthode du maximum de vraisemblance, l'attaquant peut sélectionner la trace où l'opération pour laquelle la *probabilité est maximale*.
L'analyse probabiliste et le choix de travailler avec une classification nécessite un modèle statistique utilisable dans la théorie de la décision. Cette analyse repose sur **la règle de Bayes** qui permet de

tester des probabilités "a posteriori" c'est-à-dire après l'observation concrète de certaines évènements, connaissant les distributions de probabilité conditionnelles "a priori" c'est-à-dire indépendamment de n'importe quelle contrainte sur les variables observées.
Ainsi, définir un modèle probabiliste pour un échantillon, c'est décider que les données pourraient être issues de la simulation d'une certaine loi de probabilité. On admet le fait que les données observées sont des réalisations de variables aléatoires sur lesquelles de théories de probabilités peuvent être définies pour en extraire les données donc *de présenter une prédiction ou d'apporter une décision.*

Remarque:
Quand on recueille des données, un échantillon recueilli diffère toujours du précédent à cause du bruit et de l'environnement au moment de l'enregistrement. On dit qu'**un échantillon est égale à un autre** lorsque les caractéristiques statistiques de chaque échantillon (moyenne, variance, quantile, ...) prennent des valeurs sensiblement égales. D'où l'utilité de travailler avec des *variables aléatoires*, qui sont des entités réelles observées par le sondage d'évènements élémentaires. En effet, en mathématiques, il est possible de faire une correspondance entre l'espace des événements Omega ω et l'ensemble des réels $\mathbb{R}$. Une variable aléatoire est vue comme une fonction $X : \omega \to \mathbb{R}$ où X multidimensionnelles ou vectorielles de dimension d car les signaux sont considérés de dimension d, $X \in \mathbb{R}^d$.
Il est évident que la représentation de chaque classe de données dépendra principalement de la loi de distribution statistique qu'on va utiliser. Il est d'usage d'adopter *la loi normale* du fait que sa fonction de densité est symétrique et décroissante pour approcher un échantillon autour d'un point. Et aussi, elle peut être généraliser avec plusieurs dimensions en remplaçant la moyenne μ par un vecteur et la variance par **une matrice de covariance**C. Grâce aux propriétés de la matrice de covariance, on peut déduire que la variable aléatoire X présente une dépendance linéaire (matrice singulière, une matrice carrée non inversible) et aussi sélectionner les composants les moins bruités (matrice positive avec ses plusieurs conditions réunis en même temps).

5.4.3 La méthode vraisemblance

La **vraisemblance** est définie pour tout vecteur $(x_1, ..., x_n)$ d'éléments de E ,un ensemble fini, et pour une valeur θ associé à la famille de loi de probabilité $\mathcal{L}$, par l'équation :

$$L(x_1, ..., x_n, \theta) = \prod \mathcal{L}(x_i)$$

. Soit un échantillon $(X_1, ..., X_n) \in \mathcal{L}$ des variables aléatoires indépendantes et de même loi de probabilité, la réalisation observée $(x_1, ..., x_n)$ des X_i est :

$$P[(X_1, ..., X_n) = (x_1, ..., x_n)] = L(x_1, ..., x_n, \theta)$$

. Estimer un paramètre d'une loi par la méthode de maximum de vraisemblance revient à prendre la valeur de ce paramètre $\widehat{\theta}$ qui maximise la vraisemblance:

$$\widehat{\theta} = \tau(x_1, ..., x_n) = \arg(x_1, ..., x_n, \theta)$$

où $\tau(X_1, ..., X_n)$ est l'estimateur de vraisemblance de la variable aléatoire X_i. C'est la méthode de décision probabiliste qui permet de choisir la clé la plus pertinente parmi les prédictions.

5.4.4 Attaque template - Recherche de points d'intérêt

Ce sont des points qui apportent une quantité d'information significative qui sont sélectionnés sur lesquels vont se dérouler les tests. Une méthode utilisée est celle où l'on calcule la somme des carrés des différences entre les traces moyennes des templates et où l'on récupère les extremums de la courbe. L'attaque se déroule comme suit:

5.4.5 Phase d'apprentissage

1. L'attaquant recueille un grand nombre L d'échantillons sur le clone puis calcule les signaux moyens M_i pour chaque opérations.

2. Il calcule les carrés des différences entre tous les signaux moyens M_i pour sélectionner N points d'intérêt qui sont ceux où l'on observe une différence significative.

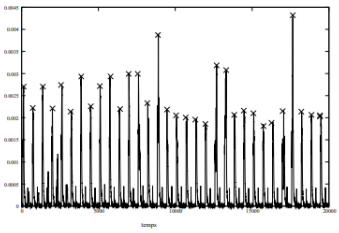

Figure 4: Exemple de points d'intérêt

3. Finalement, il calcule les matrices de covariances définies par

$$\sum[u, v] = cov(B_i(P_u), B_i(P_v))$$

où $B_i(T) = (T[P_1] - M_i[P_1], ..., T[P_n] - M_i[P_n])$ le vecteur bruit pour un échantillon T et P_i les points d'intérêt.

5.4.6 Phase d'attaque

Cette étape consiste essentiellement à classifier une trace S. Pour cela, il calcule la probabilité d'observation de S pour savoir si elle provient d'une opération déjà testée par le clone c'est à dire calculer le vecteur de bruit n en utilisant le signal moyen M_i du template. Puis il calcule la probabilité d'observation de n pour pouvoir classifier avec la méthode du maximum de vraisemblance:

$$p_{B_i}(n) = \frac{1}{\sqrt{(2\pi)^n \sum |B_i|}} \exp(-\frac{1}{2} n^T (\sum B_i)^{-1} n)$$

avec $n \in R^n$.

5.4.7 Difficultés des attaques templates

Plusieurs problèmes se dressent à l'utilisation de l'attaque template comme gérer la grande quantité des observations qui peut être vue comme une matrice de dimension importante. Pour cela, on procède à une **analyse en composantes principales (ACP)** qui est une technique statistique utilisée pour traiter des données provenant des situations où plusieurs variables sont mesurées simultanément. Cette méthode permet de réduire la dimensionnalité de la matrice du fait de la corrélation des données en se basant sur les vecteurs propres d'une matrice de covariance.

5.4.8 Modèle d'attaque template

En pratique, mener des attaques templates pour retrouver la clé du chiffrement en se basant uniquement sur les traces issues du dispositif clone n'est pas évident. En effet, du fait du réalisme de l'attaque, les traces ne proviennent jamais du même système, de ce fait ne peuvent donc pas être identiques. L'attaquant qui veut récupérer des informations sur un secret commence par isoler une partie du secret sur lequel une recherche exhaustive est possible. Puis, il construit des classes qui représentent la part du secret à extraire car en général il y a une classe par valeur possible de la partie de secret isolée.

5.4.9 Attaques template avec le modèle du poids de Hamming

Les templates construits sont très proche les uns des autres de point de vue de la densité de probabilité. De ce fait, pour pouvoir exploiter les templates, on doit mettre en place une fonction de classification qui sera utilisé dans le profilage pour essayer de réduire les informations. L'objectif est de trouver la distance de Hamming des traces sur un octet puis de les classer en un regroupement de sous classes de poids de Hamming sur un bit. (**Dégénérescence de la distance de Hamming appliquée à un algorithme DES**).
L'objectif est d'augmenter la plus grande valeur propre des modèles et par conséquent la variance totale car on sait que la variance représente la dispersion des densités de probabilité. **Une attaque Template réussira mieux si les densités sont bien séparées les unes des autres.**

6 Apprentissage automatique

L'utilisation de l'apprentissage automatique ou "Machine Learning (ML)" ici a deux objectifs:

- pour mesurer la performance de prédiction des modèles
- la méthode "sklearn.metrics.classification_report()" fournit la précision, le rappel, le score f1, et le support pour chaque classe, valeurs qui peuvent être utilisés en les combinant pour obtenir une image plus précise des performances du modèle.

6.1 Définition

"L'apprentissage automatique (AA) est la discipline donnant aux ordinateurs la capacité d'apprendre sans qu'ils soient explicitement programmés" Arthur Samuel, 1959.
L'apprentissage automatique est la science de programmer les ordinateurs de sorte qu'ils puissent apprendre à partir de données. En voici une autre plus technique :
"Étant donné une tache T et une mesure de performance P, on dit qu'un programme informatique apprend à partir d'une expérience E si les les résultats obtenus sur T, mesurés par P, s'améliorent avec l'expérience E" Tom Mitchell, 1997.

6.2 Généralités

Les exemples utilisés par le système pour son apprentissage constituent le jeu d'entraînement (**training set**). Chacun d'eux s'appelle une observation d'entraînement (on parle d'échantillon).
Le filtre antispam est un programme d'apprentissage automatique qui peut apprendre à identifier les emails frauduleux à partir d'exemples de pourriels ou **"spam"** (par exemple, ceux signalés par les utilisateurs) et de message normaux (parfois appelés **"ham"**) . Dans le cas présent, la tache T consiste à identifier parmi les nouveaux emails ceux qui sont frauduleux, l'expérience E est constitué par les données d'entraînement, et la mesure de performance P doit être définie (on peut par exemple prendre le pourcentage de courriels correctement classés). Cette mesure de performance particulière, appelée exactitude (accuracy), est souvent utilisée dans les taches de classification. Un filtre antispam basé sur des techniques de Machine Learning apprend automatiquement quels sont les mots et les phrases qui constituent de bons prédicteurs de courriels frauduleux en détectant des associations de mots aux fréquences inhabituelles dans des exemples de courriels frauduleux comparés à des exemples de courriels licites. Le programme est beaucoup plus court, plus facile à maintenir et a plus de chances de fournir de bons résultats.
Pour résumer, l'AA est excellent pour les problèmes pour lesquels les solutions existantes requièrent beaucoup d'ajustements fins ou de longues liste de règles et pour l'exploration des problèmes complexe et les gros volumes de données.

6.3 Types d'apprentissage automatique

Il existe plusieurs types de systèmes d'apprentissage automatique. Il est utile de les classer en grandes catégories :

- classification automatique d'informations bien définie par des **réseaux de neurones convolutifs(convolutional neural networks)** : par exemple la détection de tumeurs sur des scanners cérébraux ainsi que la classification automatique d'articles de presse.
- élaboration d'une représentation graphique claire et explicite à partir d'un jeu de données complexe de grande dimension : il s'agit de visualisation de données mettant fréquemment en œuvre des techniques de réduction de dimension.

6.4 Types de système d'apprentissage automatique

Ce sont les systèmes d'apprentissage automatique qui peuvent être classés en fonction de l'importance et de la nature de la supervision qu'ils demandent durant la phase d'entraînement. Il existe 4 catégories majeures :

- l'apprentissage supervisé:
  Dans l'apprentissage supervisé, les données d'entrainement que vous fournissez à l'algorithme comportent les solutions désirées, appelées étiquettes (en anglais, labels). Un exemple classique de tache d'apprentissage supervisé est la **classification**. L'apprentissage du filtre spam s'effectue à partir de nombreux exemples d'émail accompagnés de leur classe (spam ou normal), à partir desquels il doit apprendre comment classer les nouveaux emails.
  Une autre tâche classique consiste à prédire une valeur numérique cible (en anglais, "Target") à partir des valeurs d'un certain nombre d'"attributs" ou "variables". Ces valeurs sont appelées les caractéristiques (en anglais, "features") d'une observation. Ces variables sont appelées des variables explicatives ou encore prédicteurs. Une tache de ce type est une **régression**. Pour entraîner le système, on doit lui fournir à la fois la variable explicatives et les variables à expliquer (les caractéristiques et les étiquettes).

- l'apprentissage non supervisé :
  Dans l'apprentissage non supervisé, les données d'apprentissage ne sont pas étiquetées, en d'autres termes le système essaie d'apprendre sans professeur. Voici quelques-uns des plus importants algorithmes d'apprentissage non supervisé :

  - **partitionnement (clustering)** :
    le système va tenter de détecter des groupes d'individus similaires pour les ranger ensemble, sans aide de notre part.
  - **visualisation et réduction de dimension**:
    le système va analyser les différents individus, les regrouper puis donne une représentation graphique 2D ou 3D des données sous forme de nuage de points de chaque groupe extrait. Une tache connexe est la réduction de dimension. L'objectif est de simplifier les données sans perdre trop d'informations. Pour y parvenir, le système procède à une extraction de données (en anglais "feature extraction") qui consiste à agréger plusieurs variables corrélées.

- l'apprentissage semi-supervisé :
  Des algorithmes qui peuvent s'accommoder de données partiellement étiquetées.

- et l'apprentissage avec renforcement :
  L'apprentissage par renforcement est quelque chose de très différent : le système d'apprentissage, appelé "**agent**" dans ce contexte, peut observer l'environnement, sélectionner et accomplir des actions, et obtenir en retour des récompenses (ou des pénalités sous la forme de récompenses négatives). Il doit alors apprendre par lui-même quelle est la meilleure stratégie pour obtenir au final autant de récompenses que possible. C'est le cas du programme AlphaGo de Deepmind qui a battu le champion du monde de Go.

  Puis il ya les systèmes qui sont capables ou non d'apprendre progressivement à partir des flux de données entrantes:

- apprentissage groupé (ou batch):
  Dans l'apprentissage groupé (en anglais, batch learning), le système est incapable d'apprendre progressivement : il doit être entrainé avec toutes les données disponibles. Cela nécessite en général beaucoup de temps et de ressources informatiques.

- apprentissage en ligne:
  Dans l'apprentissage en ligne, le système est entraîné progressivement en alimentant peu à peu avec des observations, soit une à une, soit par petits groupes appelés mini-lots (en anglais, mini-batches). Chaque étape d'apprentissage est rapide et économique en ressources car le système apprend à partir de nouvelles données au fur et à mesure de leur arrivée. Un paramètre important de système d'apprentissage en ligne est le rythme auquel ils doivent s'adapter à l'évolution des données : c'est ce qu'on appelle le taux d'apprentissage (en anglais, learning rate), la vitesse d'adaptation aux nouvelles données.

  Enfin, ceux qui apprennent à partir de modèle ou d'observation:

- Le système apprend par cœur, puis il généralise à des nouveaux cas qu'il compare aux exemples appris (ou à un sous ensemble de ceux-ci) en utilisant une mesure de similitude. Par exemple, la nouvelle observation serait classée comme triangle car la majorité des observations les plus semblables appartiennent à cette classe.
- C'est la manière de généraliser à partir d'un ensemble d'exemples la construction d'un modèle de ces exemples pour en extraire des prédictions. Pour ce faire, l'algorithme doit :
  - charger les données initiales qui seront souvent au format csv
  - préparer les données pour pouvoir les disposer sur un graphique
  - visualiser les données
  - sélectionner un modèle linéaire avec sklearn. Linear_model.LinearRegression() car le modèle sera obtenu par régression linéaire.
  - puis entrainer le modèle,
  - et enfin, on réalise une prédiction pour les prochaines variables.

7 Applications aux attaques SCA

Le deeplearning sur les attaques par canaux cachés ou "dl-sca" définit les critères nécessaires aux différentes certifications qui sont reconnues résistants aux attaques liées aux implémentations cryptographiques qui sont les sca.
L'apport de la technique de ML permet de lancer une attaque dite **profiled side-channel attack** qui ne nécessite pas beaucoup de données récoltées pour en extraire un modèle d'attaque efficace. L'utilisation du ML en Sca se déroule généralement en 2 étapes :

- le profilage ou profiling step (ou entrainement) durant laquelle l'adversaire évalue et définit la caractérisation de la fonction de la fuite.
- l'attaque ou attack step durant laquelle il récupère les clés secrètes de la machine ciblée.

Plusieurs techniques de profilages existent mais celle qu'on retient est le modèle des templates attack qui est basé sur une distribution Gaussienne. En effet, c'est le modèle qui représente le mieux une attaque SCA lorsque la distribution gaussienne est vérifiée à la première étape et que le la taille des fuites observées est petite lors de la seconde étape. La distribution Gaussienne est connue pour être la plus efficace théoriquement. Une variante des attaques templates est l'attaque stochastique qui utilise la méthode des régressions linéaires en ML.
Les techniques de ML sont avantageuses par rapport aux autres types d'attaques lorsqu'un nombre limité de traces peuvent en être déduite alors que l'estimation sur les templates ne sont pas suffisantes. De plus, le problème de la dimensionalité est gérable en ML tandis que ce n'était pas le cas avec les templates attacks.

8 Conclusion

Pour conclure, l'utilisation de technique de machine learning est très efficace. En effet, on constate que pour un nombre de scénarios données, l'approche par ML donne de meilleur résultat comparé aux attaques templates car nécessite plus d'informations. Cela rend l'attaque moins étendue dans le temps et occupe aussi moins de mémoire lors des calculs. L'apport du machine learning rend l'attaque furtive et précise.
Aussi, il est indéniable que la participation des universitaires en cryptanalyses a permis de rendre les échanges d'informations sur internet plus sécurisées. Les études sur la sécurité de l'information ne manque pas de sujet qui reste encore a être abordé. En effet, il est certains que les avancés sur la sécurité des réseaux dépassent largement les attaques qui peuvent être matérielle. Des attaques physiques qui sont aussi nocives que ceux algorithmiques.
Enfin, une solution qui me semble logique est de rendre le matériel utilisé plus sécurisé. Pour cela, il est essentiel de faire en sorte qu'au niveau de la conception du matériel, les personnes qui s'occupent de la partie hardware travaillent plus étroitement avec ceux qui s'occupent de la partie software.