Semaine d'Etude Mathématiques et Entreprises 2 : Implémentation en boîte blanche et recherche de points d'intérêt

Davide Alessio, Guenaëlle de Julis, Benoît Gaudel, Daria Stepanova

Semaine d'Etude Maths–Entreprises 2

28 novembre – 2 décembre 2011, Université Lyon I

Implémentation en boîte blanche et recherche de points d'intérêt

D. Alessio[a] G. De Julis[b]
B. Gaudel[c] D. Stepanova[d]

[a] *IRMAR, Université de Rennes 1, France*
[b] *Institut Fourier, Université de Grenoble, France*
[c] *Université de Versailles Saint-Quentin-en-Yvelines, France*
[d] *I3M, Université de Montpellier 2, France*

Sujet proposé par :

Oberthur Technologies

Correspondants : E. Dottax et E. Prouff (Oberthur)

Maths & Entreprises — cnrs dépasser les frontières — a m i e s

Résumé

Les deux sujets proposés par Oberthur concernent la cryptographie. Le premier propose d'étudier des méthodes de recherche de points d'intérêt dans des courbes de consommation liées à un calcul cryptographique. Ces points sont d'autant plus intéressants si les attaques qui les exploitent sont efficaces. Le deuxième sujet traite de l'implémentation d'algorithmes cryptographiques (algorithme de chiffrement par exemple) de telle sorte que leurs secrets (clés de chiffrement,...) soient dissimulés dans le code.

Numéro de publication : SEME002-2011-11-A

1 Introduction

Les deux sujets proposé par Oberthur concernent l'implémentation de logiciels et plus particulièrement l'implémentation d'algorithmes cryptographiques.

Le but du premier sujet est d'identifier de nouvelles méthodes de recherche de points d'intérêts dans des courbes de consommation liées à un calcul cryptographique, l'intérêt des points étant jugé en fonction de l'efficacité des attaques qui les exploitent. Ce sujet est abordé dans la section 2.

Le but du second sujet est de rechercher des mécanismes d'implantation d'algorithmes cryptographiques (tels que des algorithmes de chiffrement par exemple) qui permettent de cacher les secrets de l'algorithme (comme sa structure et/ou ses clefs de chiffrement) dans le code, de sorte que leur extraction soit difficile, même pour une personne qui aurait un accès total à ce code offusqué. Idéalement, il s'agirait aussi de définir des méthodes associées permettant de mesurer la résistance du mécanisme d'offuscation contre ces attaques. Ce sujet est abordé dans la section 3.

2 Recherche de points d'intérêt

Ici une *observation* est une courbe de consommation. On se donne deux *groupes* d'observations. On cherche à *différencier* ces deux groupes. La mise en évidence d'une telle différenciation *valide* en effet l'hypothèse cryptographique qui a présidé à la constitution de ces deux groupes.

On peut aussi se trouver avec un groupe d'observations fournies par une carte dont la clé n'est pas connue. Et un autre groupe fourni par un simulateur qui tient compte d'une hypothèse sur la clé.

On note C_0 la carte observée (celle dont on cherche la clé k_0) et C_k la carte simulée pour la valeur k de la clé.

Dans la section 2.1 nous présentons une méthode naive pour évaluer la distance entre deux signaux. Une méthode statistique plus avancée est présentée dans les sections 2.2 et 2.3. Il s'agit de l'analyse canonique. Une étude de l'intérêt de méthodes discrètes de classification est esquissée dans la section 2.4. La section 2.5 étudie une analogie possible avec certaines méthodes de diagnostic clinique. Le temps a manqué pour évaluer finement la pertinence de toutes ces méthodes dans le contexte étudié.

2.1 Élimination par seuil

Cette démarche basique suppose que l'on dispose de k courbes de consommation synchronisées ; une par hypothèse ce clé. On mesure la distance entre ces courbes et la courbe observée sur C_0. On élimine toutes les courbes qui s'écartent trop à un moment donné. La *distance* est donc la distance L^∞. Par la suite, on note :

- $T_k(t)$ la trace, en fonction du temps, simulée en prenant l'hypothèse de clé k.
- $T_0(t)$ la trace, en fonction du temps, observée de la carte à puce.

Description de l'algorithme :

Input :

ϵ la précision verticale,

n la subdivision du temps (le nombre de points d'observation).

Processing :

- Subdiviser la fenêtre de temps : $t_1, \cdots, t_n$,
- Boucle sur k :
  Parcours des t_i : si $|T_k - T_0|_{t=t_i} > \epsilon$, jeter T_k.

Output : Un ensemble de traces candidates pour la précision (ϵ, n).

2.2 Analyse canonique

L'analyse canonique permet de relier deux ensembles de variables en déterminant dans quelle mesure elles expriment les mêmes propriétés. Pour le problème posé, on pourra considérer qu'un individu est un message à chiffrer. On suppose que les points de mesure (points d'intérêt) sont déjà trouvés, ou déjà connus.

Soit t le temps correspondant à un point d'intérêt. On note X_t la variable associée. La valeur de la variable X_t en le message m est l'intensité $T_0(t)$ mesurée au temps t lorsque la carte observée C_0 chiffre le message m. Un premier groupe de variables est ainsi obtenu. Chaque point d'observation sur la courbe de consommation de la carte C_0 donne ainsi une variable. Au besoin, nous remplacerons ces variables par les *variables centrées* associées (on retranche la valeur moyenne).

On fixe maintenant une valeur hypothétique k de la clé. Soit encore t le temps correspondant à un point d'intérêt. On note Y_t la variable associée. La valeur de la variable Y_t en le message m est l'intensité $T_k(t)$ mesurée au temps t lorsque la carte C_k, simulée avec la clé k, chiffre le message m. Un second groupe de variables est ainsi obtenu. Chaque point d'observation sur la courbe de consommation de la carte C_k donne ainsi une variable. Au besoin, nous remplacerons ces variables par les variables centrées associées (on retranche la valeur moyenne).

On fixe une clé hypothétique k. On a n individus (n messages à chiffrer). On a p points d'intérêt t_1, t_2, …, t_p. À chaque point d'intérêt t_i pour $1 \leq i \leq p$ correspondent deux variables. Une variable X_{t_i} telle que $X_{t_i}(m)$ est lue en t_i sur la courbe de consommation de C_0 quand elle chiffre m. Une variable Y_{t_i} telle que $Y_{t_i}(m)$ est lue en t_i sur la courbe de consommation de C_k quand elle chiffre m.

Il y a n messages. On obtient deux ensembles de mesures. D'une part les $n \times p$ valeurs des p variables X_{t_i} en les n messages. Ce sont les mesures de consommation sur la carte C_0. D'autre part les $n \times p$ valeurs des p variables Y_{t_i} en les n messages. Ce sont les mesures de consommation sur la carte simulée C_k.

On obtient les deux matrices de mesures suivantes.

$$X = \begin{pmatrix} x_{11} & \cdots & x_{1p} \\ \vdots & \cdots & \vdots \\ \vdots & \cdots & \vdots \\ \vdots & \cdots & \vdots \\ x_{n1} & \cdots & x_{np} \end{pmatrix}$$

$$Y_k = \begin{pmatrix} y_{11} & \cdots & y_{1p} \\ \vdots & \cdots & \vdots \\ \vdots & \cdots & \vdots \\ \vdots & \cdots & \vdots \\ y_{n1} & \cdots & y_{np} \end{pmatrix}$$

On considère les deux sous espaces de $\mathbb{R}^n$ engendrés par les vecteurs colonnes de X et de Y_k.

- Si ces deux espaces sont confondus, alors les deux ensembles de variables mesurent les mêmes propriétés.
- S'ils sont orthogonaux, il n'y a aucun lien entre les deux ensembles.

On appelle *variables canoniques* les vecteurs normés de ces deux espaces, et si $u = X \cdot a$, alors a est appelé *facteur canonique* associé à la variable canonique u. Donc une variable est une combinaison des variables initiales, et le facteur canonique est le vecteur des coefficients qui apparaissent dans cette combinaison.

Pour voir à quel point les deux espaces de variables sont proches, on recherche un premier couple (u_1, v_1) de variables canoniques, tel que

- u_1 soit dans l'espace engendré par les colonnes de X,
- v_1 soit dans l'espace engendré par les colonnes de Y_k,
- et leur *corrélation*, c'est-à-dire l'angle qu'ils forment, soit minimale.

Ensuite, on recherche un couple (u_2, v_2) tel que

- u_2 et u_1 soient orthogonaux,
- v_2 et v_1 soient orthogonaux,
- la corrélation entre u_2 et v_2 soit minimale.

Et ainsi de suite. Si l'on note :

- $V_{XX} = X^t \cdot X$ et $V_{YY} = Y_k^t \cdot Y_k$ les matrices de *variance* de X et Y_k,
- $V_{YX} = Y_k^t \cdot X$ et $V_{XY} = X^t \cdot Y_k$ les matrices de *covariance*,

alors les *facteurs canoniques* sont les solutions des équations

$$V_{XX}^{-1} V_{XY} V_{YY}^{-1} V_{YX} a_i = \lambda_i a_i,$$

$$V_{YY}^{-1} V_{YX} V_{XX}^{-1} V_{XY} b_i = \lambda_i b_i.$$

On se ramène donc à un calcul de valeurs propres et de vecteurs propres. Comme de plus les λ_i sont les carrés des $cos(u_i, v_i)$, ils sont d'autant plus grands que les espaces engendrés par les colonnes de X et Y_k sont proches.
Ainsi, en trouvant les valeurs propres des produits de matrices présents dans les équations précédentes, on peut, en faisant leur somme, mesurer à quel point les clés testées sont susceptibles d'être la clé cherchée. Plus la corrélation est grande, plus l'hypothèse de clé est vraisemblable. Les meilleurs points d'intérêts correspondent aux facteurs canoniques dotés des plus grandes valeurs propres λ. Ce sont les premiers qui apparaissent dans l'analyse. En outre, les facteurs canoniques mesurent l'importance relative de chaque point d'intérêt.

2.3 Analyse canonique, autre piste

L'analyse canonique ne requiert pas que le nombre de variables soit le même dans chacun des deux groupes. On distingue seulement un groupe de p variables et un groupe de q variables.
Nous avons donc :

- n individus,
- p variables appelées *caractères explicatifs*,
- q variables appelées *caractères à expliquer*.

Soit donc

- $X = (X_1, \ldots, X_p)$ les variables explicatives centrées
- $Y = (Y_1, \ldots, Y_q)$ les variables à expliquer centrées

Le procédé est alors le suivant :

Etape 1 : Déterminer (U_1, V_1), combinaison linéaire de X et Y de sorte que :

- $Var(U_1) = Var(V_1) = 1$ (pour assurer l'unicité de la solution)
- $Cor(U_1, V_1)$ soit maximale

On obtient donc $a = (a_1, \cdots, a_p)$ et $b = (b_1, \cdots, b_q)$ uniques, où

$$U_1 = a_1 X_1 + \cdots + a_p X_p$$

$$V_1 = b_1 X_1 + \cdots + b_q X_q$$

$$r_1 = Cor(U_1, V_1)$$

Quelques explications :

- U_1 et V_1 sont les premières variables canoniques
- a et b sont les premiers facteurs canoniques
- r_1 est la première corrélation canonique

A ce stade, la plus grande partie des variables à expliquer ont été mises en relation avec les variables explicatives, mais il reste sûrement une partie non expliquée. On réitère donc le calcul avec les variables qui ne sont pas encore expliquées.

Etape 2 : Déterminer (U_2, V_2), non corrélés à (U_1, V_1), combinaisons linéaires de X et Y de sorte que :

- $Var(U_2) = Var(V_2) = 1$ (pour assurer l'unicité de la solution)
- $r_2 = Cor(U_2, V_2)$ soit maximale

... et ainsi de suite, en prenant garde à ce que (U_k, V_k) ne soit pas corrélé à (U_l, V_l) pour $l < k$

Etape s : jusqu'à obtenir r_s, où s est le rang de la matrice V_{XY}.
On possède alors s corrélations canoniques, avec $r_i > r_{i+1}$.

Plusieurs approches semblent pertinentes dans le cadre du problème des points d'intérêts. Une première approche a été exposée au paragraphe précédent. Nous en esquissons ici une autre. Son objectif est de retrouver k_0, ce qui équivaut à trouver k tel que $T_k = T_0$. Pour se faire, les paramètres seront les suivants :

- les individus sont n messages (le chiffrement est déterministe)

– les variables à expliquer sont les mesures effectuées lors du chiffrement par C_0. Il y a autant de variables que d'instants t d'observation dans cette phase expérimentale. On note q ce nombre d'instants.
– les variables explicatives sont les mesures effectuées sur la carte à puce C_k, en chiffrant les messages avec une hypothèse de clé fixée k. Il y a p telles variables correspondant à p instants. Ces instants ne sont pas forcément les mêmes que ceux choisis pour observer C_0. En particulier p n'est pas forcément égal à q.

Nous obtenons ainsi, pour un morceau de clé suggéré k, les matrices $X(k)$ et Y suivantes :

$$X_k = \begin{pmatrix} x(k)_{1,1} & \cdots & x(k)_{1,p} \\ \vdots & \cdots & \vdots \\ \vdots & \cdots & \vdots \\ \vdots & \cdots & \vdots \\ x(k)_{n,1} & \cdots & x(k)_{n,p} \end{pmatrix} \qquad Y = \begin{pmatrix} y_{1,1} & \cdots & y_{1,q} \\ \vdots & \cdots & \vdots \\ \vdots & \cdots & \vdots \\ \vdots & \cdots & \vdots \\ y_{n,1} & \cdots & y_{n,q} \end{pmatrix}$$

Remarque : Il ne faut pas oublier de centrer X_k et Y.

On applique alors le procédé décrit précédemment pour obtenir $(r_1, \cdots, r_s)$. Ainsi, on peut associer à l'hypothèse de clé k la quantité

$$r(k) = \sum_{i=1}^{s} r_i^2.$$

On valide l'hypothèse qui obtient la plus grande corrélation.

Lorsque p est différent de q, on observe que la complexité du calcul des valeurs propres dépend surtout du minimum de p et q car ces valeurs propres sont communes à deux matrices de tailles $p \times p$ et $q \times q$. Notons aussi que l'une de ces deux matrices *ne dépend pas de l'hypothèse de clé*. C'est la matrice d'observation Y. On peut donc se permettre un plus grand nombre d'observation q sur C_0.

Conclusion Un usage raisonné de l'analyse canonique doit permettre de détecter les meilleurs points d'intérêt et la meilleure hypothèse de clé. Plusieurs stratégies sont possibles, à choisir en fonction de leur efficacité mais aussi de leur coût algorithmique (multiplication des matrices, inversion des matrices, recherche des valeurs et vecteurs propres). Il est possible de procéder en plusieurs étapes. Par exemple, dans un premier temps, utiliser l'analyse canonique pour obtenir les trois meilleurs marqueurs, en comparant deux familles d'exécutions correspondant à deux clés choisies donc connues. Puis, dans un second temps, d'utiliser ces trois marqueurs pour finaliser l'attaque et déterminer la clé. Toutefois, cela suppose d'avoir préalablement réussi à effacer les contre-mesures qui viseraient à désynchroniser les courbes. Un travail préalable sur les courbes est peut-être nécessaire. Un exemple d'un tel traitement est donné dans la section 2.5.

2.4 Autres méthodes de classification

L'analyse canonique semble bien adaptée à l'étude de variables continues. Dans notre situation, il est possible que certaines caractéristiques des courbes de consommation correspondent plutôt à des variables discrètes. Nous esquissons dans cette section une méthode

discrète de classification : l'étude des configurations d'ordres. On trouve des références sur cette méthode dans [5]. Des algorithmes philogénétiques comme ceux décrits dans [2] seraient sans doute utiles eux aussi.

Étant donnée une collection de points sur une courbe de consommation, on s'intéresse à leurs positions relatives : le premier point est il à gauche ou à droite du premier, est il plus haut ou plus bas, etc. On se restreint délibérément à ces informations qualitatives. On souhaite sélectionner parmi toutes ces informations, celles qui sont pertinentes, c'est-à-dire celles qui permettent d'organiser la population en deux groupes, ou bien celles qui permettent d'évaluer la distance entre deux groupes donnés d'observations.

On cherche un critère algébrique pour dire si une configuration d'ordres est fortuite ou pertinente. On verra que l'annulation du déterminant fournit un tel critère. Le déterminant s'annule-t-il si l'on déplace les points tout en respectant l'ordre ? Dans le cas de l'analyse des courbes de consommation, on se place dans l'ensemble $E = \mathbb{R}^2$ de dimension $n-1 = 3-1$ et l'on prend on ensemble $\mathcal{P}$ de $n = 3$ points sur chaque courbe. On note $x_{j,i}$ la i-ème coordonnée du point j, et l'on forme la matrice

$$M_{\mathcal{P}} = \begin{pmatrix} 1 & 1 & 1 \\ x_{1,1} & x_{2,1} & x_{3,1} \\ x_{1,2} & x_{2,2} & x_{3,2} \end{pmatrix}$$

Dans notre cas, il n'y a que deux coordonnées (le temps et la hauteur) notées x et y. On obtient donc une matrice

$$M_{\mathcal{P}} = \begin{pmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{pmatrix}$$

Un premier invariant discret est alors le signe du déterminant de cette matrice, appelé *orientation du simplexe*. On peut se demander si ce signe est un invariant pertinent.

Un deuxième invariant est fourni par les deux ordres induits sur les points par les deux coordonnées. On peut en effet classer les trois points selon x ou selon y.

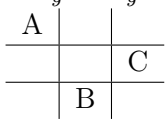
Définition 1 *On dira que l'ensemble $\mathcal{P}$* **respecte** *la configuration $\mathcal{C} = (<_i)_{1\leq i\leq n-1}$ des $n-1$ ordres $<_i$, si pour tout i de 1 à $n-1$ et tous j, k, on a $j <_i k$ implique $x_{j,i} < x_{k,i}$. Autrement dit, les points se classent selon l'ordre $<_i$ pour la i-ème coordonnée.*

Définition 2 *On dit qu'une configuration $\mathcal{C}$ est* **fixe** *si pour tout ensemble de points $\mathcal{P}$ respectant $\mathcal{C}$, l'orientation du simplexe formé par $\mathcal{P}$ est la même. Autrement dit, le déterminant ne change pas de signe lorsque l'on déplace les points, pourvu qu'on respecte tous les ordres sur les coordonnées.*

On cherche à determiner la fixité des configurations ce qui revient à savoir si le déterminant de $M_{\mathcal{P}}$ peut s'annuler. Si trois points sont toujours dans une configuration fixe, quelle que soit la courbe observée, on considère que leur orientation fournit une information intéressante, et permet de classer les observations en deux groupes bien définis.

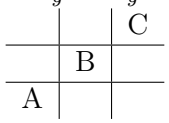
En dimension 2 il n'existe que deux configurations d'ordres totaux distinctes, à permutation près. L'une des deux est fixe.

$A <_x B <_x C$
$A <_y C <_y B$



configuration fixe

$A < xB < xC$
$A <_y B <_y C$



configuration non-fixe

Pour savoir si une configuration est fixe, on cherche à écrite le déterminant de sorte que tous les termes qui apparaissent dans son développement soient du même signe. Par exemple pour la configuration

$$A <_x B <_x C$$
$$B <_y A <_y C$$

$$\begin{pmatrix} x_A - x_C & x_B - x_C \\ y_A - y_C & y_B - y_C \end{pmatrix} : \qquad \begin{pmatrix} x_B - x_A & x_C - x_A \\ y_B - y_A & y_C - y_A \end{pmatrix} :$$
$$\begin{pmatrix} - & - \\ - & - \end{pmatrix} \qquad \begin{pmatrix} + & + \\ - & + \end{pmatrix}$$

On voit ici que la première expression ne permet par de conclure. Par contre, l'expression de droite permet de conclure car les deux termes du déterminant sont positifs.

Une première sélection des triplés de points d'intérêt pourrait donc ce faire selon ce critère. On demande que les trois points se placent toujours dans une même configuration d'ordres (pour un groupe donné d'observations) et que cette configuration d'ordre soit fixe.

2.5 Analogie avec les méthodes de diagnostic clinique

Le problème de la différentiacion de groupes d'observations se pose dans d'autres domaines. On trouve par exemple dans [3] une étude systèmatique de ce problème dans le cadre de la spectrométrie de masse appliquée aux analyses médicales. Ici une observation est une courbe qui associe à chaque fréquence une intensité. La courbe reflète la composition biochimique de l'échantillon analysé. On recherche des empreintes caractéristiques d'un diagnostic. La méthode utilisée est détaillée dans les chapitres 2 et 3 de [3]. Nous survolons les principales étapes de cette méthode. Nous essayons de voir en quoi les contextes cryptographique et médicaux sont similaires ou différents. Seule une étude expérimentale permettrait de dire si cette méthode peut s'adapter à notre contexte.

2.5.1 Préparation des signaux

Avant d'être comparés les signaux subissent un premier traitement qui vise à filtrer les bruits haute fréquence. Un filtre passe-bas, ou un analogue multi-échelle à base d'ondelettes. D'autres corrections plus globales sont apportées : synchronisation, correction des bruits basse-fréquence (affaissement progressif du signal par exemple) et d'autres normalisations.

2.5.2 Détection de pics

C'est une étape importante car elle permet l'interprétation du signal. Les pics peuvent être bruités, ils peuvent se recouvrir. Pour contourner ces difficultés on accepte un très

grand nombre de pics candidats. On utilise ensuite des algorithmes de déconvolution pour séparer les pics qui se recouvrent. Une étape d'apprentissage peut-être utile ici, afin d'identifier la forme attendue de certains pics.

2.5.3 Alignement des pics

Il s'agit de reconnaître un même pic dans diverses observations d'un même groupe. C'est un problème de *clustering*. On cherche à regrouper les pics observés lorsqu'ils ont des caractéristiques assez proches (forme, hauteur, largeur, fréquence médiane, etc). On note que la fréquence médiane est importante ici car elle est caractéristique d'un composant organique. Dans un cadre cryptographique on étudie des courbes d'intensité en fonction du temps. Le temps joue alors le rôle de la fréquence, mais c'est un paramètre moins fiable, en raison de possibles retards ou permutations dues à des contremesures. Si l'on croît à la pertinence de paires de pics. On doit énumérer et rechercher ces paires au même titre que les pics eux-mêmes.

2.5.4 Sélection de pics

Une fois que le travail d'identification et d'alignement des pics a été réalisé dans chacun des deux groupes, on associe à chaque pic du groupe A un pic du groupe B. Deux pics sont appariés s'ils ont la même fréquence médiane (alignement). Une paire de pics est particulièrement significative si le pic dans le groupe A et le pic dans le groupe B sont très différents. On cherche les paires de pics les plus significatives car elles différencient bien les deux populations (patients sains ou malades par exemple). Dans notre contexte, aligner deux pics selon leur seule médiane est discutable en raison de possibles contremesures qui peuvent décaler ou permuter des pics.

3 Cryptographie en boîte blanche

Pour bien comprendre le but de la cryptographie en boîte blanche, il est nécessaire d'abord de bien connaître son contexte d'application. Ici, l'utilisateur d'une fonction cryptographique n'est pas considéré comme une personne de confiance, au contraire. Le but de la cryptographie en boîte blanche est de se prémunir face aux attaques menées par les utilisateurs légitimes des implémentations cryptographiques (reproduction et diffusion illégales d'un contenu acquis légalement, retro-ingénierie, etc.)

Si l'on pense à une fonction de chiffrement symétrique, l'adversaire est, normalement, modélisé comme un tiers qui maîtrise complètement l'exécution du code et le materiel sur lequel l'execution tourne, et qui vise à récupérer la clé secrète choisie et encapsulée par le programmeur/concepteur dans le code de la fonction.

Le but fondamental que l'on vise est donc de programmer une routine de chiffrement en sorte qu'elle ne laisse fuir aucune information sur la clé secrète.

De manière plus rigoureuse, soit $\mathsf{AES} : \mathbb{F}_{2^{128}} \times \mathbb{F}_{2^{128}} \to \mathbb{F}_{2^{128}}$, un algorithme de chiffrement symétrique $(k, m) \mapsto c = \mathsf{AES}_k(m)$. Une implémentation white-box d'AES pour une clé $\bar{k}$ donnée est la fonction : $\mathcal{F}_{\bar{k}} : \mathbb{F}_{2^{128}} \to \mathbb{F}_{2^{128}}$ telle que pour tout message m l'égalité suivante est vérifiée :

$$\mathsf{AES}_{\bar{k}}(m) = \mathcal{F}_{\bar{k}}(m) \; \forall m.$$

On demande aussi qu'il soit « difficile » d'extraire de l'information sur la clé secrète $\bar{k}$.

Nous commençons avec l'observation que la clé *doit* être contenue de façon masquée dans le code de la fonction et la sécurité de l'implémentation (comme *sécurité de la clé*) est basée donc sur l'obscurité de cette implémentation. Il s'agit donc d'une fonction d'AES offusquée. En général, il n'est pas simple de prouver que le code est « assez, bien » offusqué. Ainsi le niveau de sécurité reste heuristique.

Si l'on définit mieux notre adversaire, nous le modélisons comme capable de pouvoir extraire le code de notre implémentation et d'en maîtriser complètement l'exécution. Ceci signifie qu'il peut lire les données en entrée et sortie de toutes fonctions intermédiaires. Qu'il peut éventuellement les modifier ou manipuler l'ordre d'exécution des fonctions appelées par notre algorithme.

Pour lui rendre l'attaque difficile, il est nécessaire d'annuler ou de réduire les points de fuite d'informations. En particulier, il est possible pour l'adversaire d'analyser les sortie de toute fonction à n'importe quel moment. Il est indispensable donc de masquer les échanges de données d'une fonction à l'autre. L'idéal serait de donner un code qui intègre toutes les fonctions d'AES dans une seule. Mais cette dernière solution nécessiterait le stockage d'une table de valeurs bien trop grande. Il faudrait en effet écrire explicitement la permutation donnée par l'algorithme AES avec la clé secrète figée. Une telle solution reste théorique à cause de la taille d'un tel tableau, mais elle montre comment les points de fuite peuvent être annulés.

Si les calculs intermédiaires ne peuvent pas être éliminés en « regroupant » les calculs, une solution qu'on propose est de décorreler ces résultats de ceux qu'on attendrait dans une exécution *standard* de l'AES. Pour simplifier, imaginons que l'on est face à un adversaire qui récupère les données à la sortie de chaque tour. On peut utiliser des transformations inversibles A_i et les appliquer en entrée et sortie de chaque tour :

$$T_i \mapsto T_i' := A_i^{-1} \cdot T_i \cdot A_{i+1}.$$

Nous pouvons obtenir ainsi une description d'un tour de AES dans des coordonnées inconnues de l'adversaire. La composition des fonctions-tours donne bien la valeur attendue mais les résultats intermédiaires que l'adversaire récupèrera sont une *randomization* des valeurs attendues. Il est évident que de telles transformations A_i doivent rester inconnues de l'adversaire.

Il est possible d'introduire des transformations similaires à un niveau plus bas, en les composant avec les fonctions blocs de l'AES : SubBytes, MixColumns, ShiftRows.

Une idée que nous avons envisagée est de modifier la fonction MixColumns. D'un point de vue algébrique cette fonction correspond à une multiplication pour le polynôme $c(x) = \mathsf{0x03} \cdot x^3 + x^2 + x + \mathsf{0x02}$. Il est possible de remplacer le tableau de MixColumns par un nouveau tableau qui représente la multiplication pour le polynôme $g(x) = c(x) \cdot a(x)$ où $a(x)$ est un polynôme secret et inversible. Il sera nécessaire quelques étapes plus tard multiplier pour $a^{-1}(x)$ (ou une transformation de ce polynôme) pour retomber sur les valeurs attendues. La multiplication par ces nouveaux polynômes, composée avec l'utilisation des transformations de type A_i entre les blocs, permettrait de cacher le bloc où s'effectue MixColumns.

Le point le plus sensible est très probablement la fonction AddRoundKey car il s'agit du moment où la clé est manipulée directement. Il est possible d'écrire sous la forme de 16 tableaux la composition de AddRoundKey et de SubBytes.

3.1 Approche avec les Dual Ciphers

Dans [4] l'auteur propose d'utiliser les différentes descriptions de l'AES pour en protéger l'implémentation white-box. On observe que l'algorithme de chiffrement AES est composé de simples calculs dans le corps $\mathbb{F}_{2^8}$ et que, au moment de la standardisation, quelques choix ont été fixés. En utilisant des descriptions alternatives, il est possibles d'écrire quelques algorithmes AES-équivalent. De plus, étant donnés deux *dual ciphers*, pour un tour donné, il est possible d'écrire la bijection qui transforme un état de l'un en un état de l'autre.

3.2 Approche AES-*like* algorithmes

Cette approche est un peu différente de celles qu'on a vues avant. Il s'agit d'implémenter un algorithme qui n'est pas compatible avec l'AES mais qui a le même niveau de sécurité.

En étudiant la description algébrique de l'algorithme, on s'aperçoit que pour la définition du standard, le NIST a fait quelques choix, comme celui du polynôme irréductible pour la représentation du corps fini, la fonction non-linéaire pour le tableau SubBytes, ... Il serait possible de faire des choix différents pour obtenir un « nouvel » algorithme de chiffrement non compatible au niveau des entrées-sorties avec le standard, mais ayant le même niveau de sécurité.

Dans [1] on peut trouver une liste des différents choix possibles. Les auteurs analysent le cas des *dual ciphers*, mais il serait possible d'étudier aussi le cas des *semi-dual ciphers*. La différence étant que les transformations qui permettent de passer d'un schéma de chiffrement à l'un des ses duaux ne sont plus nécessairement inversibles.

Une telle implémentation perd la compatibilité avec les implémentations standards et demande donc l'écriture (et les tests) de fonctions de chiffrement et déchiffrement. Elle doit aussi être appliquée dans un contexte plus restreint ou mieux maîtrisé car les données en entrée doivent être bien formées.

Une remarque : si l'on observe la fonction d'expansion de clé, on note que les connaissances de la clé secrète ou des clés de tour sont équivalentes. Il est donc nécessaire d'éviter d'appliquer cette fonction dans la routine du code. On peut facilement précalculer les clés de tours et les utiliser directement dans l'implémentation.

Il serait possible d'ajouter une couche d'offuscation en changeant la fonction d'expansion de clé en la faisant dépendre d'un paramètre secret. À nouveau, il y a une perte partielle de compatibilité avec les standard, mais qui peut être, probablement, regagnée en écrivant la fonction de « passage » entre les deux clés.

Références

[1] Elad Barkan and Eli Biham. In how many ways can you write rijndael ? Cryptology ePrint Archive, Report 2002/157, 2002. http ://eprint.iacr.org/.

[2] V. Chepoi and B. Fichet. -approximation via subdominantsℓ_∞. *Mathematical Psychology*, (44) :600–616, 2000.

[3] Tim Conrad. *New statistical algorithms for the analysis of mass spectrometry time-of-flight mass data with applications in clinical diagnostics.* Thesis. Freie Universität Berlin, 2008.

[4] Mohamed Karroumi. Protecting white-box aes with dual ciphers. In *Proceedings of the 13th international conference on Information security and cryptology*, ICISC'10, pages 278–291, Berlin, Heidelberg, 2011. Springer-Verlag.

[5] Kévin Sol. La détermination du chirotope d'une base en fonction d'ordres sur les coordonnées de ses sommets, February 2011. groupe de travail ANR TEOMATRO.

Remerciements

Nous remercions chaleureusement Emmanuel Prouff et Emmanuelle Dottax, de la société Oberthur, pour le travail de préparation et de présentation de ces deux sujets.

Un grand merci à tous les organisateurs de la semaine, et en particulier à Simon Masnou.

Nous remercions Jean-Marc Couveignes pour son assistance et son travail de coordination et des nombreuses observations très utiles, en particulier, pendant la phase de rédaction.

Nous remercions Jean-Louis Nicolas et Gabriela Ciuperca pour leur assistance scientifique et leurs précieux conseils durant toute cette semaine.