Approche didactique pour l'enseignement de l'attaque DPA ciblant l'algorithme de chiffrement AES

Lilian BOSSUET

Telecom Saint-Etienne – Laboratoire Hubert Curien
18 rue B. Lauras – 42000 SAINT ETIENNE Cedex – France
Email : lilian.bossuet@univ-st-etienne.fr

Résumé

Depuis un peu plus d'une dizaine d'années la sécurité matérielle (la sécurité des données et des systèmes électroniques) est devenue un nouvel axe de recherche en électronique. Cela s'est traduit par l'adoption en 2011 au sein du GDR SoC-SiP du CNRS d'un nouvel axe thématique « sécurité numérique » [1]. Cette évolution au niveau recherche est suivie par une prise de conscience des industriels, en particulier dans le domaine des systèmes embarqués, des risques liés à la sécurité matérielle. L'intérêt est donc de plus en plus grand pour les industriels de disposer de jeunes chercheurs et de jeunes ingénieurs formés (ou, au minimum, sensibilisés) à cette nouvelle problématique. C'est pourquoi en France, quelques cursus « électronique » de troisième cycle proposent déjà des enseignements concernant la sécurité matérielle. Concevoir un système électronique sécurisé nécessite de bien comprendre comment fonctionne les attaques physiques visant ces systèmes. Cependant, la mise en œuvre d'une attaque physique peut s'avérer longue et couteuse ce qui est difficilement compatible avec l'enseignement hors projets et stages. Le but de cet article est de présenter un exercice pratique de durée courte et peu couteux (3H00 de travaux pratiques en salle informatique sont nécessaires) qui permet aux étudiants de 3ème cycle (école d'ingénieur et master) de comprendre une des attaques physiques les plus emblématiques : l'attaque d'un chiffreur symétrique par analyse différentielle de sa consommation de puissance (en anglais, DPA : *Differential Power Analysis*).

Mots clés

Sécurité matérielle, attaque sur canaux cachés, DPA, travaux pratiques, chiffrement AES.

1. Introduction

Depuis un peu plus de dix ans, pour la conception d'un système électronique, une nouvelle contrainte a vue le jour : la sécurité matérielle. Bien souvent l'aspect sécurité est vu comme une application et non comme une contrainte de conception. Cependant, le développement des attaques physiques contre les systèmes électroniques a obligé les concepteurs à prendre en compte la sécurité comme une nouvelle contrainte [1]. Cela se traduit pour le concepteur par la nécessité de répondre à plusieurs questions.

Quelles sont les éléments du système à protéger ? Tout d'abord les données stockées et échangées par les systèmes électroniques doivent pour de nombreuses applications rester confidentielles, elles doivent être authentifiées et leur intégrité doit être garantie. Le système lui-même doit être protégé afin qu'une entité malveillante, externe (attaquant) ou interne (cheval de Troie matériel [2]), ne puisse prendre en main tout ou une partie du système. Enfin, la conception en tant que telle, c'est-à-dire la propriété intellectuelle que le concepteur a mise dans le système doit être protégée contre le vol, la copie illégale et la contrefaçon.

[1] http://www2.lirmm.fr/journees_securite/

Article publié par EDP Sciences et disponible sur le site http://www.j3ea.org ou http://dx.doi.org/10.1051/j3ea/2012004

De quoi se protéger ? Il existe de nombreuses attaques dites « physiques » qui ciblent les systèmes électroniques. Nous pouvons les classer en deux groupes distincts : les attaques actives et les attaques passives. Les attaques actives, comme leur nom l'indique vont conduire l'attaquant à agir sur le système attaqué. Les attaques passives exploitent l'analyse, en fonctionnement normal, d'informations s'échappant d'un circuit intégré qui réalise une fonction cryptographique. La figure 1 présente une classification schématique des attaques physiques.

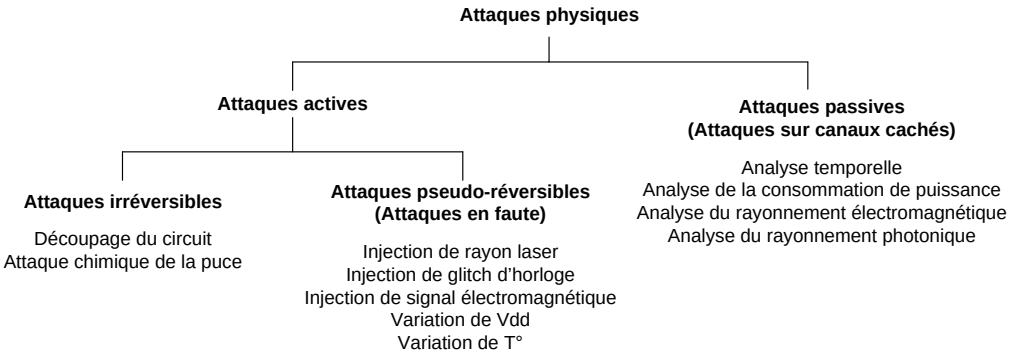


Figure 1 – Classification des attaques physiques.

Pour les attaques actives, il existe deux sous-groupes : les attaques irréversibles et les attaques pseudo-réversibles. Les attaques irréversibles conduisent à la destruction du système attaqué. Par exemple, l'abrasion chimique et le découpage laser d'un circuit intégré sont à ranger dans cette catégorie. Ce sont des attaques souvent utilisées lors de l'ingénierie inverse qui permettent alors à un attaquant d'obtenir des informations sur la conception d'un circuit. Les attaques actives pseudo-réversibles n'entrainent pas forcément la destruction du système attaqué, mais elles sont souvent invasives puisqu'elles nécessitent parfois la préparation du circuit (découpe partielle du boitier du circuit intégré par exemple) et qu'elles entrainent parfois la dégradation du système attaqué. On trouve classiquement sous cette appellation les attaques en fautes. Elles consistent, comme leur nom l'indique, à créer des fautes dans un circuit électronique par l'injection laser, par l'injection électromagnétique, par la dégradation de l'horloge d'un circuit synchrone (injection de *glitchs* d'horloge) ou encore par la variation de la tension d'alimentation. Les fautes ainsi créées peuvent entrainer le circuit dans des modes de fonctionnement conduisant à des erreurs. Il se trouve que les circuits pour la cryptographie sont très sensibles à ces erreurs et quelles peuvent être exploitées lors d'une cryptanalyse. Les attaques actives (irréversibles ou non) nécessitent du temps, des moyens, des connaissances techniques et des informations sur les systèmes attaqués. Elles sont donc des solutions réservées aux experts du domaine et elles sont relativement complexes à réaliser sous forme de travaux pratiques pour des étudiants.

Les attaques passives sont globalement beaucoup plus simples. Pour ces dernières, il s'agit d'analyser les informations issues de canaux cachés émanant du circuit en fonctionnement normal. Les canaux cachés sont des canaux physiques existants naturellement dans les systèmes le long desquels de l'information s'échappe. Or, les fonctions cryptographiques, bien que pouvant être extrêmement robustes théoriquement (c'est-à-dire mathématiquement) sont très sensibles aux fuites d'informations. Une quantité très faible d'informations qui fuite peut être exploitée pour casser un algorithme cryptographique très fort. C'est ce que les attaques par canaux cachés exploitent. Ces canaux sont par exemple, la consommation de puissance d'un circuit intégré, son rayonnement électromagnétique, son rayonnement photonique et tout ce qui peut être mesuré de l'extérieur et qui dépends de l'activité interne du circuit !

Comment se protéger (protéger son système et/ou ses données électroniques) ? Nous n'allons pas répondre ici à cette question dont la réponse demanderait un long développement et l'étude de nombreuses contre-mesures. Cependant, nous pouvons indiquer que pour développer une contre-mesure efficace la première étape consiste à bien comprendre et à maitriser l'attaque dont on souhaite se protéger. Tous les laboratoires effectuant des travaux en sécurité matérielle dispose de moyens d'attaques physiques permettant de valider les contre-mesures développées. D'un point de vue

pédagogique, si l'on souhaite amener les étudiants à proposer eux même des solutions, il est indispensable qu'ils maitrisent les attaques [3], c'est l'objet des travaux pratiques présentés en dernière partie de cet article.

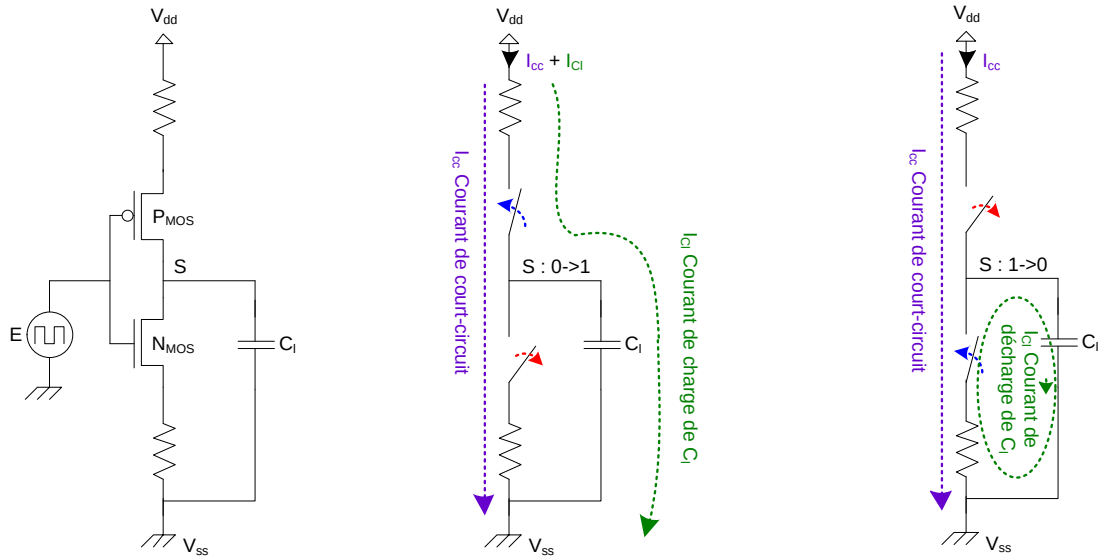
La suite de cet article est constituée de la façon suivante. La partie 2 donne quelques pré-requis nécessaires à la bonne compréhension de l'attaque par analyse différentielle de la consommation de puissance (dans la séquence pédagogique, ces pré-requis sont vus sous forme de cours qui nécessitent pour être complet au minimum huit heures). Dans un premier temps, la partie 3 présente le concept théorique de l'attaque par analyse différentielle de la consommation de puissance, attaque communément appelée DPA pour *Differential Power Analysis*. Dans un second temps, la partie 3 propose une explication didactique de cette attaque en transposant la méthode employée dans un tout autre champ facilement compréhensible par tous. Enfin la partie 4 présente un exercice pratique qui conduit à la mise en œuvre de l'attaque par analyse de la consommation de puissance.

2. Pré-requis : modèle de consommation des circuits logiques CMOS et présentation du standard de chiffrement symétrique AES

Le but de cet article n'est pas de présenter en profondeur la théorie nécessaire pour appréhender l'attaque par analyse de la consommation de puissance d'un chiffreur. C'est pourquoi ce chapitre propose un rapide rappel des points essentiels à la compréhension de cette attaque. Le lecteur qui n'a pas les connaissances nécessaires à la compréhension du modèle de consommation des circuits logiques CMOS et du standard AES (*Advanced Encryption Standard* [4]) pourra trouver sans difficulté assez d'informations dans la littérature universitaire et dans de nombreux sites internet.

2.1 Modèle de consommation des circuits logiques en technologie CMOS

Rappelons succinctement que le terme CMOS veut dire *Complementary MOS*, c'est-à-dire que la structure de base de toutes portes logiques en technologie CMOS se compose d'un couple de transistors MOS complémentaires (P-MOS et N-MOS) associés symétriquement et fonctionnant en régime de commutation. La figure 2-a donne le schéma classique simplifié d'un inverseur logique en technologie MOS.



a) Schéma de l'inverseur CMOS **b) Modèle à la commutation 1->0 de E** **c) Modèle à la commutation 0->1 de E**

Figure 2 – Schéma simplifié d'un inverseur CMOS et de son comportement en commutation.

Tant que les fuites dans les transistors MOS sont maitrisées, la puissance statique de l'inverseur (c'est-à-dire lorsque l'entrée (*E*) est stable à l'état haut ou à l'état bas) est négligeable. La puissance dynamique, consommée lors des commutations de l'entrée, est importante et dépends du type de

commutation. Par exemple, lors d'un passage de l'entrée (*E*) de l'état haut à l'état bas, illustré à la figure 2-b, l'alimentation fournit au travers du transistor P-MOS un courant de charge (I_{Cl}) à la capacité de ligne (C_l) connectée à la sortie (*S*), plus un courant transitoire de court-circuit (I_{cc}) (lorsque les deux transistors sont simultanément en régime de fonctionnement linéaire). Lors d'un passage de l'entrée (*E*) de l'état bas à l'état haut, illustré à la figure 2-c, la capacité de ligne se décharge au travers du transistor N-MOS, l'alimentation fournit un courant transitoire de court circuit (I_{cc}).

A partir de ces constatations, il est classiquement admis que le modèle de puissance dynamique (P_{dyn}) d'un circuit CMOS synchrone dépend quadratiquement de la tension d'alimentation (V_{dd}) et linéairement de la fréquence des commutations (f), de la capacité de charge (C_l) et d'un terme (τ) représentant le taux d'activité des signaux du circuit. Le modèle de consommation d'un circuit CMOS est donc décrit par l'équation suivante :

$$P_{dyn} = \tau \times C_l \times V_{dd}^2 \times f$$

Ce qu'il faut retenir de ce rappel c'est que la consommation de puissance des circuits logiques en technologie CMOS est principalement dynamique et qu'elle dépend du nombre et du type de commutations. C'est-à-dire qu'il y a une corrélation entre les commutations des signaux internes au circuit et sa consommation de puissance. Dans le cas de la réalisation par le circuit d'une opération arithmétique ou logique, le résultat de l'opération entraine des commutations de signaux et a donc une influence sur la consommation de puissance de circuit. Il y a donc une partie de l'information liée à ce calcul qui se trouve dans le signal de consommation de puissance du circuit. C'est cette fuite d'information qui peut être exploitée pour attaquer une opération de chiffrement. Une attaque de ce type cherche à trouver la clé de chiffrement utilisée par le chiffreur. Si le lien peut être fait entre la consommation de puissance et la clé, c'est-à-dire si l'information clé se trouve dans le signal de consommation de puissance, alors une attaque par analyse de la consommation de puissance est possible.

Voyons maintenant, en étudiant rapidement le standard de chiffrement symétrique AES largement utilisé aujourd'hui, si il est possible de faire un lien entre la clé secrète utilisée et la consommation de puissance d'un circuit réalisant ce chiffrement.

2.2 Présentation du standard de chiffrement symétrique AES

Le standard AES décrit un algorithme de chiffrement symétrique. Comme l'a préconisé en 1883 Auguste Kerckoffs dans les célèbres lois de la cryptographie qui portent son nom [5] : « un système cryptographique ne doit pas exiger le secret et peut sans inconvénient tomber entre les mains de l'ennemi ». Cela veut dire qu'un algorithme de chiffrement peut et même doit être connu de tous. Conserver le secret de l'algorithme n'apporte pas de sécurité, c'est pourquoi le chiffrement AES est disponible publiquement [4] alors même qu'il est utilisé dans les transactions bancaires et dans les protocoles sécurisés sur internet. Le secret d'un chiffrement symétrique tient uniquement dans une clé secrète qui est partagée (de façon sécurisée, mais ceci est un autre problème) entre l'expéditeur et le destinataire d'un message chiffré. La figure 3 schématise le principe d'un chiffrement symétrique. Le terme « symétrique » vient du fait que la même clé secrète est utilisée pour l'opération de chiffrement et l'opération de déchiffrement.

Un attaquant potentiel peut facilement obtenir le texte chiffré, mais sans la connaissance de la clé secrète il ne pourra rien faire de cette connaissance. C'est pourquoi les attaques cherchent en premier lieu à découvrir la clé secrète. Si celle-ci tombe dans les mains de l'attaquant alors la sécurité est « cassée ». Cependant, dés lors qu'ils utilisent des tailles de clé assez grandes (aujourd'hui 256 bits ou plus) les algorithmes de chiffrement symétriques tel que AES sont théoriquement sures. Les éléments en rouge sur la figure 3 (le texte en clair et la clé secrète) sont les éléments sensibles qui ne doivent pas tomber dans les mains de l'attaquant.

Puisque l'algorithme AES est publiquement disponible [4] et très largement documenté, nous ne détaillons pas son fonctionnement dans ce court chapitre. Nous donnons uniquement quelques informations permettant d'appréhender le développement de l'attaque par analyse différentielle de la consommation de puissance.

L’algorithme AES est un algorithme de chiffrement par blocs. C'est-à-dire qu’il travail avec en entrée des blocs de taille fixe du texte en clair. Par exemple, pour la version d’AES utilisant une clé secrète de longueur 128 bits, le texte en clair est traité par paquet de 128 bits.

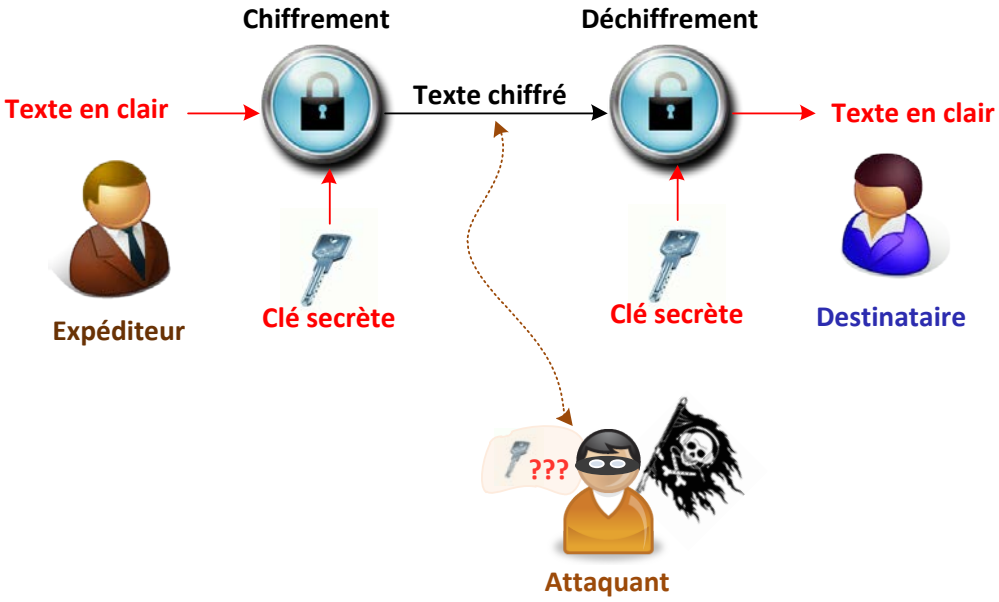


Figure 3 – Principe du chiffrement-déchiffrement symétrique.

Quatre opérations simples sont utilisées pour le chiffrement dans l’algorithme AES : *AddRoundKey*, *SubBytes*, *ShiftRow* et *MixColumn*. Ces quatre opérations sont réalisées plusieurs fois dans un ordre précis. La figure 4 décrit le déroulement de l’algorithme AES. Dans cette figure, comme pour la figure 3, les éléments en rouge sont les éléments sensibles à protéger.

Un bloc de 128 bits du texte en clair est pris comme entrée du chiffreur, il est représenté par une matrice de 4 par 4 octets. Une clé secrète, *Ks*, de longueur 128 bits est entrée par l’utilisateur, elle est connue uniquement de lui et du destinataire. D’un point de vue représentation, cette clé est vue comme une matrice de 4 par 4 octets comme le texte en clair (il en est de même pour chacune des sorties des opérations que nous allons décrire dans la suite). Une première opération de chiffrement est réalisée, il s’agit de l’addition dans un Corps de Gallois de la clé secrète et du bloc de texte en clair. Cette opération revient à un ou-exclusif bit à bit des deux entrées. Elle est appelée *AddRoundKey* dans le standard.

La nouvelle donnée de 128 bits ainsi obtenue va parcourir neuf rounds consécutifs. Un round de l’algorithme AES est constitué des quatre opérations de base de cet algorithme. A chaque itération une nouvelle clé secrète *K(i)*, l’indice *i* variant de 1 à 9, est utilisée. Chacune de ces clés est calculée à partir de la clé secrète *Ks* dans un bloc d’expansion de la clé que nous n’allons pas développer ici car sa compréhension n’est pas nécessaire pour expliquer l’attaque qui nous intéresse dans cet article.

La première des quatre opérations d’un round est l’opération de substitution d’octets *SubBytes*. Elle utilise un tableau de substitution, de 16 par 16 octets, appelé *Sbox*. Les valeurs de ce tableau sont décrites dans le standard. Un octet en entrée de cette fonction désigne une case du tableau *Sbox* en sélectionnant une colonne (avec les 4 bits de poids faibles de l’octet d’entrée) et une ligne (avec les 4 bits de poids forts de l’octet d’entrée). La valeur de l’octet contenue dans la case du tableau *Sbox* ainsi sélectionnée est substituée à la valeur de l’octet d’entrée. L’opération *SubBytes* est non-linéaire ce qui, nous le verrons plus tard, est une caractéristique importante pour la réalisation de l’attaque DPA.

L’opération suivante est une opération de rotation d’octet, *ShiftRow*, sur la matrice de 4 par 4 octets en entrée de l’opération. Cette opération effectue un décalage circulaire d’un octet vers la droite de la

seconde ligne du tableau, un décalage de deux octets vers la droite de la troisième ligne du tableau et un décalage de trois octets vers la droite de la quatrième ligne.

L'avant dernière opération, *MixColumn*, est une opération linéaire de diffusion de l'information entre les colonnes de la matrice de 4 par 4 octets en entrée. Elle correspond à la multiplication de deux matrices. Elle est suivie par une opération *AddRoundKey* qui utilise la clé secrète *K(i)* correspondant au round en cours.

A la suite de ces neufs rounds complets, un dernier et dixième round est effectué. A la différence des neuf rounds précédents, celui-ci ne comporte pas d'opération *MixColumn*.

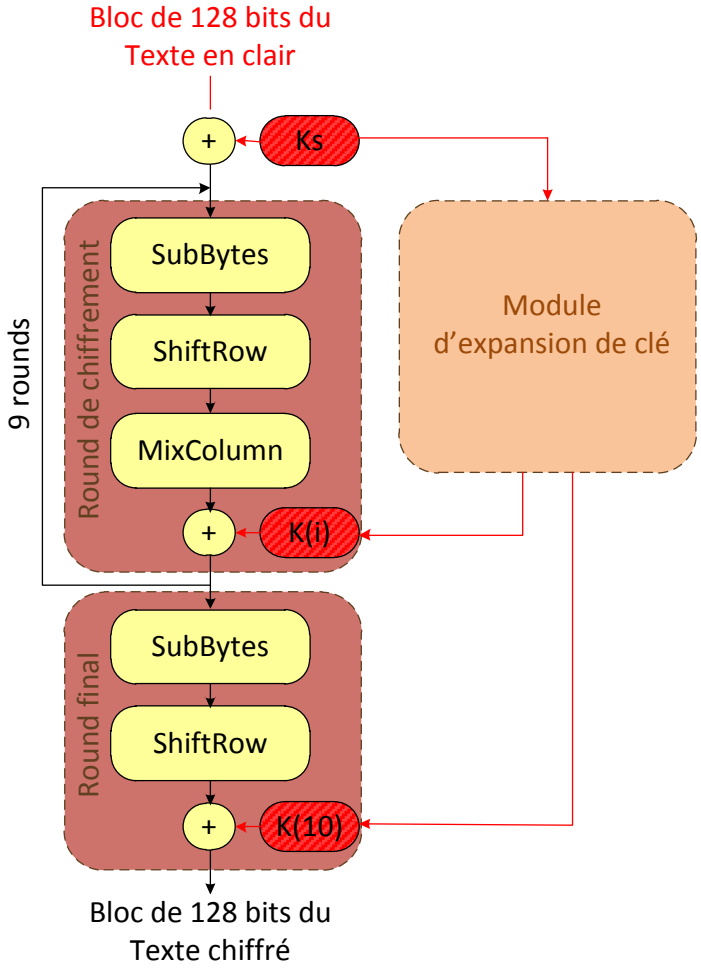


Figure 4 – Principe du chiffrement AES 128 bits.

Il est nécessaire pour préparer l'attaque DPA d'étudier s'il existe dans l'algorithme AES, à partir du moment ou une implantation physique (logicielle ou matérielle) est réalisée dans un circuit, une corrélation entre la puissance consommée par le circuit et la clé secrète. En d'autres termes, est-il possible à un moment du calcul de trouver dans la consommation de puissance du circuit l'information *clé secrète* ?

Pour cela étudions par exemple le modèle de consommation à la sortie de la première étape, *AddRoundKey*, qui est un ou-exclusif entre le texte en clair et la clé secrète *Ks*. Comme le modèle de consommation de puissance d'un circuit CMOS nous l'a montré, il n'y a de consommation de puissance significative qu'au moment d'une transition de la sortie des opérateurs logiques. C'est-à-dire lors de l'écriture en sortie d'une opération d'une nouvelle valeur. Il faut donc étudier la consommation de puissance de l'opération considérée avec deux entrées consécutives distinctes (T_1, T_2). La consommation de puissance est alors directement dépendante du nombre de bits de la sortie qui changent d'état, c'est-à-dire de la valeur de la distance de Hamming (fonction $DH(x_1,x_2)$ *avec* x_1, x_2

deux mots binaires de même longueur) entre deux valeurs successives de la sortie. Cela peut se traduire de la façon suivante :

$$P_{dynAddRoundKey} = f\left(DH\left(AddRoundKey(T_1)_{(K_s)}, AddRoundKey(T_2)_{(K_s)}\right)\right)$$

Que l'on peut écrire :

$$P_{dynAddRoundKey} = f\left(DH(T_1 \oplus K_s, T_2 \oplus K_s)\right)$$

La distance de Hamming entre deux nombres binaires est égale au poids de Hamming (fonction *PH(x)*) du résultat du ou-exclusif entre ces deux nombres, il vient donc que :

$$P_{dynAddRoundKey} = f\left(PH(T_1 \oplus K_s \oplus T_2 \oplus K_s)\right)$$

Que l'on peut simplifier comme suit :

$$P_{dynAddRoundKey} = f\left(PH(T_1 \oplus T_2)\right)$$

Nous pouvons donc conclure que la consommation de puissance durant l'opération *AddRoundKey* ne dépend que de la distance de Hamming des deux blocs de texte en entrée et pas du tout de la clé secrète. Nous obtenons un résultat identique pour les opérations linéaires *ShiftRow* et *MixColumn*. Il n'est donc pas possible d'exploiter ces opérations pour l'attaque par analyse de la consommation de puissance.

Nous pouvons faire cette même étude pour l'opération non-linéaire *SubBytes*, les textes T_1 et T_2 sont joués initialement successivement. La consommation de puissance dynamique pour cette opération s'exprime de la façon suivante :

$$P_{dynSubBytes} = f\left(DH\left(SubBytes(T_1 \oplus K_s), SubBytes(T_2 \oplus K_s)\right)\right)$$

En utilisant la fonction de calcul du poids de Hamming, il vient que :

$$P_{dynSubBytes} = f\left(PH\left(SubBytes(T_1 \oplus K_s) \oplus SubBytes(T_2 \oplus K_s)\right)\right)$$

Comme la fonction *SubBytes* est non-linéaire, il n'y a aucune façon de simplifier cette égalité. Ainsi nous pouvons conclure que l'information *clé sécrète* (*Ks*) est « contenue » dans le signal de consommation de puissance du circuit lors du calcul de la première opération *SubBytes* ! C'est cette faille de sécurité qui va être exploitée lors de l'attaque par analyse différente de la consommation de puissance, l'attaque DPA, que nous allons présenter dans la suite de cet article.

3. L'attaque du chiffrement symétrique par analyse de la consommation de puissance

3.1 Fondements de l'attaque DPA et de l'attaque CPA

En 1999 Paul Kocher a révolutionné le monde de la sécurité matérielle (dans ces années là ce monde était restreint aux cartes à puce) en publiant une nouvelle attaque sur canaux cachés exploitant la mesure de la consommation de puissance d'un circuit au cours d'un chiffrement [6]. La faille utilisée est la corrélation dans les circuits électroniques de technologie CMOS entre la consommation de puissance dynamique et le nombre de transistors qui commutent de l'état haut à l'état bas et de l'état bas à l'état haut. Ainsi, il est possible de corréler la consommation de puissance aux résultats des opérations réalisées dans le circuit. Par exemple, comme nous venons de le voir, dans l'algorithme de chiffrement symétrique AES, les valeurs successives de la sortie d'une opération non-linéaire telle que *SubBytes* [4] sont liées à la clé secrète utilisée dans le chiffrement. Cela entraine une corrélation entre la consommation de puissance lors du calcul de cette fonction et la clé secrète. La méthode d'attaque différentielle proposée par Kocher, la DPA (*Differential Power Analysis*), utilise cette caractéristique pour trouver la clé secrète en ciblant des sous-clés de la longueur d'un octet. L'attaque sur les sous-clés est possible car durant les deux premières étapes de l'algorithme AES, *AddRoundKey* et *SubBytes*, les octets du texte en entrée ne sont pas mélangés.

Lors de l'attaque DPA, pour chacun des textes en clair utilisés en entrée du chiffreur, un modèle logiciel reproduit les deux premières étapes de l'AES et donne un résultat sur 1 bit pour chacune des 256 sous-clés possibles. Ce résultat est un des bits en sortie de l'étape *SubBytes*, le premier est couramment utilisé. Pour le texte T_i, si ce bit vaut 0 pour la clé K_j la trace de puissance correspondant à ce texte, Ptr_i, est mémorisée dans un groupe G_0, si le bit vaut 1 elle est mémorisée dans un groupe G_1. La différence de la moyenne temporelle des traces du groupe G_1 avec la moyenne temporelle des traces du groupe G_0 pour la sous-clé K_j donne la courbe DPA associée à cette sous-clé, comme illustré sur la figure 5. Il existe donc 256 courbes DPA. Parmi celles-ci, la courbe dont la valeur moyenne est la plus élevée correspond à la clé secrète. Effectivement, il existe dans ce cas une corrélation entre les courbes de puissance et leur classement en deux groupes (la partie 3.2 de cet article apporte une preuve didactique de cette conclusion).

Cette attaque est particulièrement efficace, un nombre relativement faible de traces de puissance sont nécessaires pour la réaliser avec succès. En 2006, nous avons mis en œuvre un banc d'attaque DPA permettant de réaliser cette attaque en utilisant 15 000 textes en clair générés aléatoirement de taille 128 bits. Nous avons réalisé avec succès l'attaque DPA d'un chiffreur AES implanté dans un FPGA Altera Stratix-II de technologie SRAM 0,18µm [7]. L'efficacité de l'attaque DPA ciblant un FPGA de technologie SRAM avait été prouvée expérimentalement trois ans plus tôt pour un chiffrement asymétrique basé sur les courbes elliptiques [8] et deux ans plus tôt pour le chiffrement symétrique AES [9].

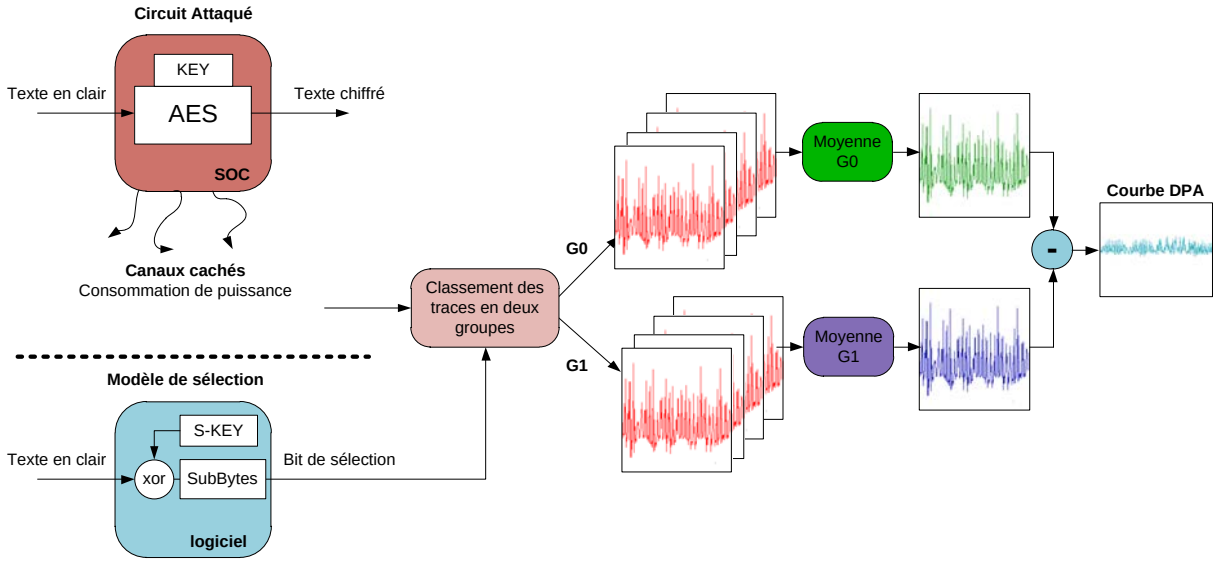


Figure 5 – Principe de l'attaque DPA.

En 2004, l'attaque DPA fut améliorée par des chercheurs de la société Gemplus[1] qui proposent de calculer directement la corrélation entre la consommation de puissance et le poids de Hamming en sortie du modèle pour chacune des sous-clés [10]. Cette nouvelle attaque DPA par corrélation, CPA (*Correlation Power Analysis*), utilise le coefficient de Pearson pour le calcul de la corrélation [9]. Ce coefficient est donné par l'expression suivante :

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

Dans laquelle, M est un vecteur contenant N traces de puissance Ptr_i, P_j est une colonne de la matrice P (de taille $Nx256$) 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. Le

[1] Fabricant de cartes à puce, devenu aujourd'hui Gemalto (http://www.gemalto.com/france/).

coefficient $C(M,P_j)$ représente la corrélation entre le vecteur de mesure M et le vecteur des poids de Hamming P_j calculé pour la sous-clé K_j. Parmi toutes les sous-clés disponibles, avec un nombre N suffisant (quelques milliers de textes en clair suffisent), celle qui donne le coefficient de corrélation le plus grand est la clé sécrète utilisée dans le circuit. Le principe de l'attaque CPA est décrit par la figure 6 sur laquelle apparait le calcul de la corrélation exprimée par l'expression de $C(M,P_j)$.

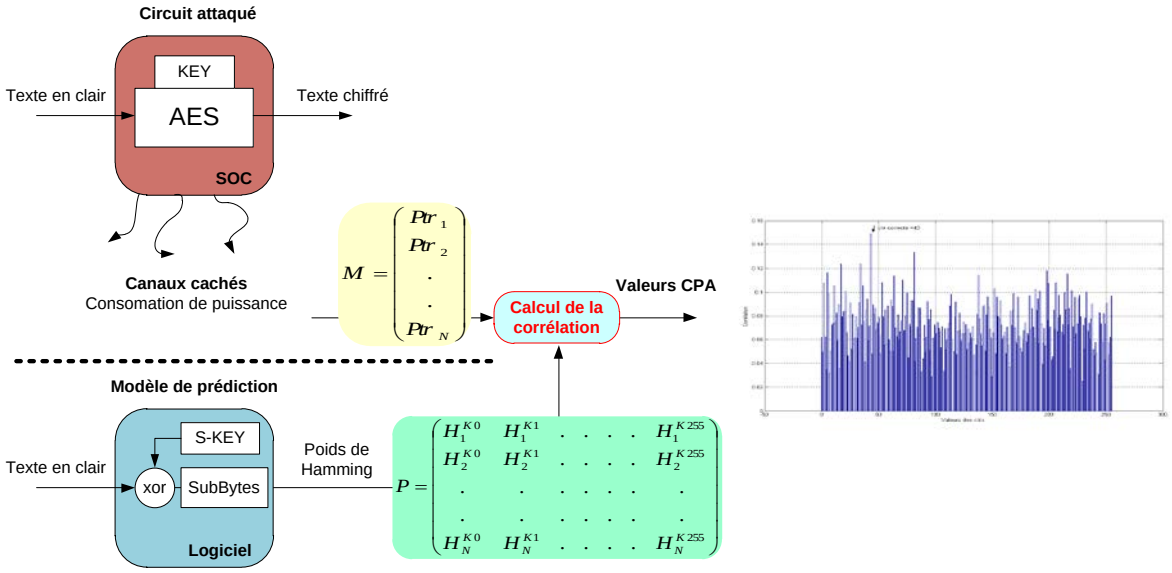


Figure 6 – Principe de l'attaque par corrélation, CPA.

Nous avons réalisé cette attaque en ciblant des FPGA de technologie SRAM (Altera Stratix EP1S25 en technologie 0,18 µm, Xilinx Virtex-II 2VP30 technologie en 0,15µm et Xilinx Virtex-4 SX25 en technologie 90 nm) et en ciblant des FPGA de technologie FLASH (Actel Fusion S600 en technologie 0,13µm) [11]. L'attaque réussie quelle que soit la technologie du FPGA ciblé en utilisant seulement un millier de traces de consommation de puissance.

3.2 Approche didactique pour l'explication de l'attaque

Il est souvent difficile pour les étudiants des cursus « électronique » de comprendre rapidement le concept de l'attaque DPA. Pour permettre d'augmenter cette compréhension nous avons pensé à une démarche pédagogique qui consiste à éloigner l'explication du contexte en la plaçant dans un contexte facilement compréhensible par tous. Il faut remarquer que l'inventeur de l'attaque DPA, Paul Kocher, est à la base spécialisé en biologie qu'il a étudiée à l'Université de Sanford. Pour un biologiste, la recherche de corrélation entre des caractéristiques biologiques est un travail courant. C'est grâce à cette culture que Paul Kocher a rapidement développé une étude de corrélation adaptée à la cryptanalyse. Allons donc sur un champ un peu plus biologique pour comprendre l'attaque DPA …

L'expérience menée est la suivante : prenons un groupe de personnes, disons un groupe constitué de 1000 individus. Prenons une caractéristique physique permettant de séparer ce groupe de 1000 individus en deux sous-groupes : la couleur des cheveux. Nous pouvons considérer pour simplifier qu'un premier groupe est constitué des individus aux cheveux blonds et aux cheveux roux (groupe *BR*), un second groupe est constitué des individus aux cheveux châtains et aux cheveux bruns (groupe *CB*). Bien entendu, des nuances capillaires devraient pouvoir être prises en compte dans ce découpage, mais nous sommes dans l'obligation de nous limiter à une approche grossière pour ne pas nuire à la clarté de l'explication.

Suite à ce découpage en deux groupes nous obtenons la répartition suivante : 30% d'individus dans le groupe *BR* et 70% dans le groupe *CB*.

Nous disposons des informations suivantes pour chacun des individus : date de naissance, lieu de naissance et couleurs des yeux. La question que nous nous posons est la suivante : il y a-t-il parmi ces informations une qui soit en corrélation avec le découpage initial en deux groupes ? Pour répondre à cette question nous allons pour chaque information disponible faire un nouveau découpage en deux groupes et comparer la répartition des couleurs de cheveux (*BR* ou *CB*) avec le découpage initial. La figure 7 schématise l'expérience.

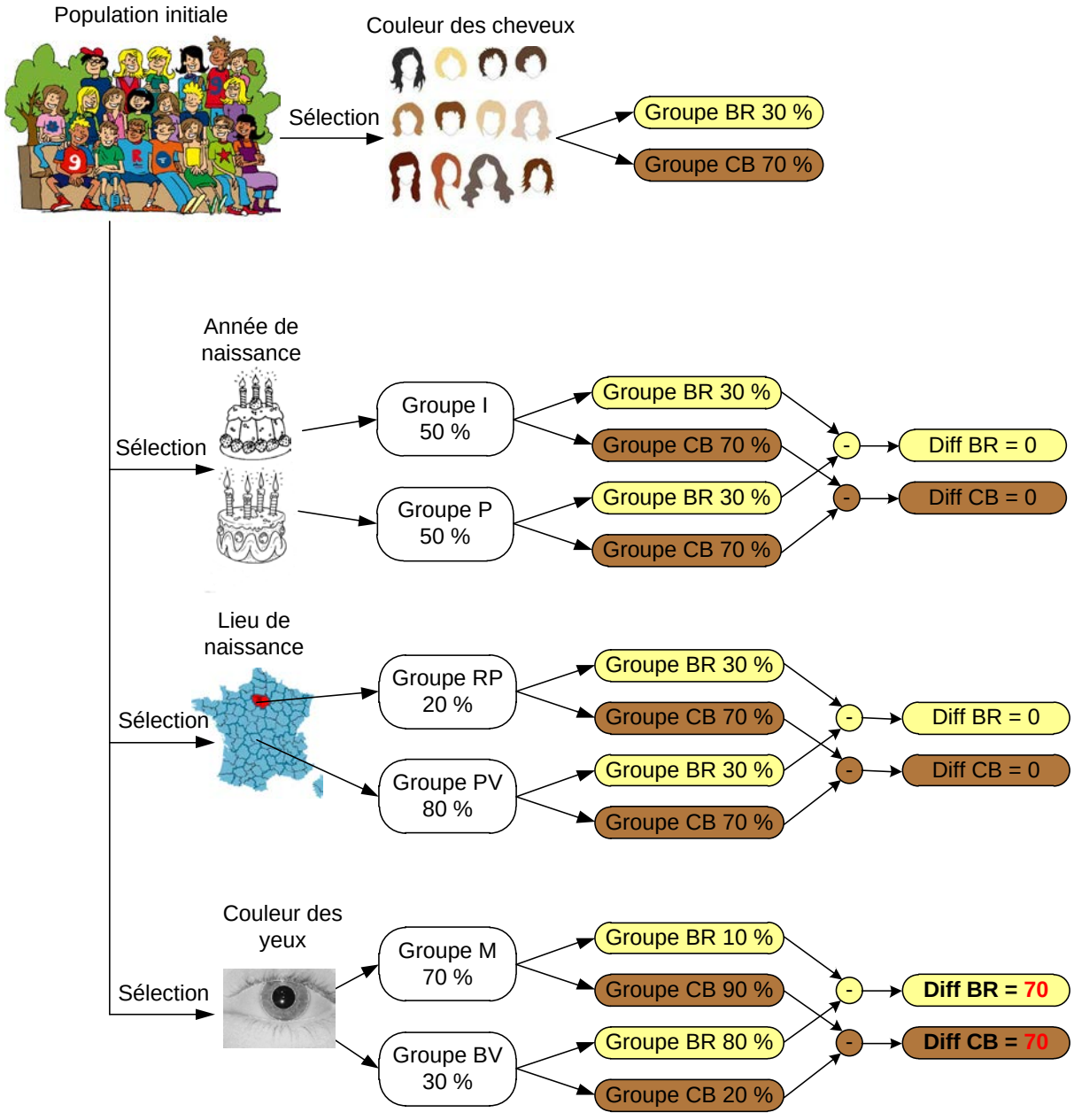


Figure 7 – Schématisation de l'expérience envisagée.

Dans un premier temps, nous nous posons la question de savoir pour chaque individu si son année de naissance est un nombre pair ou un nombre impair. En fonction de la réponse à cette question, nous séparons les individus en deux sous groupes *P* et *I*. Bien entendu chacun comprend qu'il n'y a pas, en moyenne, par exemple plus de blonds qui naissent une année paire qu'une année impaire. De ce fait la répartition des individus dans chacun des sous groupes *P* et *I* en fonction de la couleur des cheveux (*BR* ou *CB*) est à peu près identique à celle des deux groupes *BR* et *CB* sur la population totale. La

différence des *BR* dans les deux groupes *P* et *I* ainsi que celles des *CB* est proche de 0. Nous obtenons un résultat identique si l'on considère cette fois-ci le lieu de naissance. La question que l'on peut se poser dans ce cas est la suivante : l'individu est-il né dans la région parisienne (groupe *RP*) ou en province (groupe *PV*). De ces deux expériences nous pouvons en conclure qu'il n'y a pas de corrélation entre l'année de naissance et la couleur des cheveux des individus ainsi qu'entre le lieu de naissance (Paris/Province) et la couleur des cheveux.

Une nouvelle expérience nous conduit à examiner la dernière information disponible, la couleur des yeux. La question que l'on peut se poser pour séparer les individus en deux groupes est la suivante : l'individu a-t-il les yeux marrons (groupe *M*) ou non (groupe *BV* pour bleu et vert) ? Bien entendu cette fois ci la population du groupe *M* contient en proportion moins d'individus du groupe *BR* (disons 10 %) et plus d'individus du groupe *CB* (90%). Le groupe *BV* contient une proportion d'individus du groupe *BR* (80 %) bien plus importante que la proportion d'individus du groupe *CB* (20 %). Si l'on fait la différence des proportions de *BR* et de *CB* dans les groupes *M* et *BV* on trouve dans les deux cas la valeur 70 %. Cela indique, personne ne s'en étonnera, qu'il existe une forte corrélation entre la couleur des yeux et la couleur des cheveux des individus de la population choisie.

Le principe de l'attaque DPA, tel qu'il a été présenté en 1999, est le même que celui que nous venons de parcourir. Si l'hypothèse de classement (un bit du résultat de sortie de l'opération *SubBytes* avec une hypothèse de clé) des groupes de traces de consommation puissances est corrélée avec l'information contenue dans ces traces alors la différence des valeurs moyennes des courbes des deux groupes est non nulle. Comme la consommation de puissance durant l'opération *SubBytes* est corrélée avec la clé secrète, la clé d'hypothèse qui conduit à la plus grande différence des valeurs moyennes correspond à cette clé secrète.

Discussion : plusieurs éléments peuvent venir compliquer notre expérience didactique. Imaginons que les individus de notre sélection aient eu recours à une coloration des cheveux. La couleur prise en compte ne serait pas la couleur réelle. Autre moyen de fausser l'étude, si des individus portent des lentilles de couleurs, il peut y avoir des erreurs de jugement de la couleur des yeux. Bien entendu si d'autres informations sont fausses (année ou lieu de naissance) alors notre analyse est trompée. Ces considérations portent un nom dans le cas de la sécurité matérielle : des contre-mesures. Effectivement, le masquage, le brouillage de l'information (de façon aléatoire) sont des exemples de ce qu'il est possible de faire pour rendre l'attaque par analyse de la consommation de puissance plus difficile voir impossible. Ce dernier cas est obtenu lorsque la contre-mesure permet d'éliminer la corrélation entre la consommation de puissance du circuit et la clé secrète utilisée par le chiffreur.

Maintenant que le principe de l'attaque est compris, il semble indispensable de renforcer cette compréhension théorique par un exercice pratique, c'est ce que nous allons voir dans la suite de cet article.

4. Exercice pratique de mise en œuvre de l'attaque DPA ciblant un chiffreur AES

4.1 Présentation

En 2009, nous avons proposé pour la première fois un exercice pratique (de durée trois heures) permettant à des étudiants de troisième cycle (fin d'études d'ingénieur et/ou master) de réaliser l'attaque CPA d'un algorithme AES implanté sur un FPGA sans aucune contre-mesure. Nous avons choisi l'attaque CPA plutôt que la DPA car elle nécessite moins de textes en entrées et elle est donc plus rapide.

Les concepts théoriques sont vus en cours, les étudiants doivent uniquement décrire l'algorithme en langage *Matlab*. L'aspect expérimental de l'attaque (mesure des courbes de consommation de puissance) n'est pas réalisé par les étudiants comme le montre le figure 8. Effectivement, le temps nécessaire pour développer le chiffreur (description en langage VHDL), l'implanter dans le FPGA, effectuer la mesure de puissance pour un grand nombre de textes n'est pas compatible avec la durée du module pédagogique dédié à la sécurité matérielle. De plus, il s'agit ici de se concentrer sur le principe de l'attaque. Ceci dit, l'ensemble peut faire partie d'un projet et/ou d'un stage.

Nous fournissons donc aux étudiants les données brutes de l'attaque. Celles-ci sont disponibles aux personnes intéressées à condition d'envoyer un mail motivé à l'auteur de cet article. Les données fournis aux étudiants sont les suivantes :

- Une matrice de *T* de 1000 textes en clair générés aléatoirement (de taille 128 bits) que nous avons utilisée comme entrée du chiffrement AES. Celui-ci est implanté dans une cible FPGA ACTEL Fusion de technologie FLASH. Dans un souci de simplicité et de rapidité nous donnons uniquement le premier octet de chacun des textes. Ainsi les élèves réaliseront l'attaque uniquement sur le premier octet de la clé secrète.
- Une matrice *M* comprenant pour chacun des textes une courbe de consommation de puissance mesurée expérimentalement sur le circuit cible lors du chiffrement. Chacune de ces traces comporte 512 points. Chacun de ces points est codé sur 8 bits par l'oscilloscope que nous utilisons.

De plus, nous fournissons aux étudiants la norme AES [4] et une matrice *Sbox* qui contient les valeurs du tableau de substitution.

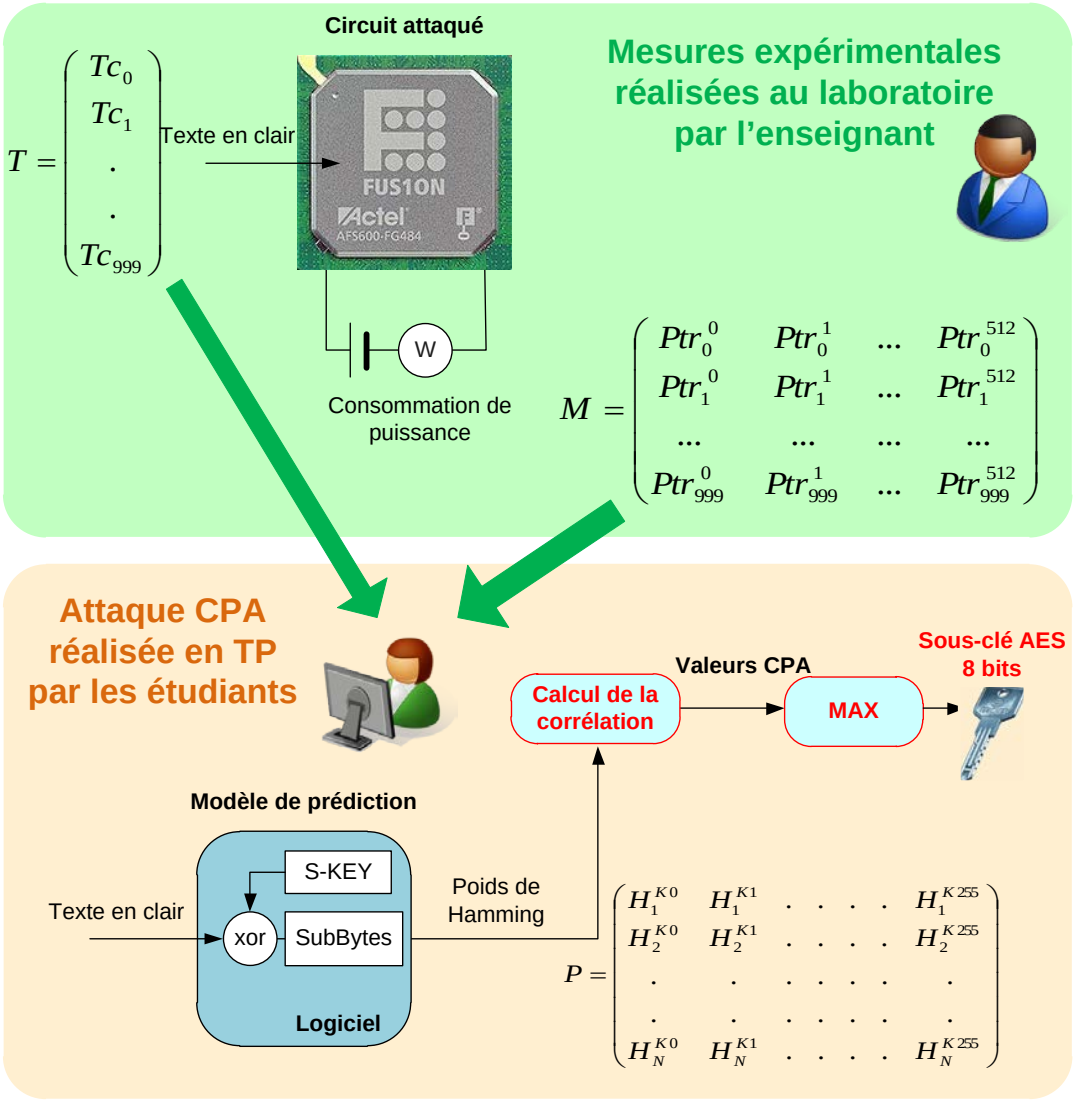


Figure 8 – Principe du TP CPA, le professeur réalise les mesures physiques et les étudiants développent l'algorithme de l'attaque.

4.2 Développement du modèle en langage *Matlab*

Les étudiants doivent donc modéliser en langage *Matlab*[1] les deux premières étapes de l'algorithme AES (*AddRoundKey* et *SubBytes*), puis ils doivent modéliser le calcul de la matrice *P* d'estimation du poids de Hamming de la sortie de l'opération *SubBytes* pour toutes les sous-clés possibles (sur 8 bits). Enfin un calcul de corrélation doit leur permettre d'extraire la valeur de la clé secrète, comme celle qui donne le maximum de corrélation.

La première étape est donc le calcul de la fonction *AddRoundKey* (pour cette étape, la variable *Key* est un tableau de 256 valeurs qui croient linéairement de 0 à 255). Cette fonction est un ou-exclusif logique que l'on peut décrire de la façon suivante :

```
Key = 0:255; %initialisation du vecteur de clés

Output_AddRoundKey= bitxor(Key,T);
```

Le résultat de cette étape, *Output_AddRoundKey*, est donc une matrice de taille 1000x256. Chaque élément de cette matrice est un nombre entre 0 et 255 pouvant être représenté par un octet.

La seconde étape est le calcul de la sortie de la fonction de substitution, *SubBytes*, pour cela nous fournissons un vecteur *Sbox* de dimension 256 éléments. Chaque élément de ce vecteur représente un octet du tableau de substitution. La description en langage *Matlab* est simple mais il faut prendre garde à la valeur de l'indice du vecteur qui va de 1 à 256, cela se décrit donc de la façon suivante :

```
Output_SubBytes = Sbox(Output_AddRoundKey+1);
```

Le résultat de cette étape, *Output_SubBytes*, est donc une matrice de taille 1000x256. Chaque élément de cette matrice est un nombre entre 0 et 255 pouvant être représenté par un octet.

La troisième étape est le calcul du poids de Hamming en sortie de la fonction de substitution. C'est une étape, qui quoi que simple, peut prendre du temps aux étudiants. De plus il existe plusieurs façons, plus ou moins élégantes, de modéliser ce calcul en langage *Matlab*. Voici un exemple de modélisation possible :

```
Output_SubBytes_bin=dec2bin(Output_SubBytes,8); %conversion en binaire

Weight_Hamm=bin2dec(Ouput_SubBytes_bin(1))+
bin2dec(Output_SubBytes_bin(2))+ bin2dec(Output_SubBytes_bin(3))+
bin2dec(Output_SubBytes_bin(4))+ bin2dec(Output_SubBytes_bin(5))+
bin2dec(Output_SubBytes_bin(6))+ bin2dec(Output_SubBytes_bin(7))+
bin2dec(Output_SubBytes_bin(8))

%attention   bin2dec(Sortie_SubBytes_bin(1))   correspond   au   MSB   et
bin2dec(Sortie_SubBytes_bin(8)) au LSB
```

Une autre solution possible plus élégante :

```
Weight_Hamm = zeros(length(Inputs),length(Key)); % initialisation

for i=1:length(T) % i est incrémenté de 1 à 1000
  for j=1:length(Key) % j est incrémenté de 1 à 256
    for k=1:8
      Weight_Hamm(i,j)= Weight_Hamm(i,j)+ bitget(Output_SubBytes(i,j),k);
    end
  end
end
```

Une dernière solution plus directe :

```
Weight_Hamm = sum(bitget(Output_SubBytes,1:8));
```

[1] Nous avons choisi le langage *Matlab* car la programmation est rapide mais il est envisageable de développer le même algorithme avec un langage du type *C*, dans ce cas le traitement peut être accéléré.

Le résultat de cette étape, *Weight_Hamm*, est donc une matrice de taille 1000x256. Chaque élément de cette matrice est un nombre entier compris entre 0 et 8. Cette matrice est notée *P* sur la figure 8.

La dernière étape de l'attaque consiste à calculer la corrélation entre la matrice *P* (*Weight_Hamm* dans notre modèle) et la matrice *M (Traces* dans notre modèle). Le calcul de la corrélation est immédiat en langage *Matlab* grâce à la fonction *corrcoef*. La corrélation est calculée pour chacun des points de la trace de puissance, ce qui donne en sortie une matrice de 256 *512 pour chaque texte (à faire 1000 fois)

```
[m,n] = size(Traces); % m (1000) ligne de n points (512)

for i=1:length(Key)
  for j=1:n
        r = corrcoef(Weight_Hamm(:,i), traces(:,j)); % Calcul direct pour
                                                       les 1000 textes
        Corr_Matrix(j,i) = r(1,2);
        clear r;
  end
end

[y i] = max(max(abs(Corr_Matrix))); % Recherche du maximum de corrélation
Secret_Key = Key(i) % extraction de la clé secrète
```

4.3 Résultat obtenu

Le résultat obtenu par les étudiants est présenté sur la figure 9 dans un espace à trois dimensions (axe X : valeur de clé, axe Y : échantillon et axe Z : corrélation). Pour une meilleure visualisation, cette figure ne présente qu'une vue partielle puisque pour celle-ci la valeur de clé ne varie que de 25 à 60 (et non de 0 à 255). Une recherche de la valeur maximum de corrélation permet de découvrir la clé secrète. Les étudiants mesurent également le temps nécessaire à la réalisation de l'attaque afin de comparer leurs algorithmes. Des évolutions sont possibles pour étendre cet exercice pratique : mettre en œuvre une attaque DPA du second ordre, expérimenter des contre-mesures. Cette dernière proposition nécessite de faire des mesures sur des implantations protégées de l'AES.

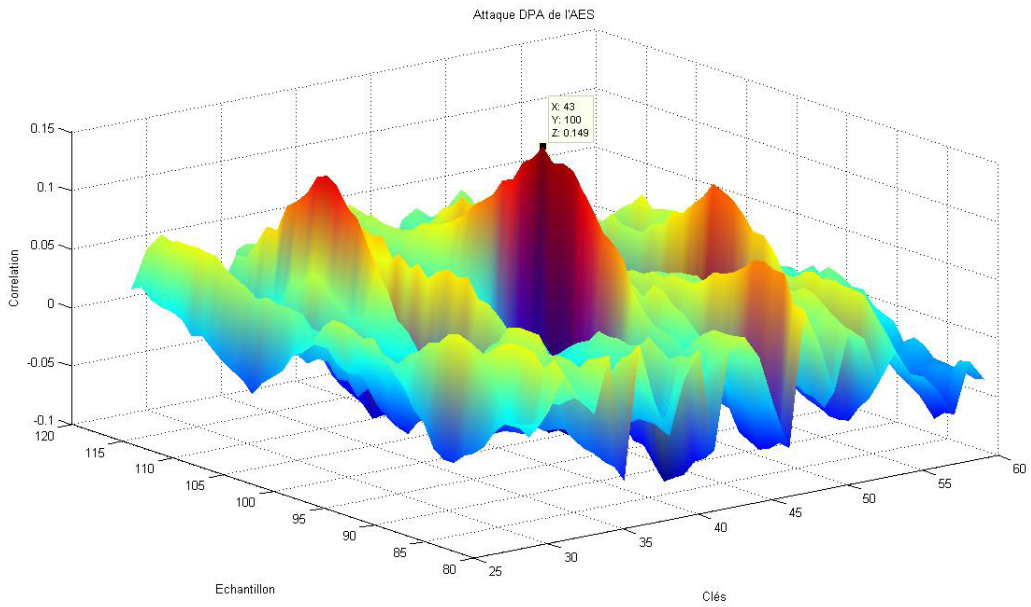


Figure 9 – Résultat du calcul de la corrélation obtenu par les étudiants pour toutes les sous-clés possibles et pour tous les échantillons de mesures de consommation de puissance.

4.4 Intérêts pédagogiques de cet exercice pratique

L'intérêt est grand pour les étudiants de mettre l'attaque en pratique. Tout d'abord il n'y a pas de difficulté technique à la réalisation de l'attaque. L'algorithme à développer en langage *Matlab* est très simple et demande un niveau de maitrise faible du langage. La difficulté réside dans la compréhension du fonctionnement de l'attaque vue en cours. Cet exercice pratique leur permet donc de bien comprendre le déroulement de l'attaque et de vérifier son efficacité. Ainsi, ils sont sensibilisés sur un cas réel à la fragilité d'une implantation matérielle cryptographique sans protection particulière. Couplé à des travaux pratiques sur l'implantation matérielle d'un circuit de chiffrement et/ou déchiffrement et de quelques contre-mesures, cet exercice pratique illustre parfaitement le thème de la sécurité matérielle des données. De plus, en fin de formation en troisième cycle, les étudiants utilisent leurs compétences en électronique numérique, cryptographie et traitement du signal à travers un même exercice pratique.

5. Conclusion

La sécurité matérielle a pris depuis quelques années une place très importante dans la recherche en électronique. Ce nouvel axe de recherche sort des domaines traditionnels d'application de la sécurité électronique (cartes à puce, produits militaires) pour élargir la problématique à l'ensemble des applications électroniques au premier rang desquelles les systèmes embarqués. Cela a conduit les industriels de l'électronique à s'intéresser à cette problématique. Il ressort, qu'ils ont une nécessité grandissante d'ingénieur et de chercheur sensibilisés et formés à la sécurité matérielle. De plus en plus de formations en électronique et systèmes embarqués, au niveau troisième cycle, incluent des modules d'enseignement concernant la sécurité matérielle.

Un des éléments caractéristiques des travaux en sécurité matérielle est le développement de contre-mesures aux attaques dites physiques qui ciblent les implantations matérielles de fonctions cryptographiques. Pour comprendre et développer ces contre-mesures, il est bien entendu absolument nécessaire de comprendre les attaques physiques. Ces attaques sont la plupart du temps très pratiques et nécessitent des connaissances dédiées. C'est pourquoi l'enseignement de la sécurité matérielle doit s'appuyer sur des travaux pratiques. Cependant, réaliser des attaques physiques est une tâche longue qui peut nécessiter du matériel. Dans cet article, après une présentation de l'attaque du chiffrement AES par analyse différentielle de la consommation de puissance, nous avons présenté le déroulement d'un TP peu cher et rapide permettant aux étudiants d'appréhender cette attaque. Ce TP peut être très facilement repris par des collègues enseignants puisque les données de l'attaque peuvent être diffusées. Nous espérons ainsi, promouvoir et faire évoluer l'enseignement de la sécurité matérielle dans les cursus « électronique » qui prend et va prendre une place de plus en plus importante dans les années à venir, nous en sommes convaincus.

Remerciements

L'enseignement de la sécurité matérielle est effectué depuis 2009 par l'auteur de cet article dans de nombreux établissements parmi lesquels : l'ENSEIRB-MATMECA (Institut Polytechnique de Bordeaux), SUP'COM Tunis, Telecom Saint-Etienne (TSE), Master EEAP parcours ESE (INSA Lyon, Central Lyon, CPE, TSE, Univ. Lyon). L'auteur remercie chaleureusement l'ensemble des étudiants à qu'il a enseigné ces aspects qui par leurs questions et remarques lui ont permis, à leur insu, de faire évoluer le cours et les travaux pratiques. En ce qui concerne l'enseignement de la sécurité matérielle, l'auteur de cet article a collaboré avec de nombreux collègues qu'il convient de remercier tout aussi chaleureusement, parmi lesquels Guy Gogniat, Bertrand Le Gal, Viktor Fisher, Florent Bernard, Robert Fouquet, Lubos Gaspard, Chiheb Rebaï et Najeh Kamoun.

Bibliographie

[1] L. Bossuet, G. Gogniat. *Chapitre 5 : La sécurité matérielle des systèmes embarqués*. Traité IC2, série réseaux et télécoms. Les systèmes embarqués communicants : mobilité, sécurité, autonomie, sous la direction de Francine Krief, aux éditions Hermes Science, septembre 2008, 306 pages (Chapitre 5 : 31 pages).
ISBN: 978-2746218734

[2] R. Karri, J. rajendran, K. Rosenfeld, M. Tehranipoor. *Trustworthy Hardware: Identifying and Classifying Hardware Trojans*. IEEE Computer, Octobre 2010.

[3] L. Bossuet, B. Le Gal. *Site internet collaborative pour l'enseignement de la sécurité matérielle*. 7ème Colloque sur l'Enseignement des Technologies et des Sciences de l'Information et des Systèmes, CETSIS, Bruxelles, Belgique, 27-29 Octobre 2008.

[4] National Institute of Standards and Technology (NIST). FIPS-197: Advanced Encryption Standard, November 2001.

[5] Kerckhoffs Auguste, *La cryptographie militaire*, Journal des sciences militaires, vol. IX, pp. 538, pp161-191, Janvier 1883.

[6] P. Kocher, J. Jaffe, B. Jun. *Differential Power Analysis*, in the proceedings of Advances in Cryptology, CRYPTO99, Lecture Notes in Computer Science, vol 1666, pp 398-412, Santa-Barbara,USA, August 1999, Springer-Verlag, 1999.

[7] M. Selmi, *Mise en place d'une attaque DPA ciblant un FPGA SRAM*, Matser Thesis, Ecole Supérieure des Communications, Tunis, Juin 2006.

[8] S. B. Örs, E. Oswald and B. Preneel. *Power-Analysis Attacks on an FPGA--First Experimental Results*, Proceedings of Cryptographic Hardware and Embedded Systems – CHES 2003, 5th International Workshop Cologne, Germany, September 8–10, 2003, pp. 35-50, LNCS 2779.

[9] F. Standaert, S. B. Ors, and B. Preneel. *Power Analysis attack on anFPGA implementation of AES*, In Proceedings of Cryptographic Hardwareand Embedded Systems - CHES, Marc Joye, Jean-Jacques Quisquater (Eds.),Lecture Notes in Computer Science (LNCS), Springer-Verlag, pp. 30-44, 2004.

[10] E. Brier, c. Clavier, F. Olivier, *Correlation Power Analysis with a Leakage Model*, in Proceeding of Internetional Conference of Cryptographic Hardware and Embedded Systems, CHES04, Lecture Notes in Computer Science, vol. 3156, p. 135-152, Springer, 2004.

[11] N. Kamoun, L. Bossuet, A. Gazel. *Experimental Implementation of DPA Attacks on AES Design with Flash-based FPGA Technology*, in the Proceeding of the Sixth IEEE International Multi-Conference on Systems Signals and Devices, SSD 2009, Djerba, Tunisa, March 2009.

***Lilian Bossuet**, âgé de 36 ans, ingénieur ENSEA, diplômé de l'INSA de Rennes, ancien élève de l'Ecole Normale Supérieure de Cachan et agrégé d'électronique, a fait sa thèse au Lab-STICC, Laboratoire en Sciences et Techniques de l'Information de la Communication et de la Connaissance, à l'Université de Bretagne Sud. Après un séjour dans l'équipe du Professeur Wayne Burleson à l'Université du Massachusetts aux USA, il est recruté comme Maître de Conférences à l'Institut Polytechnique de Bordeaux en 2005 et rejoins le laboratoire IMS et l'ENSEIRB-MATMECA. Dans cette école d'ingénieur il créa et dirigea le département Systèmes Electroniques Embarqués. En 2010, il obtient sont Habilitation à Diriger des Recherches de l'Université de Bordeaux. La même année, il devient titulaire d'une chaire CNRS-Université spécialisée en cryptographie appliquée et sécurité des systèmes embarqués et rejoins le Laboratoire Hubert Curien et Telecom Saint-Etienne. Ecole dans laquelle il enseigne l'électronique numérique et la sécurité matérielle. Il est de plus responsable pédagogique de la thématique « électronique » de cette école.*