Techniques d'estimation d'entropie efficaces pour l'attaque par analyse d'information mutuelle

Alexandre VENELLI[*,**]

[*] *IML - ERISCS Université de la Méditerranée,*
Case 907, 163 Avenue de Luminy, 13288 Marseille Cedex 09, FRANCE
[**] *Vault-IC France, an Inside Secure Company*
Avenue de la Victoire, Z.I. Rousset, 13790 Rousset, FRANCE
avenelli@insidefr.com

RÉSUMÉ. *L'attaque par analyse de consommation de courant utilisant le coefficient de Pearson (CPA, Correlation Power Analysis) est l'attaque par canaux cachés la plus utilisée en pratique. Néanmoins, le coefficient de corrélation de Pearson ne peut mesurer que les relations linéaires entre deux variables alors qu'un autre outil statistique comme l'Information Mutuelle (IM) enregistre les relations linéaires et non-linéaires. Une attaque utilisant l'IM a d'ailleurs été proposée et permet de généraliser la CPA en tenant compte de toutes les relations possibles entre variables. Cette attaque est appelée MIA (Mutual Information Analysis). Nous présentons dans cet article deux manières d'améliorer les résultats d'une attaque MIA. Nous introduisons d'abord la notion d'information mutuelle généralisée et son application à notre contexte. Nous étudions ensuite une amélioration de l'estimation d'entropie utilisant les B-splines.*

ABSTRACT. *The Correlation Power Analysis (CPA) is probably the most used side-channel attack. However, the Pearson correlation coefficient used in the CPA measures only linear statistical dependencies where the Mutual Information (MI) takes into account both linear and nonlinear dependencies. Recently, an attack that uses MI has been proposed and has generalized CPA by recording every possible relations between variables. The authors call this attack Mutual Information Analysis (MIA). We present two ways of improvement of MIA results. We introduce generalized mutual information and its application in our context. Then, we improve MIA results using B-splines for entropy estimation.*

MOTS-CLÉS : *Estimation d'entropie, Attaque par analyse de courant, Analyse d'information mutuelle*

KEYWORDS: *Entropy estimation, Power analysis, Mutual information analysis*

1. Introduction

Les algorithmes cryptographiques, qui sont considérés comme surs grâce à des preuves mathématiques, ne le sont plus forcément une fois implémentés en pratique. Ces algorithmes sont codés dans des composants qui interagissent et sont influencés par l'environnement dans lequel ils se trouvent. Les composants électroniques, comme les cartes à puces, consomment du courant et émettent des radiations lors de leur fonctionnement. Ces composants réagissent aussi aux changements de température et aux différents champs magnétiques qui les entourent. Ces interactions dues à l'environnement peuvent être contrôlées et enregistrées par un attaquant pour réaliser des attaques dites par canaux cachés.

L'attaque par analyse différentielle de courant (DPA de l'anglais *Differential Power Analysis*), introduite dans (Kocher *et al.*, 1999), est une des attaques par canaux cachés les plus connues. L'attaque DPA exploite des différences statistiques dans un grand ensemble de mesures de consommation de courant enregistrées par un attaquant sur un composant. Grâce à ces différences, il peut déduire le secret utilisé dans l'algorithme cryptographique implémenté dans le composant. Pour cela, l'attaque utilise une fonction de sélection qui permet de partitionner les mesures de consommation en deux ensembles. Cette fonction simule un calcul intermédiaire effectué par le composant où des parties du secret sont manipulées. L'attaque DPA consiste ensuite à utiliser la distance entre les moyennes des deux ensembles de mesures pour faire apparaître un pic lorsque l'attaquant suppose la bonne valeur du secret.

En 2004, Brier et al. (Brier *et al.*, 2004) proposent l'attaque CPA (de l'anglais *Correlation Power Analysis*). Elle utilise un test statistique différent de celui de la DPA, le coefficient de corrélation de Pearson. Ce coefficient de corrélation semble donner les meilleurs résultats sur la grande majorité des composants basés sur la technologie CMOS (de l'anglais *Complementary Metal Oxyde Semiconductor*). Ce facteur mesure les relations linéaires entre les mesures de consommation de courant et une variable de l'algorithme attaqué. Batina et al. (Batina *et al.*, 2008) proposent d'utiliser un test statistique non-paramétrique afin de mesurer les relations qui s'éloignent du linéaire. Ils montrent que l'utilisation du coefficient de Spearman, l'équivalent non-paramétrique du coefficient de Pearson, peut être intéressante dans de tels cas.

Gierlichs et al. (Gierlichs *et al.*, 2008) présentent en 2008 l'utilisation de l'information mutuelle dans le contexte des attaques par canaux cachés. L'attaque, que les auteurs appellent MIA (de l'anglais *Mutual Information Analysis*), est plus générique que les précédentes. En effet, pour mesurer l'information mutuelle nous n'avons besoin d'aucunes suppositions sur les relations statistiques entre la consommation de courant et la fonction de sélection. La consommation de la plupart des composants CMOS semble pouvoir être modélisée par un modèle linéaire en le poids de Hamming des instructions traitées par le composant à un instant donné. De ce fait, l'attaque CPA obtient souvent de meilleurs résultats que l'attaque MIA (Moradi *et al.*, 2009; Prouff *et al.*, 2009; Veyrat-Charvillon *et al.*, 2009). Sur d'autres types de composants utilisant une logique équilibrée, notamment la technologie Wave Dynamic Differential Logic

(Tiri *et al.*, 2005) où le modèle linéaire en le poids de Hamming est moins apparent, l'attaque MIA semble prometteuse (Gierlichs *et al.*, 2008).

L'attaque MIA, telle que présentée dans (Gierlichs *et al.*, 2008), peut être nettement améliorée dans la mise en œuvre de l'algorithme afin d'être compétitive avec l'attaque CPA même sur des composants CMOS. Nous présentons dans cet article deux directions afin d'améliorer l'attaque MIA. L'attaque peut aussi être améliorée en modifiant l'algorithme d'attaque en lui-même comme présenté dans (Maghrebi *et al.*, 2010) qui propose une généralisation de la MIA. Nous introduisons la notion d'information mutuelle généralisée (Pompe *et al.*, 1998; Pompe *et al.*, 1993; Pompe *et al.*, 1995) dans le contexte des attaques par canaux cachés. Nous obtenons des résultats toujours inférieurs à l'attaque CPA bien que meilleurs par rapport à l'attaque MIA classique. Dans le cadre des attaques par canaux cachés, il faut noter qu'une attaque est dite meilleure si elle permet de retrouver le secret en utilisant le moins possible d'enregistrements du canal caché. Cette notion d'efficacité des attaques est généralement mesurée grâce à deux métriques proposées dans la littérature : *guessed entropy* et *first-order success rate*. Nous les présentons en détail dans cet article. Nous étudions ensuite l'utilisation des B-splines en tant qu'estimateurs de densités de probabilités afin d'améliorer la précision du calcul d'information mutuelle. Par construction, les B-splines permettent de prendre en compte le bruit dû à la mesure des courbes de consommation sur le composant. Ce bruit peut réduire nettement l'efficacité des attaques par canaux cachés. Nous évaluons l'efficacité apportée par les B-splines grâce à des résultats expérimentaux sur composant. Nous appliquons aussi la méthode de B-splines au test de Cramér-von-Mises récemment proposé (Veyrat-Charvillon *et al.*, 2009).

La section 2 introduit plus en détails les différents types d'attaques par analyse de courant. La section 3 rappelle quelques notions de théorie de l'information ainsi que la notion d'information mutuelle généralisée. Nous rappelons aussi le principe d'une attaque par analyse d'information mutuelle dans la section 4. Dans la section 5 nous étudions quelques méthodes classiques d'estimation de densités de probabilités. Nous donnons l'algorithme permettant le calcul de l'information mutuelle généralisée dans la section 6. La section 7 présente l'utilisation des B-splines en tant qu'estimateurs de densités de probabilités ainsi que leur intérêt dans le contexte des attaques par canaux cachés. Des résultats expérimentaux sont présentés dans la section 8 et la section 9 conclut l'article.

2. Attaques par analyse de courant

Les attaques par analyse de courant appartiennent à l'ensemble des attaques par canaux cachés qui font elles mêmes parties des attaques matérielles. Comme leur nom le suggère, les attaques matérielles ont pour but de tester la bonne mise en pratique d'un algorithme cryptographique par un ingénieur. L'attaquant peut soit observer de manière passive le comportement d'une implémentation, soit de manière semi-passive

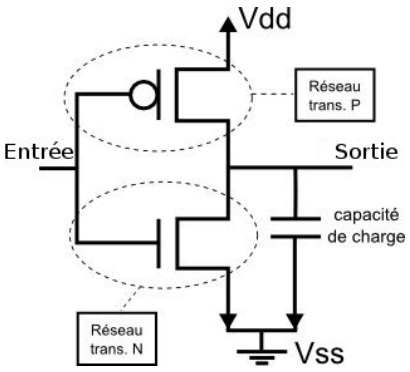


Figure 1 – Structure CMOS de base : inverseur

enregistrer des signaux internes à une implémentation, e.g. signaux de consommation d'un bus de données d'une carte à puce, soit de manière active en injectant des fautes.

Surveiller la consommation de courant d'une carte à puce ou ses émissions électromagnétiques a un avantage conséquent comparé aux autres attaques matérielles : les hypothèses sur l'accès à l'appareil attaqué sont réduites. De plus, grâce aux tests statistiques utilisés dans ce type d'attaques, l'attaquant n'a besoin que de très peu d'informations sur l'implémentation à attaquer. On distingue deux grandes familles d'attaques par analyse de courant : les attaques par analyse élémentaire de consommation (en anglais, *Simple Power Analysis*, SPA) et les attaques par analyse différentielle de consommation (DPA).

2.1. *Analyse élémentaire*

La grande majorité des cartes à puces utilisent la technologie *Complementary Metal Oxyde Semiconductor* (CMOS) car elle est peu chère et efficace. La Fig. 1 montre un inverseur CMOS, une structure de base d'un circuit CMOS. On remarque que l'inverseur est constitué de réseaux de transistors *P-channel Metal Oxyde Semiconductor* (PMOS) et *N-channel Metal Oxyde Semiconductor* (NMOS). Lorsqu'aucune opération n'est effectuée, la tension est soit de V_{dd} soit de V_{ss} au niveau de l'entrée et de la sortie. Néanmoins, lors d'une transition de l'entrée de V_{dd} vers V_{ss}, ou vice-versa, il y a durant un court instant un courant de court circuit. De plus à cet instant, les capacités de charges, comme les bus ou les portes logiques, sont chargées ou déchargées.

La consommation de courant d'un composant s'observe en mesurant la différence de tension divisée par la résistance aux bornes d'une résistance montée en série entre le V_{ss} du composant et une alimentation externe. On utilise ensuite un oscilloscope pour numériser le courant et l'enregistrer sur un ordinateur.

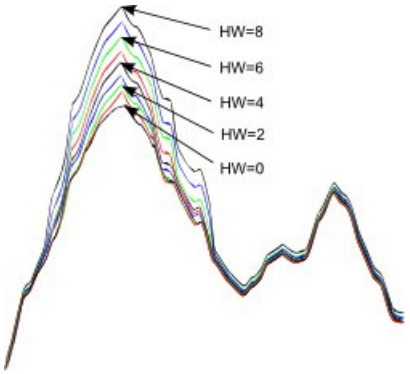


Figure 2 – La consommation de courant est liée au poids de Hamming des données traitées.

Messerges et al. (Messerges *et al.*, 1999a) observent une relation linéaire entre le nombre de bits à 1 présents dans un bus ou un registre interne à un instant t et la consommation de courant du composant à ce même instant. Ce modèle a d'ailleurs été étendu en considérant la distance de Hamming entre deux états consécutifs du composant (Brier *et al.*, 2004), au lieu du poids de Hamming. On modélise alors la consommation de courant $\mathcal{C}$ à un instant t par :

$$\mathcal{C}(t) = a \cdot HW(M \oplus R) + b, \qquad [1]$$

où a est une constante liée au composant utilisé, $HW(M \oplus R)$ est la distance de Hamming entre l'état actuel M d'un registre interne ou d'un bus et l'état précédent R. Enfin, b correspond à un bruit gaussien, plus particulièrement une somme de bruits provenant de différentes sources : bruit intrinsèque au composant, bruit dû à la mesure de l'oscilloscope, etc. D'après le théorème central limite, cette somme de bruits tend vers une distribution gaussienne. La Fig. 2 illustre une relation approximativement linéaire entre la consommation de courant et le poids de Hamming d'une donnée de 8 bits transférée dans un registre interne du composant.

De nombreux algorithmes cryptographiques symétriques et asymétriques ont été attaqués par des attaques SPA. On peut citer l'attaque sur le DES de Messerges et al. (Messerges *et al.*, 1999a) ou celle de Mangard (Mangard, 2002) sur l'AES. Du côté des algorithmes asymétriques, à la fois la signature RSA (Messerges *et al.*, 1999b) et la multiplication scalaire sur courbe elliptique (Oswald, 2002) ont été attaquées par SPA. Dans le cadre des attaques SPA, les cryptosystèmes asymétriques sont les moins évidents à protéger alors qu'une implémentation soignée des algorithmes de DES et AES permet d'éviter ces attaques. Ces cryptosystèmes sont beaucoup plus sensibles aux attaques par analyse différentielle que nous étudions ci-dessous.

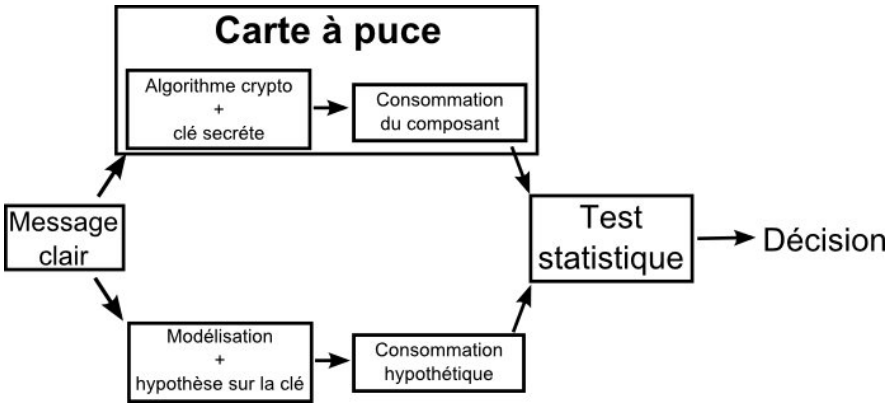


Figure 3 – Principe d'une analyse différentielle

2.2. *Analyse différentielle*

Les attaques par analyse différentielle suivent ces grands principes :

– l'attaquant utilise un grand nombre de courbes de consommation de courant du fonctionnement d'un algorithme cryptographique ayant une clé secrète fixée sur un composant,

– il cherche à attaquer une variable intermédiaire de cet algorithme cryptographique, variable qui doit dépendre de la valeur de la clé secrète,

– il émet des hypothèses sur la partie de la clé secrète utilisée dans le calcul de la variable intermédiaire visée,

– il applique un test statistique afin de connaître la validité de son hypothèse de clé.

Ce processus générique aux attaques DPA est résumé par la Fig. 3.

Nous donnons à présent une description plus formelle. Soit K une variable aléatoire représentant une partie du secret. Soit X une variable aléatoire représentant une partie de la donnée d'entrée, ou de sortie, de l'algorithme cryptographique. On suppose que l'attaquant s'intéresse à une valeur intermédiaire calculée par la fonction F, appelée fonction de sélection, qui prend X et K en paramètres. Soit L la variable aléatoire représentant la fuite d'information du canal caché produite par le calcul de $F(X, K)$. En pratique, l'attaquant est seulement capable d'obtenir N réalisations de la variable aléatoire L, notées $V_L = (l_1, \ldots, l_N)$, à partir de N valeurs d'entrées X différentes, notées $V_X = (x_1, \ldots, x_N)$. Grâce à un distingueur D, il combine ces deux vecteurs avec une hypothèse sur le secret k'. Si le distingueur D est pertinent et si le vecteur V_L donne assez d'informations sur $F(X, K)$, alors la valeur correcte k prise par K peut être retrouvée. Nous détaillons dans la suite deux des principaux distingueurs proposés dans la littérature.

2.2.1. *Distance de moyennes*

L'attaque DPA historique de Kocher et al. (Kocher *et al.*, 1999) propose d'utiliser comme test statistique la distance de moyennes. Une fois l'hypothèse de clé réalisée par l'attaquant, celui-ci partitionne en deux ensembles la sortie de la fonction de sélection suivant sa valeur. Soit N courbes de consommation de courant acquises sur composant par l'attaquant. Pour chaque acquisition, il utilise un message clair aléatoire x_i. Si on suppose, par exemple, que la fonction de sélection F retourne le bit le plus significatif de la valeur intermédiaire visée. On appelle alors l'attaque, DPA mono-bit. Soit k' l'hypothèse sur le secret. L'attaquant forme deux ensembles :

$$G_0 = \{L \mid F(x_i, k') = 0\} \quad \text{et} \quad G_1 = \{L \mid F(x_i, k') = 1\}$$

où k' est l'hypothèse de clé réalisée par l'attaquant. On calcule ensuite la courbe de distance des moyennes $\Delta_{k'}(t)$ entre les deux partitions G_0 et G_1. En principe, la courbe $\Delta_{k'}(t)$ est différente de zéro lorsque k' correspond à la vrai sous-clé du composant car on trouve une relation entre la consommation de courant et l'état du bit le plus significatif de la valeur intermédiaire visée. De plus, cette déviation de zéro apparaît à l'instant t_0 où la vraie clé est manipulée par le composant. On définit la courbe de distance de moyennes $\Delta_{k'}(t)$ comme :

$$\Delta_{k'}(t) = \frac{\sum_{l \in G_0} l}{|G_0|} - \frac{\sum_{l \in G_1} l}{|G_1|}.$$

L'attaquant calcule ensuite $|K|$ courbes de différences $\Delta_{k'}(t)$ pour chaque valeur possible de la sous-clé k'. Il décide que l'hypothèse k' est plus vraisemblablement la vraie en considérant le plus grand pic sur les courbes $|\Delta_{k'}(t)|$. La qualité d'une attaque DPA, qui est liée à la qualité des courbes de différences $\Delta_{k'}(t)$, dépend principalement du nombre N de mesures de courbes de consommations réalisée par l'attaquant.

2.2.2. *Coefficient de corrélation de Pearson*

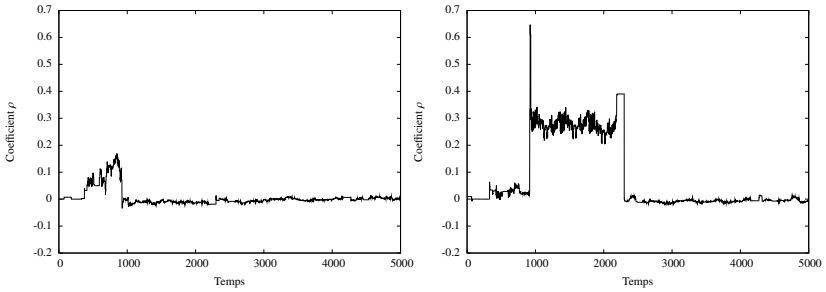
Le coefficient de Pearson est un test connu dans le domaine des statistiques. Il permet de mesurer les relations linéaires entre deux variables X et Y. Ce coefficient est définit ainsi :

$$\rho(X, Y) = \frac{\text{cov}(X, Y)}{\sqrt{\text{var}(X) \cdot \text{var}(Y)}} = \frac{E(X \cdot Y) - E(X) \cdot E(Y)}{\sqrt{\text{var}(X) \cdot \text{var}(Y)}}.$$

La formule donne un coefficient qui a une valeur comprise entre -1 et $+1$. Soit k' une hypothèse sur le secret. Si on considère N observations des variables X et L, on obtient :

$$\rho_{k'}(X, L) = \frac{N \sum_i l_i F(x_i, k') - (\sum_i l_i \sum_i F(x_i, k'))}{\sqrt{N \sum_i l_i^2 - (\sum_i l_i)^2} \sqrt{N \sum_i F(x_i, k')^2 - (\sum_i F(x_i, k'))^2}}.$$

Si les valeurs de L augmentent en même temps que celles de X, les points (x_i, l_i) forment une ligne droite de pente positive, alors $\rho_{k'}(X, L) = +1$. Si les points forment



(a) $k' = \texttt{0x8a}$ (b) $k' = \texttt{0x9a}$, vraie clé

(c) $k' = \texttt{0x20}$ (d) $k' = \texttt{0xa6}$

Figure 4 – Attaque CPA en sortie de Sbox au premier tour sur les premiers 8 bits d'une clé AES. Courbes CPA pour les hypothèses de clés `0x8a`, `0x9a`, `0x20` et `0xa6` sachant que la vraie clé vaut `0x9a`.

une ligne droite de pente négative, alors $\rho_{k'}(X, L) = -1$. S'il n'y a aucune relations entre X et L alors $\rho_{k'}(X, L) = 0$.

On appelle CPA (*Correlation Power Analysis*) l'utilisation de ce test statistique dans le contexte des attaques par canaux cachés (Brier *et al.*, 2004). On obtient souvent des résultats d'attaques meilleurs que la DPA avec distance des moyennes. On entend par résultats meilleurs qu'un attaquant a besoin de moins de mesures de consommation pour que la vraie valeur de la clé apparaisse plus clairement. La Fig. 4 montre un exemple de courbes CPA pour différentes hypothèses de clés lors d'une attaque CPA sur l'algorithme AES.

Nous parlons dans le reste de l'article de l'utilisation d'un test statistique proposé par Gierlichs et al. (Gierlichs *et al.*, 2008) dans le cadre des attaques par canaux cachés : l'information mutuelle. L'attaque correspondante est appelée MIA (de l'anglais *Mutual Information Analysis*). Il s'agit d'une alternative intéressante à la CPA car l'attaquant n'a pas besoin de supposer un modèle de consommation particulier, notamment le poids de Hamming [1]. En effet, l'information mutuelle mesure à la fois les relations linéaires et non-linéaires entre variables alors que le coefficient de Pearson

n'enregistre que les relations linéaires. L'utilisation de l'information mutuelle semble, en théorie, particulièrement intéressante puisque l'attaque prend désormais en compte plus d'informations dans la relation entre les variables. Néanmoins en pratique les attaques MIA sont, dans la majorité des cas, moins performantes que les attaques CPA. Nous introduisons dans l'article différentes techniques permettant d'améliorer les résultats d'attaques MIA. Tout d'abord, nous précisons quelques notions de théorie de l'information nécessaires par la suite.

3. Rappels de théorie de l'information

3.1. *Information mutuelle classique*

En théorie de l'information, l'Information Mutuelle (IM) est définie comme une mesure de la dépendance mutuelle entre deux variables. Contrairement au coefficient de corrélation de Pearson, elle est aussi sensible aux relations entre variables qui n'apparaissent pas dans la covariance.

Soit X une variable aléatoire ayant un ensemble fini de M_X états possibles X_i avec $i = 1, \ldots, M_X$. Soit $\mathbb{P}_X$ la distribution de probabilités de X. L'entropie de Shannon de X, notée $H(X)$ ou $H(\mathbb{P}_X)$, est définie comme :

$$H(X) = -\sum_{i=1}^{M_X} p(X_i) \log(p(X_i)), \quad [2]$$

avec $p(X_i)$ la probabilité de l'état X_i.

L'entropie conjointe $H(X, Y)$ de deux variables aléatoires X et Y est définie de la même manière :

$$H(X, Y) = -\sum_{i=1,j=1}^{M_X, M_Y} p(X_i, Y_j) \log(p(X_i, Y_j)). \quad [3]$$

L'entropie conditionnelle $H(X \mid Y)$ indique l'incertitude liée à X connaissant Y. Elle est définie comme :

$$H(X \mid Y_j) = -\sum_{i=1}^{M_X} p(X_i \mid Y_j) \log(p(X_i \mid Y_j)), \quad [4]$$

$$H(X \mid Y) = \sum_{j=1}^{M_Y} p(Y_j) H(X \mid Y_j). \quad [5]$$

L'information mutuelle $I(X; Y)$ est définie par :

$$I(X; Y) = H(X) - H(X \mid Y), \quad [6]$$

$$\text{ou} \quad I(X; Y) = H(X) + H(Y) - H(X, Y). \quad [7]$$

3.2. *Information mutuelle généralisée*

Soit X une variable aléatoire discrète définie comme ci-dessus. L'entropie de Rényi d'ordre α est :

$$\mathrm{H}_\alpha(X) = \begin{cases} \frac{1}{1-\alpha} \log \sum_{i=1}^{M_X} p(X_i)^\alpha & \text{pour } \alpha \geq 0, \alpha \neq 1 \\ -\sum_{i=1}^{M_X} p(X_i) \log p(X_i) & \text{pour } \alpha = 1. \end{cases}$$

On peut alors noter l'entropie de Shannon $\mathrm{H}_1(X)$. Grâce à ces définitions d'entropie de Rényi, on peut introduire la quantité :

$$\mathrm{I}_\alpha(X;Y) = \mathrm{H}_\alpha(X) + \mathrm{H}_\alpha(Y) - \mathrm{H}_\alpha(X,Y).$$

L'information mutuelle I_α a la propriété suivante :

$$\mathrm{I}_\alpha \geq 0 \quad \text{si et seulement si } \alpha = 0 \text{ ou } 1.$$

La valeur I_α ne correspond à la définition usuelle d'information mutuelle que dans ces deux cas.

L'entropie de Rényi H_2 est aussi appelée entropie de collision. Elle correspond à l'opposé du logarithme de la probabilité que deux variables aléatoires indépendantes, ayant la même distribution de probabilité, aient la même valeur. Les valeurs les plus probables ont plus tendance à entrer en collision et ont donc plus d'effets sur l'entropie de collision que sur l'entropie de Shannon.

Grâce au théorème (Pompe *et al.*, 1993, Chapitre 3, Basic Theorem), les auteurs utilisent l'entropie de collision H_2 et appellent $\mathrm{I}_2(X;Y)$ Information Mutuelle Généralisée (ou GMI de l'anglais *Generalized Mutual Information*) avec une variable, X ou Y, suivant la distribution uniforme. La GMI et l'IM ont plusieurs propriétés en commun (Pompe *et al.*, 1998). Elles sont notamment toutes les deux positives et elles détectent l'indépendance entre deux variables. La GMI a donc les propriétés nécessaires pour notre étude. Nous discutons dans la section 6 de l'application de la GMI à notre étude et de l'algorithme pour la calculer.

4. Attaque par analyse d'information mutuelle

Nous présentons brièvement l'attaque par analyse d'information mutuelle proposée par Gierlichs et al. (Gierlichs *et al.*, 2008).

Pour comprendre la définition d'une fonction de sélection F, nous prenons comme exemple l'étude de l'algorithme DES. Soit x une partie du message clair et k une partie du secret. Des exemples, très utilisés en pratique, de valeurs intermédiaires attaquées sont :

– $F_1(x,k) = HW(Reg_L \oplus Reg_R)$ tel que Reg_L et Reg_R sont les valeurs sur 4 bits des registres L et R du DES, d'où $F_1(x,k) = i$, pour $i \in \{0 \ldots 4\}$,

– $F_2(x,k) = HW(\text{SBox}(x \oplus k))$ tel que la sortie de la SBox du DES soit sur 4 bits, d'où $F_2(x,k) = i$, pour $i \in \{0 \dots 4\}$,

– $F_3(x,k) = \text{SBox}(x \oplus k)$ tel que $F_3(x,k) = i$, pour $i \in \{0 \dots 15\}$.

Soit $L(t)$ une variable aléatoire ayant pour valeurs des observations physiques du circuit électronique étudié à un instant t.

Supposons qu'un attaquant acquière N courbes de consommation de courant du circuit $\{l_{x_1}, \dots, l_{x_N}\}$ liées aux messages clairs $\{x_1, \dots, x_N\}$ et avec une clé secrète k fixée. Soit m la valeur maximale que peut prendre la fonction choisie attaquant une valeur intermédiaire de l'algorithme. Par exemple, pour $F_1(x,k)$ nous avons $m = 4$. Le prototype d'une attaque par analyse d'information mutuelle est le suivant :

1) Nous estimons l'entropie $H(L(t))$ à chaque temps t.

2) Pour chaque hypothèse k' sur la clé secrète, nous appliquons la fonction F. Celle-ci a comme sortie une valeur comprise dans $\{0, \dots, m\}$. Comme vu précédemment, F prend en paramètre une partie de clé secrète k' et une partie du message clair. Nous appliquons donc N fois la fonction F pour chaque message clair x_i, $i = 1, \dots, N$.

3) Nous partitionnons les messages clairs x_i suivant la valeur de sortie de F dans les ensembles :

$$\mathcal{H}_j^{k'} = \{x_i \mid F(x_i, k') = j\} \text{ avec } j \in \{0, \dots, m\}\,.$$

Nous avons de manière analogue une partition sur les observations :

$$\mathcal{G}_j^{k'} = \left\{ l_{x_i} \mid x_i \in \mathcal{H}_j^{k'} \right\}.$$

4) Pour $j \in \{0, \dots, m\}$, et à chaque instant t, nous estimons l'entropie $H(L(t) \mid F)$. Nous calculons en premier les entropies $H(L(t) \mid F = j)$ pour chaque $j \in \{0, \dots, m\}$ en utilisant [4] et les partitions $\mathcal{G}_j^{k'}$. Nous utilisons ensuite [5] pour trouver l'entropie conditionnelle $H(L(t) \mid F)$.

5) À chaque instant t, nous calculons pour une hypothèse de clé k' l'information mutuelle grâce à la définition [6] :

$$I_{k'}(L;F)(t) = H(L(t)) - H(L(t) \mid F).$$

Gierlichs et al. (Gierlichs *et al.*, 2008) démontrent que l'information mutuelle $I_{k'=k}(L;F)$ doit être maximum pour une hypothèse de clé k' correcte :

$$I_{k'=k}(L;F)(t) = \max_{k',t}(I_{k'}(L;F)(t)).$$

L'estimation de densités de probabilité est au centre de l'attaque par analyse d'information mutuelle. Plus généralement, une estimation précise permet d'améliorer l'efficacité des attaques par canaux cachés (Standaert *et al.*, 2009).

5. Techniques d'estimation d'entropie

Il existe deux approches principales à l'estimation d'entropie : l'estimation paramétrique et non-paramétrique. Nous nous contentons d'étudier des méthodes non-paramétriques. Les méthodes paramétriques supposent plusieurs propriétés à propos des fonctions de régression qui décrivent la relation entre des variables. En effet, les densités de probabilité considèrent que les données proviennent d'une loi de probabilité connue, comme la loi normale. Les paramètres des fonctions de densité sont ainsi optimisés en faisant correspondre le modèle provenant de la loi de probabilité aux données. L'estimation non-paramétrique, au contraire, est une méthode où le choix des paramètres se fait sans aucune supposition sur la loi de probabilité des données. Nous présentons par la suite différentes méthodes d'estimation non-paramétrique efficaces dans le contexte d'attaques par canaux auxiliaires.

5.1. *Utilisation d'histogrammes*

Toutes les définitions de la section 3 supposent la connaissance de la loi de probabilité des données. Néanmoins, en pratique, les probabilités sont inconnues et doivent être estimées à partir de mesures. La méthode la plus simple et la plus utilisée est l'utilisation d'histogrammes.

Soit un ensemble de N mesures d'une variable aléatoire X. Les données sont découpées en B partitions. Ces partitions sont définies grâce à B intervalles $a_i = [o + i.h, o + (i + 1).h]$ tel que o est la valeur d'origine, h est la longueur des partitions et $i = 0, \ldots, B - 1$. On note k_i le nombre de mesures qui appartiennent à l'intervalle a_i. Les probabilités $p(a_i)$ sont ensuite approximées grâce aux fréquences :

$$p(a_i) = \frac{k_i}{N}.$$

À partir de ces approximations, on peut calculer les entropies et l'information mutuelle. Le choix du nombre de partitions B est crucial. En effet, il s'agit d'un paramètre de lissage de l'estimation (Fig. 5). Le nombre de partitions détermine à quel point l'approximation reflète la distribution idéale continue et à quel point le partitionnement correspond au traitement des données par le composant en pratique. L'estimation de densités à l'aide d'histogrammes se calcule très rapidement mais donne souvent des résultats très approximatifs du point de vue statistique.

5.2. *Estimation par noyau*

Il existe de nombreuses méthodes concurrentes à l'estimation par histogrammes. On s'intéresse à l'estimation par noyau (Moon *et al.*, 1995), aussi appelée méthode de Parzen (Parzen, 1962), une technique bien connue qui donne généralement de bons résultats. Cette méthode suppose que la densité de probabilité est assez lisse pour que

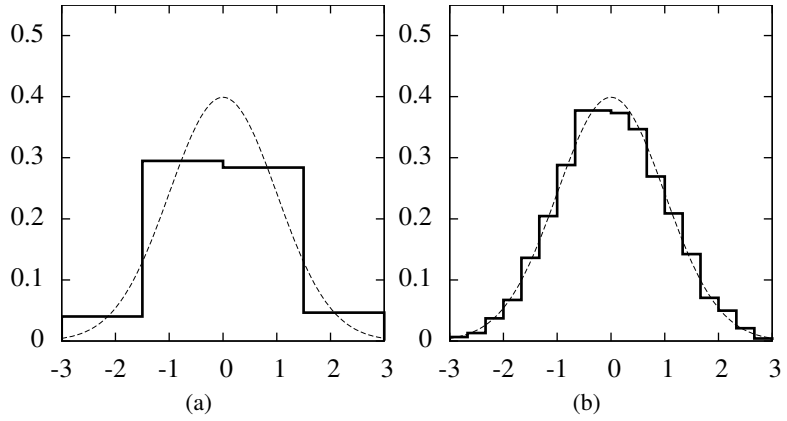


Figure 5 – Effet du nombre de partitions sur la correspondance entre l'estimation et la vraie loi de probabilité. Dans les deux figures, la ligne pointillée est une distribution gaussienne, la ligne pleine correspond à l'estimation. Figure 5a est une estimation avec 4 partitions. Figure 5b est une estimation avec 18 partitions.

les structures présentes en dessous d'une certaine fenêtre puissent être ignorées. Les noyaux se contentent essentiellement de mettre des poids aux distances de chaque point de l'échantillon par rapport à un point référence. Ces poids dépendent de la forme du noyau et de la fenêtre h utilisée. La méthode la plus simple consiste à estimer une densité à un point x en comptant le nombre de points contenus dans une boîte centrée en x de taille h et en divisant par son volume. Au lieu de simplement compter les points, les noyaux sont utilisés pour donner des poids dépendant de la distance entre les points. Un estimateur naïf $f(x)$, qui améliore tout de même l'estimation de la probabilité $p(x)$, peut s'écrire :

$$f(x) = \frac{1}{2Nh} \sum_{i=1}^{N} \Theta(h - |x - x_i|),$$

avec Θ la fonction de Heaviside définie par :

$$\Theta(z) = \begin{cases} 1 & \text{si} \quad z > 0 \\ 0 & \text{si} \quad z \leq 0. \end{cases}$$

Pour une définition plus générale, on note $K(x)$ le noyau. On définit alors un estimateur par noyau $f(x)$ comme :

$$f(x) = \frac{1}{Nh} \sum_{i=1}^{N} K\left(\frac{x - x_i}{h}\right). \qquad [8]$$

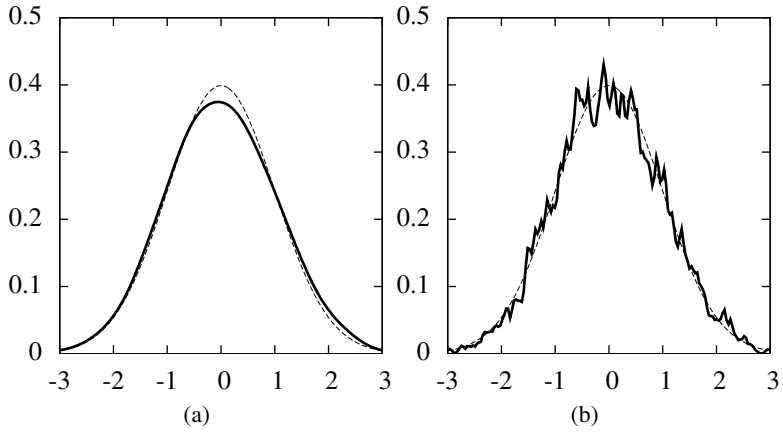


Figure 6 – Estimation par noyau utilisant le noyau de Heaviside avec différentes valeurs de fenêtres. Dans les deux figures, la ligne pointillée est une distribution gaussienne, la ligne pleine correspond à l'estimation. Figure 6a est une estimation avec pour fenêtre $h = 0.3$. Figure 6b est une estimation avec pour fenêtre $h = 0.03$.

Un exemple de noyau K souvent utilisé est le noyau gaussien, la fonction d'estimation est ainsi :

$$f(x) = \frac{1}{Nh\sqrt{2\pi}} \sum_{i=1}^{N} \exp\left(-\frac{(x - x_i)^2}{2h^2}\right).$$

Le noyau gaussien peut être vu comme le fait de placer de petites 'bosses' gaussiennes à chaque point x_i. L'estimation correspond alors à la somme des ces 'bosses'.

On se rappelle qu'un paramètre critique de l'estimation par histogrammes est le choix du nombre de partitions. Dans l'estimation par noyau, le choix de la valeur de la fenêtre h devient crucial. Si h est trop grand, l'estimation souffre de trop peu de précision alors que si h est trop petit, l'estimation a une trop grande variabilité statistique (Fig. 6).

Même si l'estimation par noyau donne des résultats plus précis que les histogrammes, elle requière une puissance de calcul importante. L'estimation par noyaux est particulièrement bien adaptée au calcul de l'entropie de Rényi comme nous l'étudions dans la section suivante.

6. Calcul efficace de l'information mutuelle généralisée

On peut obtenir une estimation simple et précise de l'entropie de Rényi en utilisant la méthode de Parzen vue précédemment [8].

L'entropie de Rényi d'ordre α d'une variable aléatoire X suivant la loi de probabilité p est définie par :

$$H_\alpha(X) = \frac{1}{1-\alpha} \log E\left[p^{\alpha-1}(X)\right],$$

avec E la fonction de calcul d'espérance. Avec les mêmes notations, l'entropie de Shannon de X est :

$$H_1(X) = -E\left[\log p(X)\right].$$

Lors de l'estimation de la mesure d'entropie à partir d'un nombre fini d'échantillons, on substitue l'espérance E par la moyenne et la probabilité p par la méthode de Parzen. L'entropie de Shannon devient alors :

$$\widehat{H}_1(X) = -\frac{1}{N}\sum_{j=1}^{N} \log\left[\frac{1}{Nh}\sum_{i=1}^{N} K\left(\frac{x-x_i}{h}\right)\right],$$

avec h la fenêtre du noyau.

De la même manière, l'entropie de Rényi est :

$$\widehat{H}_\alpha(X) = \frac{1}{1-\alpha} \log\left[\frac{1}{N^\alpha h}\sum_{j=1}^{N}\left(\sum_{i=1}^{N} K\left(\frac{x-x_i}{h}\right)\right)^{\alpha-1}\right].$$

D'après la section 3, $\widehat{H}_2(X)$ s'utilise comme l'entropie de Shannon :

$$\widehat{H}_2(X) = -\log\left[\frac{1}{N^2 h}\sum_{j=1}^{N}\left(\sum_{i=1}^{N} K\left(\frac{x-x_i}{h}\right)\right)\right].$$

L'entropie de Rényi est très intéressante du point de vue de l'estimation. En effet, on somme des puissances d'un noyau, ce qui est plus simple que l'utilisation de la méthode de Parzen avec Shannon. En combinant l'entropie de Rényi et la méthode de Parzen, on gagne en temps de calcul. Néanmoins, comparée à l'estimation par histogrammes, cette technique requière toujours beaucoup de calculs.

Nous adaptons à notre étude un des algorithmes proposé dans (Pompe *et al.*, 1995) permettant de calculer la GMI. Comme détaillé dans la section 3, l'algorithme requiert que l'une des deux variables aléatoires suivent la distribution uniforme ce qui est très rarement le cas en pratique. Pour améliorer l'efficacité de l'algorithme, nous appliquons la transformation en distribution uniforme aux deux variables. Soit X et Y deux variables aléatoires discrètes ayant le même nombre d'états n.

1) On utilise le rang des données triées pour obtenir une distribution uniforme à partir des états de X. On note les différents états possibles de X : X_i avec $i = 1, \ldots, N$. Soit les deux tableaux de taille N :

$$\begin{cases} \mathrm{TData}_X\left[i\right] = X_i & \text{pour } i \in \{1, \ldots, N\} \\ \mathrm{TIndex}_X\left[i\right] = i & \text{pour } i \in \{1, \ldots, N\} \end{cases}$$

2) On modifie un algorithme de tri classique pour qu'il réarrange le tableau d'index TIndex_X en fonction des valeurs du tableau TData_X.

3) Soit TRang_X un tableau de taille n tel que :

$$\mathrm{TRang}_X\left[\mathrm{TIndex}_X\left[i\right]\right] = i \quad \text{pour } i \in \{1, \ldots, N\}$$

On obtient alors une transformation en distribution uniforme.

4) Les trois premières étapes sont répétées pour la variable Y afin d'obtenir TRang_Y et TIndex_Y.

5) Un certain niveau de granularité ϵ est fixé (Pompe *et al.*, 1995). Dans notre cas, ϵ équivaut en quelque sorte au nombre de partitionnement d'une estimation par histogrammes.

6) Le principe de l'Algorithme 1 est de trouver les plus proches voisins d'un point tel que la distance n'excède pas ϵ.

Algorithme 1: Calcul de l'information mutuelle généralisée

Données : Nombre d'états N, tableau de rangs TRang_Y, tableau d'index TIndex_X, paramètre ϵ
Résultat : $I_2(X;Y)$
$N_{total} = (N-1)(N-2)/2$
$C_{1,\epsilon} = (2N-\epsilon)(\epsilon-1)/(N(N-1))$
$Somme = 0$
$i = 1$
tant que $i \leq n-1$ **faire**
  $idx_1 = \mathrm{TIndex}_X[i]$
  $j = i+1$
  tant que $j < (i+\epsilon)$ *et* $j < N$ **faire**
    $idx_2 = \mathrm{TIndex}_X[j]$
    $Somme = Somme + 1$
    si $|\mathrm{TRang}_Y[idx_1] - \mathrm{TRang}_Y[idx_2]| < \epsilon$ **alors**
      $Somme = Somme + 1$
    $j = j + 1$
  $i = i + 1$
$Somme = Somme/N_{total}$
retourner $\log(Somme/C_{1,\epsilon})$

Nous étudions dans la partie suivante un compromis entre efficacité et temps de calcul avec un estimateur de densités de probabilités utilisant des B-splines. Grâce aux B-splines, nous améliorons nettement les résultats donnés par l'estimation par histogrammes tout en conservant un faible temps de calcul.

7. Améliorer l'attaque par analyse d'information mutuelle grâce aux B-splines

7.1. *Introduction aux polynômes par morceaux et splines*

Soit X une variable à une dimension. Une fonction polynômiale par morceaux $f(X)$ s'obtient en divisant le domaine de X en intervalles contigus et en représentant f par un différent polynôme dans chaque intervalle. Figures 7a et 7b montrent des polynômes par morceaux simples. On préfère souvent des fonctions polynomiales plus lisses qui s'obtiennent en augmentant le degré des polynômes. Figure 7d correspond à une fonction polynômiale cubique, on l'appelle spline cubique. De manière générale, une spline de degré k avec des nœuds t_i, $i = 0, \ldots, m$ est un polynôme par morceaux de degré k et est deux fois continûment dérivable. Une spline cubique a pour degré $k = 4$. Ainsi, Fig. 7a est une spline d'ordre 1 et Fig. 7b est une spline d'ordre 2.

7.2. *Calcul des B-splines*

Nous introduisons les B-splines qui sont une généralisation des courbes de Bézier. Pour plus de détails sur les splines, le lecteur intéressé peut consulter (Deboor, 1978). Une présentation plus succincte des résultats de cette section a été publiée dans (Venelli, 2010).

Une courbe B-spline définie sur l'intervalle $[a, b]$ est caractérisée par :

– son degré d (ou ordre $k = d + 1$), tel que chaque morceau du polynôme par morceaux est de degré d ou moins,

– une séquence de $m + 1$ entiers, $t_0, \ldots, t_m$, appelé vecteur de nœuds, tel que $t_i \leq t_{i+1}, \forall i \in \{1, \ldots, m-1\}$,

– des points de contrôle, $b_0, \ldots, b_n$.

Une courbe B-spline est définie en tant que fonctions B-splines de base. La i-ème fonction de base de degré d, notée $B_{i,d}$, déterminée par le vecteur de nœuds $t_0, \ldots, t_m$ est définie par récurrence grâce à la formule de Cox-de Boor :

$$B_{i,0}(z) = \begin{cases} 1 & \text{si} \quad t_i \leq z < t_{i+1} \\ 0 & \text{sinon.} \end{cases} \qquad [9]$$

$$B_{i,d}(z) = \frac{z - t_i}{t_{i+d} - t_i} B_{i,d-1}(z) + \frac{t_{i+d+1} - z}{t_{i+d+1} - t_{i+1}} B_{i+1,d-1}(z), \qquad [10]$$

pour $i = 0, \ldots, n$ et $d \geq 1$.

La courbe B-spline de degré d avec points de contrôle $b_0, \ldots, b_n$ et nœuds $t_0, \ldots, t_m$ est définie par :

$$B(z) = \sum_{i=0}^{n} b_i B_{i,d}(z),$$

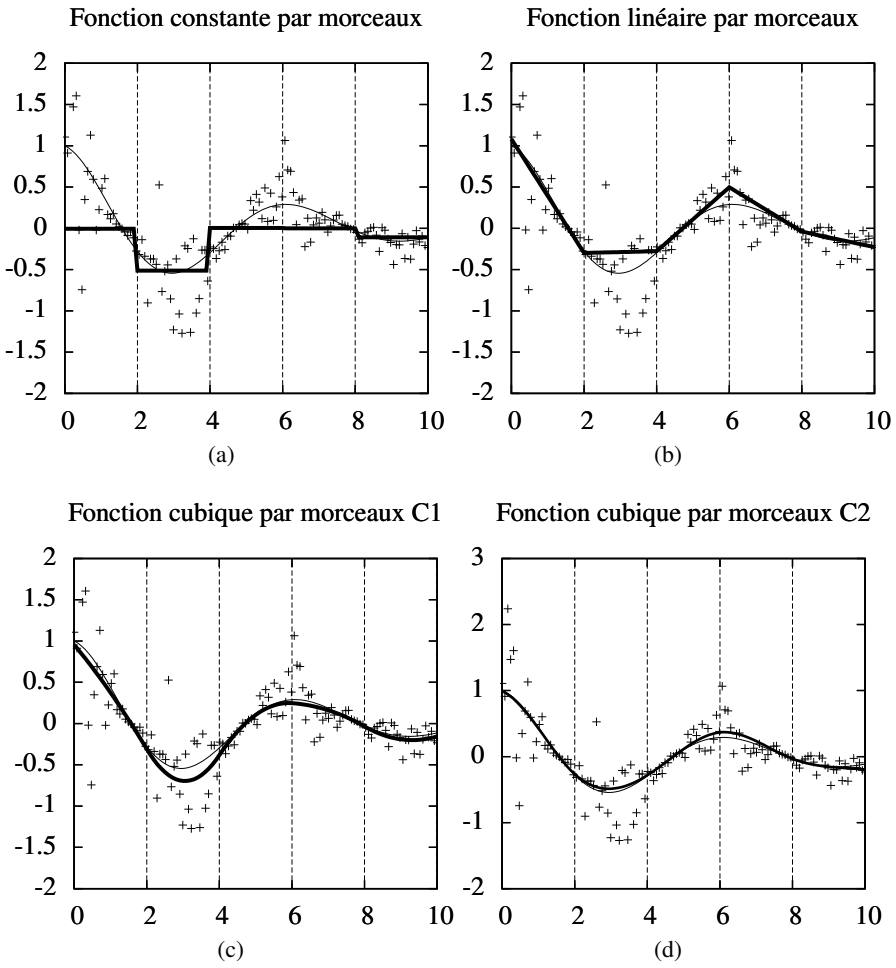


Figure 7 – Dans chacune des figures, les lignes pointillées correspondent à la position des nœuds. La ligne fine est la fonction $y(x) = \cos(x)\exp(-x/5)$. Les croix sont des données générées depuis la fonction $y(x)$ auxquelles on ajoute du bruit gaussien. La ligne épaisse représente l'estimation.

avec $B_{i,d}(z)$ la fonction B-spline de base définie précédemment.

Grâce à [10], on peut noter que $B_{i,d}(z)$ est non-nul sur l'intervalle $[t_i, t_{i+d+1}]$. Par exemple, une fonction B-spline de base cubique $B_{i,3}(z)$ est non-nulle sur l'intervalle $[t_i, t_{i+4}]$. On peut aussi remarquer que, si les nœuds ne sont pas répétés, la B-spline est nulle aux nœuds extrêmes t_i et t_{i+d+1}. Mais les nœuds peuvent être répétés dans la définition d'une B-spline. Si le vecteur de nœuds contient un nombre suffisant de

nœuds répétés alors une division par zéro peut apparaître. On suppose alors que $0/0 := 0$. Finalement, la propriété la plus importante pour notre étude est la partition de l'unité d'une courbe B-spline :

$$\sum_{i=0}^{n} B_{i,d}(z) = 1, \quad \forall z.$$

On peut ainsi facilement adapter les fonctions B-spline pour être des estimateurs de densités de probabilités (Daub *et al.*, 2004). Figure 8 montre des exemples de fonctions B-spline de base de différents degrés.

7.3. *Estimation de la densité de probabilité avec des B-splines*

Dans (Daub *et al.*, 2004), les auteurs comparent leur méthode avec d'autres techniques d'estimation de probabilités. Ils montrent, sur des données artificiellement générées, que l'estimation par B-splines offre des résultats en moyenne deux fois meilleurs qu'avec des histogrammes. Ils concluent qu'utiliser des degrés de splines trop élevés ne donnent pas forcément de meilleurs résultats qu'un ordre $k = 2$ ou $k = 3$. Nous utilisons ainsi pour la suite de l'article l'ordre $k = 3$.

L'inconvénient majeur de la technique d'estimation par histogrammes est que chaque donnée n'est affectée qu'à une seule partition. On perd ainsi de l'information sur les données situées à la limite de deux partitions. En effet, suivant le bruit lié à cette donnée lors de la mesure par exemple, celle-ci peut se retrouver arbitrairement dans l'une ou l'autre des partitions. L'idée principale de (Daub *et al.*, 2004) est de permettre à une donnée d'être dans plusieurs partitions à la fois en utilisant des fonctions B-splines.

On veut reproduire une approche par histogrammes en remplaçant le partitionnement naïf de l'intervalle de valeurs par un partitionnement plus évolué grâce aux fonctions B-splines. Dans ces deux types de partitionnement, l'axe des abscisses est découpé en un certain nombre d'intervalles où chaque point limitant l'intervalle s'appelle point d'arrêt. Pour changer la forme d'une courbe B-spline, on a précédemment vu qu'on peut modifier : l'ordre de la courbe, les points de contrôles ou le vecteur de nœuds. Le nombre de points d'arrêts est lié à ces valeurs grâce à la formule : $narret = n - k + 2$ avec n le nombre de points de contrôles, ou fonctions de base, et k l'ordre de la courbe. L'ordre de la courbe B-spline est généralement fixé au préalable. On peut donc modifier le vecteur de nœuds et le nombre de points d'arrêts pour que les fonctions B-splines se comportent comme des partitions d'un histogramme. En général, les courbes B-splines ne sont pas tangentes aux nœuds extrêmes. Pour notre étude, nous voulons que les B-splines soient non-nulles à ces extrémités. On veut que les fonctions B-splines de base couvrent l'intervalle entier. Pour cela, on répète le premier et dernier nœud $d + 1$ fois dans le vecteur de nœuds.

Les courbes B-splines qui correspondent à ces propriétés du vecteur de nœuds sont appelées courbes B-splines ouvertes. On les construit avec un vecteur de nœuds

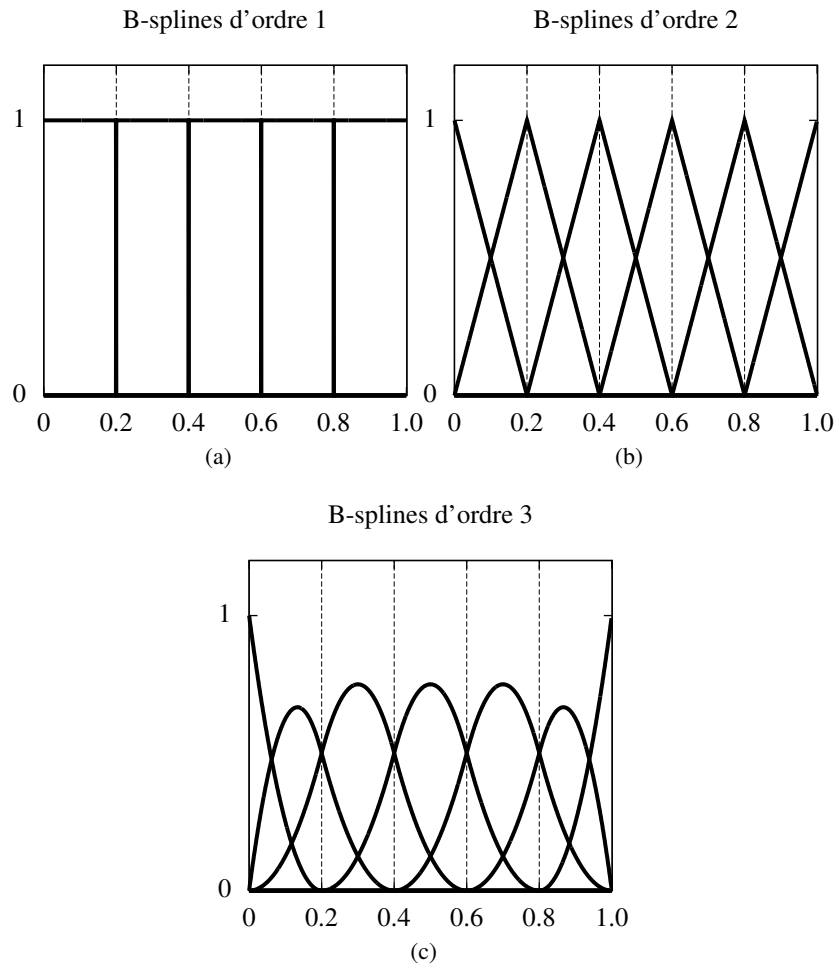


Figure 8 – Exemples de formes de fonctions B-splines de base $B_{i,k}(z)$ de degrés $k = 0$ (a), 1 (b), 2 (c) en utilisant le vecteur de nœuds $T = [0, 0.2, 0.4, 0.6, 0.8, 1]$ avec $z \in [0, 1[$.

appelé vecteur de nœuds uniforme non-périodique. On utilise ce type de construction pour notre application à la MIA. Premièrement, nous définissons ce type de vecteur de nœuds.

Soit $B_{i,d}(z)$ une fonction B-spline de degré d (ordre $k = d + 1$) avec $i = 0, \ldots, n$ et $z \in [0, n - k + 2]$. On définit le vecteur de nœuds $t_0, \ldots, t_{n+k}$ tel que :

$$t_i = \begin{cases} 0 & \text{si} \quad 0 \leq \ i < k \\ i - k + 1 & \text{si} \quad k \leq \ i \leq n \\ n - k + 2 & \text{si} \quad n < \ i \leq n + k. \end{cases}$$

Par exemple, le vecteur de nœuds uniforme non-périodique pour $n = 5$ et $k = 3$ est $[0, 0, 0, 1, 2, 3, 4, 4, 4]$. En général, ce type de vecteur à la structure suivante :

$$\underbrace{0, \ldots, 0}_{k \text{ noeuds}}, 1, 2, \ldots, n - k - 1, \underbrace{n - k + 2, \ldots, n - k + 2}_{k \text{ nœuds}}.$$

L'Algorithme 2 permet d'estimer l'IM à partir de l'estimation B-spline entre deux variables X et Y (Daub *et al.*, 2004).

7.4. *Apport des B-splines dans le contexte des attaques par canaux cachés*

Il existe une similitude importante entre l'estimation de densité par B-splines et la méthode par histogrammes puisque les B-splines d'ordre 1 sont en fait des fonctions escalier (Fig. 8a). En effet, au lieu d'affecter une donnée à une seule partition, i.e. un intervalle, on peut la placer dans un intervalle plus grand tout en lui affectant un poids grâce aux fonctions B-splines. Plus le degré d de la spline est élevé, plus l'intervalle considéré est grand. Cette particularité est notamment intéressante dans le contexte des attaques par canaux cachés. Chaque point d'une courbe de consommation est généralement composé d'une partie de bruit gaussien provenant de la méthode de mesure notamment. Ce bruit peut faire passer un point d'une partition à une partition voisine alors erronée. L'estimation par B-splines affecte un poids à chaque donnée afin que celle-ci soit placée dans plusieurs intervalles voisins prenant ainsi en compte le possible bruit.

De plus, chaque point a un poids affecté par une courbe sur l'intervalle contrairement à la méthode par histogramme qui affecte un poids par une simple fonction escalier. Grâce à cette propriété, l'estimation par B-splines semble similaire à l'estimation par noyau tout en étant plus simple et donc demandant moins de puissance de calcul. Cela est démontré expérimentalement dans la section 8. L'estimation de densités par B-splines est donc un bon compromis entre la méthode classique avec histogrammes rapide à calculer et l'estimation par noyau trop complexe.

Nous détaillons dans ce paragraphe un exemple de paramétrage des B-splines pour une attaque sur l'algorithme DES. Soit F une fonction de sélection dépendant d'une hypothèse de clé et d'une valeur intermédiaire de l'algorithme attaqué. Par exemple, nous supposons que l'attaquant souhaite attaquer les trois bits les plus significatifs de $\mathrm{SBox}(x \oplus k)$ durant le premier tour d'un DES. Il semble naturel d'utiliser $B = 8$ partitions dans une méthode d'estimation par histogrammes afin de ranger les valeurs

Algorithme 2: Estimation de l'information mutuelle en utilisant les B-splines

Données : Variables aléatoires $X = \{x_1, \ldots, x_N\}$ et $Y = \{y_1, \ldots, y_N\}$, k l'ordre de la spline, n_X le nombre de fonctions B-spline de base pour X et n_Y pour Y.

Résultat : $I(X;Y)$

1 Estimer l'entropie de X.
Déterminer les n_X Coefficients B-spline (CB) pour chaque $x_u, u = 1, \ldots, N$ tel que $B_{i,d}(x), i = 1, \ldots, n_X$. Sauvegarder la matrice $\mathit{MatrixCB}_X$ de taille $(n_X \times N)$ contenant tous les Coefficients B-spline :

$$\mathit{MatrixCB}_X[i][u] = B_{i,d}(x_u).$$

Calculer les n_X probabilités $p(a_i)$, $i = 1, \ldots, n_X$:

$$p(a_i) = \frac{1}{N} \sum_{u=1}^{N} B_{i,d}(x_u).$$

Calculer l'entropie [2] :

$$H(X) = -\sum_{i=1}^{n_X} p(a_i) \log(p(a_i)).$$

2 Répéter l'étape 1 pour la variable Y afin d'obtenir la matrice $\mathit{MatrixCB}_Y$ ainsi que l'entropie $H(Y)$.

3 Déterminer les probabilités jointes $p(a_i, b_j)$ pour toutes les $(n_X \times n_Y)$ partitions :

$$\begin{aligned} p(a_i, b_j) &= \tfrac{1}{N} \textstyle\sum_{u=1}^{N} (B_{i,k}(x_u).B_{j,k}(y_u)) \\ &= \tfrac{1}{N} \textstyle\sum_{u=1}^{N} (\mathit{MatrixCB}_X[i][u].\mathit{MatrixCB}_Y[j][u]). \end{aligned}$$

4 Calculer l'entropie jointe $H(X,Y)$ [3] :

$$H(X,Y) = -\sum_{i=1,j=1}^{n_X,n_Y} p(a_i, b_j) \log(p(a_i, b_j)).$$

5 Calculer l'information mutuelle [7] :

$$I(X;Y) = H(X) + H(Y) - H(X,Y).$$

retourner $I(X;Y)$

de sortie qui varient de 0 à 7. Dans le cas de l'estimation par B-splines, nous voulons aussi couvrir l'ensemble des valeurs visées possibles $[0, 7]$. On se rappelle que les fonctions B-splines sont définies sur $[0, n-k+2]$ et sont non-nulles aux nœuds extrêmes car nous utilisons un vecteur de nœuds uniforme non-périodique. Le paramètre k est généralement fixé à $k = 3$ pour que le calcul de fonctions B-splines ne soit pas trop complexe tout en garantissant des courbes assez lisses (Daub *et al.*, 2004). Le nombre de points d'arrêt $narret = n - k + 2$ correspond au paramètre B de l'estimation par histogrammes. Dans notre exemple, avec $k = 3$ et $narret = B = 8$, on obtient $n = narret + k - 2 = 9$ fonctions de bases. Il suffit ainsi de ne modifier que les paramètres k et $narret$, le nombre de fonctions de bases n est déduit.

7.5. *Utilisation du test de Cramér-von-Mises avec les B-splines*

Dans (Veyrat-Charvillon *et al.*, 2009), les auteurs proposent une attaque par canaux cachés basée sur le test non-paramétrique de Cramér-von-Mises (CVM). Ils montrent son efficacité face à différentes attaques différentielles. Le test CVM est similaire au test de Kolmogorov-Smirnov (KS) plus connu. Ce dernier est largement utilisé dans le domaine des statistiques non-paramétriques. Le test KS à deux échantillons évalue la différence maximale entre deux fonctions de répartition. Ce test KS à deux échantillons peut d'ailleurs être rapproché du test non-paramétrique Mann-Whitney qui est l'équivalent non-paramétrique du T-test utilisé dans la DPA.

Nous rappelons brièvement la définition d'une fonction de répartition. Soit X une variable aléatoire discrète ayant pour valeurs $\{x_1, \ldots, x_N\}$ avec probabilités $P(X = x_i)$ pour $i = 1, \ldots, N$. La fonction de répartition de X, notée $F_X(x)$, est définie telle que :

$$F_X(x) = P(X \leq x) = \sum_{x_i \leq x} P(X = x_i).$$

Nous notons les fonctions de répartition pour les variables X et Y par $F_X(x)$ et $F_Y(x)$ respectivement. Le test KS est défini par :

$$D_{\mathrm{KS}}(X \parallel Y) = \sup_i |F_X(i) - F_Y(i)| .$$

De la même manière, le test CVM est défini par :

$$D_{\mathrm{CVM}}(X \parallel Y) = \sum_i (F_X(i) - F_Y(i))^2.$$

Nous considérons par la suite le test CVM qui donne des résultats légèrement meilleurs au test KS.

En pratique, les fonctions de répartition se calculent à partir d'une structure d'histogramme. Le test de CVM ne demande donc que peu de temps de calcul par rapport

à l'estimation par histogrammes classique. La méthode des B-splines présentée précédemment peut s'appliquer de manière similaire dans ce contexte d'estimation de fonctions de répartition. Les différentes valeurs des échantillons sont affectées à plus d'une partition grâce aux fonctions B-splines. Une fois cet histogramme lissé créé, les fonctions de répartition et le test de CVM peuvent être calculés de manière classique. L'amélioration apportée par les B-splines est néanmoins moins importante que dans le cas d'estimation de densités de probabilités. En effet, l'estimation des fonctions de répartition converge plus rapidement que l'estimation d'une densité de probabilité. Même faible, cette amélioration peut toutefois être intéressante dans certains cas.

Dans la section suivante, nous démontrons avec des données réelles l'amélioration par deux qu'apportent les B-splines par rapport à une estimation avec histogrammes. De plus, cette technique s'applique particulièrement bien au domaine des attaques par canaux cachés. En effet, pouvoir permettre à un point d'une courbe de consommation de courant d'être dans plusieurs partitions voisines compense le bruit qui pourrait être lié à la mesure.

8. Résultats expérimentaux

Comme précisé précédemment, nous nous concentrons sur la comparaison de l'efficacité d'estimateurs de densités de probabilités non-paramétriques dans un contexte d'attaques par canaux cachés. Nous les comparons tout de même avec l'un des meilleurs estimateurs de probabilité paramétrique et avec l'attaque de référence utilisant le coefficient de Pearson. L'attaque CPA est d'ailleurs considérée comme référentiel pour évaluer l'efficacité des différentes propositions dans cette section. En effet, il s'agit de l'une des meilleures attaques sur composants CMOS. Il convient de noter que de récentes propositions (Souissi *et al.*, 2010) d'attaques permettent d'obtenir des résultats similaires, voire supérieurs à la CPA. Nous avons mené des attaques sur deux implémentations de deux algorithmes : un DES et un algorithme de multiplication multi-précision. Nous comparons :

– l'attaque par analyse d'information mutuelle utilisant des histogrammes présentée Section 5.1 (notée HE pour *Histogram Estimator*),

– l'attaque utilisant la technique d'estimation par noyau avec le noyau gaussien présentée Section 5.2 (notée KDE pour *Kernel Density Estimator*),

– l'attaque utilisant la GMI présentée Section 6 (notée GMIA),

– l'attaque utilisant l'estimation par B-splines présentée Section 7.4 (notée BSE pour *B-Splines Estimator*),

– le test de Cramér-von-Mises présentée Section 7.5 (notée CVM),

– le même test utilisant le lissage par B-splines présentée Section 7.5 (notée CVMB),

– l'attaque utilisant l'estimateur paramétrique à base de cumulants (Lee *et al.*, 2010) (notée CE pour *Cumulant Estimator*),

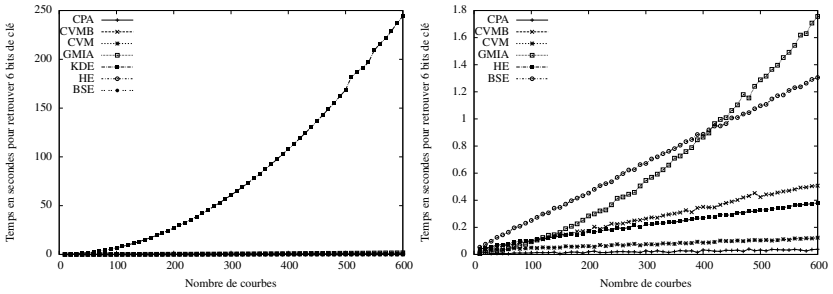


Figure 9 – Comparaison du temps de calcul moyen nécessaire pour attaquer 6 bits de clés dans le cas du DES. La figure de droite est un agrandissement de celle de gauche en enlevant l'attaque KDE.

– l'attaque CPA présentée Section 2.2.2.

Tout d'abord, nous introduisons quelques métriques importantes permettant de mesurer l'efficacité d'attaques par canaux cachés.

8.1. *Évaluation de l'efficacité d'attaques par canaux cachés*

Nous utilisons deux métriques bien connues de la littérature et proposée dans (Standaert *et al.*, 2008) qui permettent d'identifier l'efficacité d'une attaque par canaux cachés :

– la *guessed entropy*, qui est la position, en moyenne, de l'hypothèse correcte dans le vecteur d'hypothèses triées à la sortie d'une attaque,

– le *first-order success rate* qui est, pour un nombre de traces données, la probabilité que l'hypothèse correcte soit en première position dans le vecteur d'hypothèses triées.

Nous analysons d'abord brièvement la complexité de calcul nécessaire pour chacune des attaques dans la Figure 9. Les mesures de temps ont été enregistrées sur un ordinateur de bureau classique équipé d'un processeur Pentium 4. On remarque bien que, même si l'attaque BSE demande plus de calculs que l'attaque HE classique, elle est nettement plus rapide qu'une analyse KDE.

8.2. *Comparaison sur les traces du DPA Contest 2008/2009*

Notre premier jeu d'attaque (Fig. 10) est réalisé en utilisant les traces du DPA Contest 2008/2009 (VLSI research group and TELECOM ParisTech, 2008). La valeur intermédiaire attaquée est la sortie de la SBox au dernier tour du DES. Pour chaque

attaques utilisant un estimateur non-paramétrique, nous ne considérons aucun modèle de consommation comme suggéré dans (Gierlichs *et al.*, 2008). Le modèle en poids de Hamming (Messerges *et al.*, 1999a) est utilisé pour les attaques paramétriques. Chaque attaque est réalisée sur 135 ensembles de 600 courbes de consommation afin de moyenner les résultats. On constate l'amélioration par un facteur de deux entre l'attaque HE et BSE. L'intérêt des B-splines par rapport aux histogrammes se confirme en pratique. Les attaques CVM et CVMB sont proches et donnent de meilleurs résultats. Néanmoins, leurs résultats sont assez loin derrière les très efficaces CPA et CE.

8.3. *Comparaison sur des traces d'une multiplication multi-précision sur un Atmel STK 600 utilisant un AVR ATmega2561*

Nous testons ensuite l'efficacité des attaques sur une plateforme différente et en utilisant un autre algorithme. Nous implémentons sur une carte Atmel STK 600 utilisant un processeur 8-bit AVR ATmega2561 (ATMEL, n.d.) un algorithme de multiplication multi-précision. L'algorithme utilise la méthode bien connue de Comba (Comba, 1990). Le but de l'attaquant est de retrouver les octets d'un facteur secret fixé alors que de nombreux facteurs aléatoires, connus de l'attaquant, sont donnés à l'algorithme. Le contexte d'acquisition de ces traces est bien différent de celui du DPA Contest. En effet, la carte Atmel STK 600 n'est pas adaptée pour des mesures de courant. Ainsi les traces de consommation de courant contiennent beaucoup plus de bruit que celles du DPA Contest. Comme précédemment, nous ne considérons aucun modèle de consommation particulier pour les attaques utilisant un estimateur non-paramétrique alors que les autres attaques utilisent le modèle en poids de Hamming. Chaque attaque est réalisée sur 20 ensembles de 2000 courbes de consommation. Les résultats dans ce contexte sont particulièrement intéressants (Fig. 11). Tout d'abord, nous confirmons le facteur deux d'amélioration entre l'estimation d'information mutuelle avec histogrammes et avec B-splines. De plus, les résultats de BSE sont maintenant relativement proches de ceux de CVM et CVMB. L'observation la plus intéressante est la réduction de l'écart entre la CPA et l'attaque utilisant la méthode de B-splines.

9. Conclusion

Nous présentons dans cet article différentes techniques d'estimation non-paramétrique de densités de probabilités utilisant notamment les B-splines dans le contexte des attaques par canaux cachés. L'estimation par B-splines s'applique particulièrement bien à ce domaine puisqu'elle permet de tenir compte du possible bruit de mesure associée à une acquisition de courbe de consommation. Nous montrons cela grâce à un comparatif de différentes attaques par canaux cachés sur une plateforme avec du bruit où l'estimation par B-splines donne de bons résultats. Les tests statistiques paramétriques classiques sont néanmoins toujours parmi les plus intéressants. La méthode d'estimation paramétrique à base de cumulants proposée dans (Lee *et*

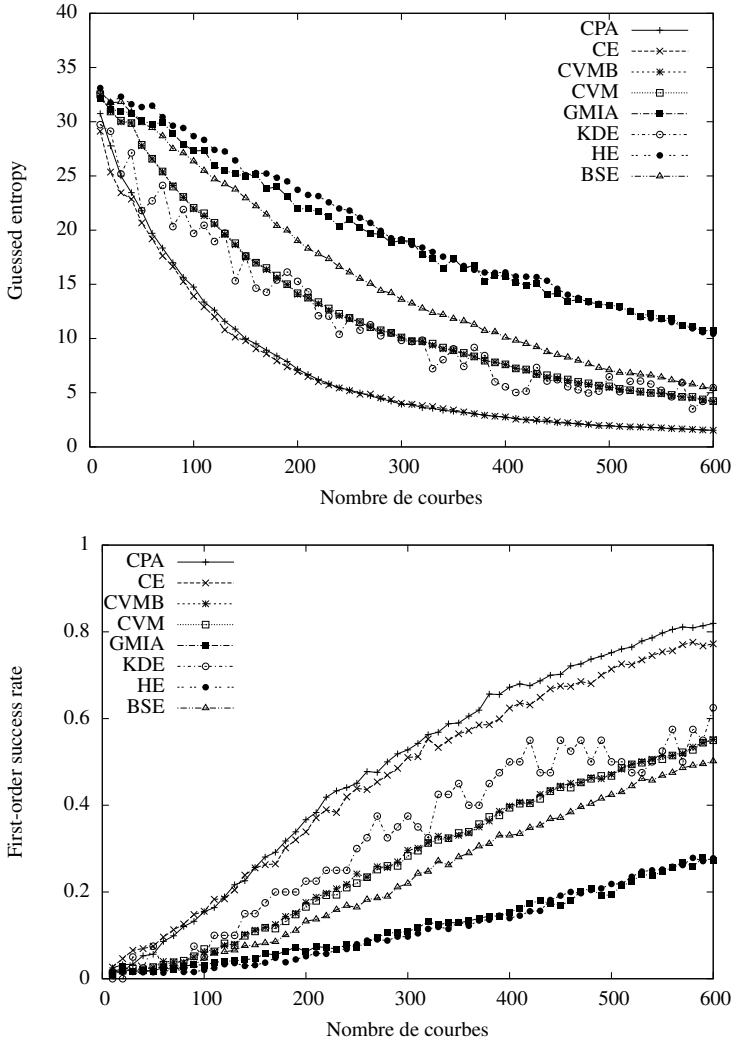


Figure 10 – Comparatif de résultats d'attaques sur des traces du DPA Contest 2008/2009 implémentant un DES. L'axe des abscisses correspond au nombre de courbes de consommation utilisées. L'axe des ordonnées correspond au rang pour la métrique *guessed entropy* ou à une probabilité pour la métrique *first-order success rate*.

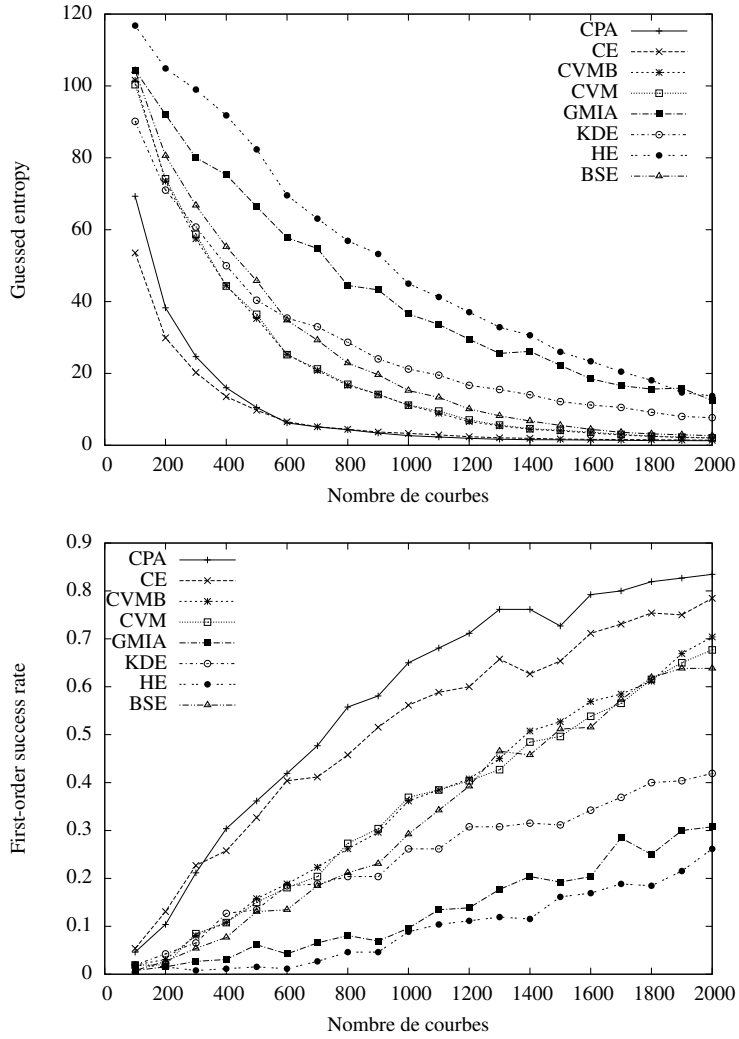


Figure 11 – Comparatif de résultats d'attaques sur des traces enregistrées sur une carte Atmel STK600 utilisant un AVR ATmega2561 implémentant une multiplication multi-précision. L'axe des abscisses correspond au nombre de courbes de consommation utilisées. L'axe des ordonnées correspond au rang pour la métrique *guessed entropy* ou à une probabilité pour la métrique *first-order success rate*.

al., 2010) donne de très bons résultats. La distinction entre estimateurs paramétriques et non-paramétriques devrait être plus clairement énoncée lors de la comparaison d'attaques par canaux cachés. Il faut distinguer ces deux classes d'attaques suivant le niveau de connaissances de l'adversaire sur le composant qu'il attaque. Notamment sa connaissance du type de logique utilisée par le composant ou du modèle de consommation de courant correspondant peuvent être considérés comme des informations non triviales dans certains scénarios d'attaques. L'attaque MIA utilisant des estimateurs non-paramétriques n'est toujours pas la plus puissante. Néanmoins, nous améliorons grandement ses performances comparé à l'utilisation d'histogrammes pour l'estimation qui est utilisée dans de nombreux articles comme référence. Nous rappelons que toutes ces expériences sont effectuées sur des composants CMOS. Les attaques paramétriques qui utilisent le modèle du poids de Hamming sont donc naturellement avantagées. Même si l'étude a été effectuée sur des composants utilisant la logique CMOS, on peut s'attendre à une amélioration de performances similaire sur différents types de logiques, qui peuvent être vues comme des protections aux attaques par canaux cachés, avec l'utilisation d'estimateurs non-paramétriques efficaces.

10. Bibliographie

ATMEL, « ATmega 2561 Data Sheet », , `http://www.atmel.com/dyn/resources/prod_documents/doc2549.pdf`, n.d.

Batina L., Gierlichs B., Lemke-Rust K., « Comparative Evaluation of Rank Correlation Based DPA on an AES Prototype Chip », *ISC 2008, LNCS*, vol. 5222, p. 341-354, 2008.

Brier E., Clavier C., Olivier F., « Correlation Power Analysis with a Leakage Model », *CHES 2004, LNCS*, vol. 3156, p. 135-152, 2004.

Comba P., « Exponentiation Cryptosystems on the IBM PC », *IBM Syst. J.*, vol. 29, p. 526-538, 1990.

Daub C., Steuer R., Selbig J., Kloska S., « Estimating Mutual Information Using B-spline Functions - an Improved Similarity Measure for Analysing Gene Expression Data », *BMC Bioinformatics*, vol. 5, p. 118, 2004.

Deboor C., *A Practical Guide to Splines*, Springer-Verlag Berlin and Heidelberg GmbH & Co. K, December, 1978.

Gierlichs B., Batina L., Tuyls P., Preneel B., « Mutual Information Analysis - A Generic Side-Channel Distinguisher », *CHES 2008, LNCS*, vol. 5154, p. 426-442, 2008.

Kocher P., Jaffe J., Jun B., « Differential Power Analysis », *CRYPTO 1999, LNCS*, vol. 1666, p. 388-397, 1999.

Lee T.-H., Berthier M., « Mutual Information Analysis under the View of Higher-Order Statistics », *Advances in Information and Computer Security, LNCS*, vol. 6434, p. 285-300, 2010.

Maghrebi H. andGuilley S., Danger J.-L., Flament F., « Entropy-based Power Attack », *Hardware-Oriented Security and Trust (HOST)*, p. 1-6, 2010.

Mangard S., « A Simple Power-Analysis (SPA) Attack on Implementations of the AES Key Expansion », *ICISC 2002, LNCS*, vol. 2587, p. 343-358, 2002.

Messerges T. S., Dabbish E. A., Sloan R. H., « Investigations of Power Analysis Attacks on Smartcards », *USENIX Workshop on Smartcard Technology*, p. 151-162, 1999a.

Messerges T. S., Dabbish E. A., Sloan R. H., « Power Analysis Attacks of Modular Exponentiation in Smartcard », *CHES 1999, LNCS*, vol. 1717, p. 144-157, 1999b.

Moon Y.-I., Rajagopalan B., Lall U., « Estimation of Mutual Information using Kernel Density Estimators », *Physical Review E*, vol. 52, n° 3, p. 2318-2321, 1995.

Moradi A., Mousavi N., Paar C., Salmasizadeh M., « A Comparative Study of Mutual Information Analysis under a Gaussian Assumption », *Information Security Applications, LNCS*, vol. 5932, p. 193-205, 2009.

Oswald E., « Enhancing Simple Power-Analysis Attacks on Elliptic Curve Cryptosystems », *CHES 2002, LNCS*, vol. 2523, p. 82-97, 2002.

Parzen E., « On the Estimation of a Probability Density Function and Mode », *Annals of Mathematical Statistics*, vol. 33, p. 1065-1076, 1962.

Pompe B., Blidh P., Hoyer D., Eiselt M., « Using Mutual Information to Measure Coupling in the Cardio Respiratory System », *IEEE Engineering in Medicine and Biology Magazine*, vol. 17, n° 6, p. 32-39, 1998.

Pompe B., Heilfort M., « On the Concept of the Generalized Mutual Information Function and Efficient Algorithms for Calculing it », 1995.

Pompe B., Physik F., « Measuring Statistical Dependences in a Time Series », *Journal of Statistical Physics*, vol. 73, p. 587-610, 1993.

Prouff E., Rivain M., « Theorical and Practical Aspects of Mutual Information Based Side Channel Analysis », *ACNS 2009, LNCS*, vol. 5536, p. 499-518, 2009.

Souissi Y., Nassar M., Guilley S., Danger J.-L., Florent F., « First Principal Components Analysis : A New Side Channel Distinguisher », *International Conference on Information Security and Cryptology (ICISC)*, 2010.

Standaert F., Koeune F., Schindler W., « How to Compare Profiled Side-Channel Attacks ? », *Applied Cryptography and Network Security, LNCS*, vol. 5536, p. 485-498, 2009.

Standaert F.-X., Gierlichs B., Verbauwhede I., « Partition vs . Comparison Side-Channel Distinguishers : An Empirical Evaluation of Statistical Tests for Univariate Side-Channel Attacks against Two Unprotected CMOS Devices », *ICISC 2008, LNCS*, vol. 5461, p. 253-267, 2008.

Tiri K., Hwang D., Hodjat A., Lai B.-C., Yang S., Schaumont P., Verbauwhede I., « Prototype IC with WDDL and Differential Routing - DPA Resistance Assessment », *CHES 2005, LNCS*, vol. 3659, p. 354-365, 2005.

Venelli A., « Efficient Entropy Estimation for Mutual Information Analysis using B-splines », *WISTP 2010, LNCS*, vol. 6033, p. 17-30, 2010.

Veyrat-Charvillon N., Standaert F., « Mutual Information Analysis : How, When and Why ? », *CHES 2009, LNCS*, vol. 5747, p. 429-443, 2009.

VLSI research group and TELECOM ParisTech, « The DPA Contest 2008/2009 », , `http://www.dpacontest.org`, 2008.

Alexandre Venelli *est docteur en informatique et ingénieur cryptographe au sein de la société Inside Secure à Rousset. Les travaux publiés au sein de cet article ont été présentés dans le cadre de son manuscript de thèse.*