(Texte draft février 2010)

Les télécommunications sécurisées comme je les vois

Emile Musyck

Ancien collaborateur scientifique à l'Université Libre de Bruxelles

Introduction

Les télécommunications ont atteint aujourd'hui un niveau technologique qu'on ne le pouvait pas imaginer encore il y a quelques dizaines d'années. Grâce à internet, il est possible d'envoyer ou de recevoir des gros fichiers avec une facilité déconcertante. Est ce pour cela qu'on peut considérer qu'on est arrivé à un point où tout est pour le mieux dans le meilleur des mondes? Malheureusement non, il faut subir les conséquences de graves manquements qui permettent aux virus et aux fichiers non sollicités de perturber les télécommunications utilisées par le grand public. Certes, il y a les antivirus et antispams qui aident efficacement les internautes, mais c'est une contrainte astreignante à assumer. Il est impensable qu'on puisse un jour arriver à éradiquer les virus par des actions d'ordre politique ou judiciaire. Il y a peut-être une solution technologique laquelle consiste à introduire le principe de la traçabilité de l'information et n'accepter qu'un fichier puisse entrer dans un ordinateur qu'à la condition de pouvoir vérifier l'enregistrement de l'identité de l'expéditeur auprès d'organismes reconnus comme organisme de confiance et cette possibilité doit être impérativement gratuite.

Notre étude se compose de quatre chapitres dans lesquels sont développés l'algorithme de chiffrement SED, le caractère aléatoire des résultats des chiffrements de l'algorithme SED, le cryptosystème ClassicSys lequel assure la gestion des clefs des messages chiffrés entre internautes et les perspectives futures: la traçabilité de toutes les informations sur Internet.

L'algorithme SED, algorithme de troisième génération?

L'évolution des microprocesseurs a fait l'objet d'une nomenclature bien définie en termes de générations. Il n'en est pas de même pour les algorithmes de chiffrement par bloc. Si la grandeur des blocs traités constitue probablement l'élément le plus approprié pouvant caractériser une génération de microprocesseurs, il va de soi que la grandeur des boîtes de substitution des algorithmes de chiffrement constituerait peut-être la référence la plus adéquate à classifier ces derniers.

La première génération, c'est le DES lequel est constitué de 8 boîtes (S1 ... S8) de substitution composées chacune de 4*16 nombres de quatre bits et opérant suivant le principe de Feistel. Dans ce cas, des trois éléments, le texte clair, le texte chiffré et la clef,

deux de ces éléments sont reliés par une relation linéaire tandis que la relation de deux autres éléments est non linéaire par la mise en place des boîtes S de substitution. Cette disposition permet de restituer le texte clair à partir du texte chiffré, mais comme les boîtes de substitution ne se composent que de nombres de 4 bits et que les blocs à chiffrer sont formés de 64 bits pour le DES, il importe d'effectuer de nombreuses rondes (16 rondes pour le DES) qui se chevauchent séquentiellement. La petite taille des boîtes de substitution est visiblement la cause des clefs faibles et semi-faibles.

Dans la seconde génération avec l'algorithme AES, on s'affranchit de la trop petite taille des boîtes de substitution du DES en introduisant le principe, utilisé aussi par d'autres algorithmes, entre autres certains du «Block Cypher Lounge», dans lequel les textes clairs et chiffrés et la clef se situent parmi quatre éléments lesquels répondent aux relations qui lient un dividende, un diviseur et un quotient et un reste. En attribuant des valeurs définies à trois éléments, le quatrième élément est automatiquement défini. C'est la filière qui permet de restituer le texte clair à partir du texte chiffré. Avec cette procédure, il n'est plus nécessaire de dénommer les boîtes comme dans le DES. Pour l'AES, les boîtes de substitution sont formées en quelque sorte par quatre ensembles de 32 bits. Ici aussi comme avec le DES, il importe de faire chevaucher les différentes opérations en effectuant dix rondes. Le nombre de rondes dans l'AES est une valeur qui a été estimée suffisante pour ne pas permettre une cryptanalyse linéaire ou différentielle, mais elle ne correspond pas au résultat d'un calcul ayant pour but de rechercher la valeur optimum.

La troisième génération est caractérisée par l'application de deux transformations consécutives opérées dans deux groupes multiplicatifs différents, mais les deux groupes ne peuvent pas présenter une relation d'isomorphisme. Tout nombre entier positif de 127 bits peut être considéré comme un élément faisant partie des N (= 2^n-1) états du LFSR : $X^{127} + X^{63} + 1$ (polynôme caractéristique). Ce nombre, ou texte clair d'entrée, a un certain logarithme discret et il est possible de calculer le nombre qui a comme logarithme discret, celui du nombre d'entrée multiplié par la clef k modulo N dans ce LFSR, et cela sans en connaître le logarithme discret en question. C'est en fait une exponentiation dans le corps fini défini par le LFSR. Le vecteur résultat ainsi obtenu existe également dans la séquence du LFSR: $X^{127} + X^{30} + 1$, une et une seule fois, mais avec un tout autre logarithme discret et peut également faire l'objet d'une seconde exponentiation dans le second LFSR. C'est le fait du passage du logarithme discret dans le résultat obtenu dans la première exponentiation au logarithme discret du vecteur d'entrée de la seconde exponentiation que l'on crée une boîte de substitution formée de n bits. Les deux polynômes caractéristiques susmentionnés sont primitifs car leurs racines standards forment des groupes multiplicatifs d'ordre N (=2^n-1), c'est à dire égal à la séquence maximum.

Les deux groupes multiplicatifs vérifient les trois axiomes d'associativité, d'identité et d'existence d'un inverse. De plus, les deux groupes sont différents par leur loi de formation ou d'ordonnancement des vecteurs allant de 0 à N (=2^n-1). Le vecteur ayant tous ses bits égal à "1" est défini comme ayant son numéro d'ordre ou plutôt son logarithme discret égal à zéro. Cette dernière appellation provient du fait que le passage d'un vecteur au suivant dans la loi d'ordonnancement est obtenu par la multiplication matricielle de la matrice compagnon par le vecteur à incrémenter. Chaque impulsion horloge au LFSR correspond à une incrémentation d'une unité du logarithme discret.

L'introduction d'un vecteur clair définit son logarithme discret appartenant au premier corps fini tandis que le vecteur résultat du chiffrement correspond au logarithme discret dans le second corps fini. La relation liant les deux logarithmes discrets comprend le produit deux multiplications par la clef k dans chaque corps fini mais également un

coefficient de substitution qui intervient par le passage du premier au second corps fini. C'est ce coefficient qui définit la boîte de substitution et qui a la grandeur de *n* bits.

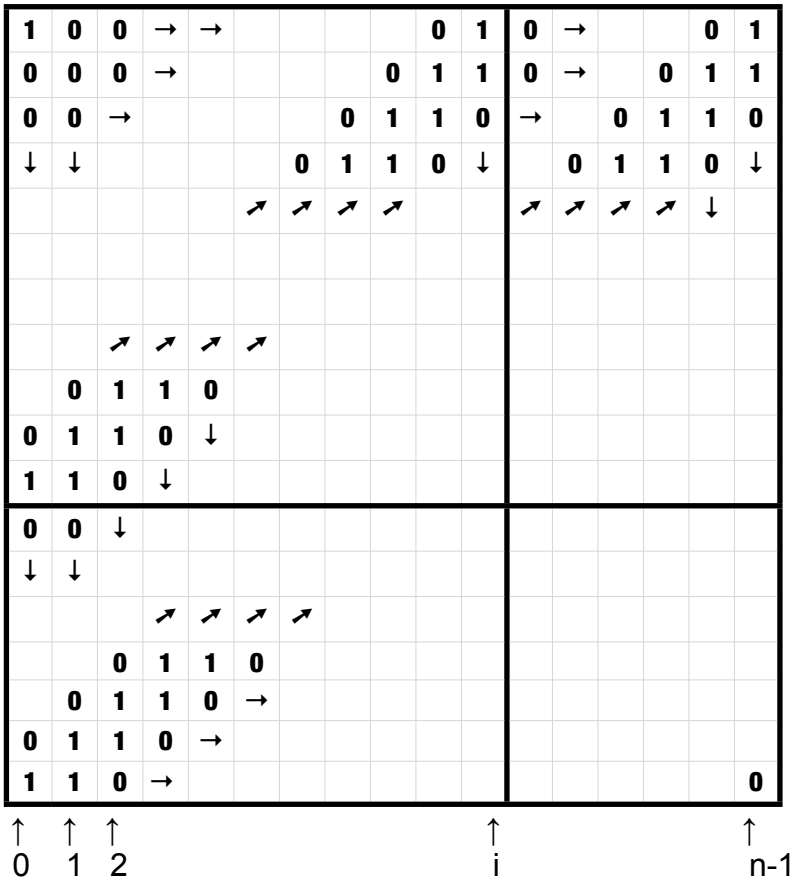
Dans l'algorithme SED, Il est possible d'utiliser *N* clefs *k* différentes ainsi que *N* vecteurs v_e d'entrée et le choix de la boîte de substitution utilisée sera défini par le produit (v_e * *k*) modulo *N*. En fait, les espaces des données et de la clef correspondent à 2^{128} éléments, mais le 128-ième bit n'intervient pas dans la sécurité.

Dans tous les algorithmes de chiffrement par bloc, la transformation du vecteur clair à chiffrer en le vecteur chiffré est réalisée par une suite d'extensions et de partitions. Pour l'AES, à chacune des dix rondes, il y a quatre ensembles de 32 bits qui subissent une extension suivie d'une partition et un brassage entre les quatre ensembles. Pour le SED, l'élévation à la puissance *k* du vecteur d'entrée dans chacun des deux groupes nécessite en moyenne (127 + 127/2) extensions et partitions en supposant que chaque bit de la clef a une probabilité de 1/2 d'être 1 ou 0. Vu le nombre plus élevé d'extensions et de partitions dans le SED et surtout l'aspect de la systématisé des opérations, il est logique d'admettre que la diffusion au sens donné par Shannon soit mieux réalisée avec le SED plutôt qu'avec l'AES.

La recherche du vecteur dont le logarithme discret est égal à la multiplication des logarithmes discrets opérée sur deux autres vecteurs dans les deux corps finis ne nécessite pas la connaissance proprement dite des logarithmes discrets. Chaque vecteur dv_i dans un corps fini peut être exprimé linéairement par un développement polynomial, par exemple, en fonction des n premiers vecteurs à partir du vecteur v_0, lequel a son logarithme discret égal à zéro: $dv_i = c_0 v_0 + c_1 v_1 + c_2 v_2 \ldots + c_{n-1} v_{n-1}$, les coefficients c_0 , c_1, c_2 ,. . . c_{n-1} étant des valeurs binaires. Dans la relation qui précède, le vecteur dv_i est exprimé en fonction de n vecteurs consécutifs à partir du vecteur v_0, mais il est possible en remplaçant le vecteur v_0 par le vecteur v_j, de calculer le vecteur $[v_{(i+j)}]$ qui a comme logarithme discret la somme des logarithmes discrets i et j de deux autres vecteurs $v_{(i)}$ et $v_{(j)}$. Pour ce faire, on établi la matrice carrée $C_{(j)}$, formée par les 127 vecteurs consécutifs générés par le LFSR à partir du vecteur $v_{(j)}$ mis en colonnes et on multiplie cette matrice par le vecteur $dv_{(i)}$ correspondant au développement polynomial du vecteur $v_{(i)}$ défini par rapport au logarithme zéro.

Le calcul du vecteur $dv_{(i)}$ du développement polynomial du vecteur $v_{(i)}$ s'effectue par la multiplication de la matrice inverse (S_0^{-1}) par le vecteur $v_{(i)}$. La matrice S_0 est formée des 127 vecteurs consécutifs au vecteur (111.. ..111) mis en colonnes et générée par LFSR. La matrice inverse est relativement simple et tous ses éléments ont des valeurs constantes pour tous les blocs à chiffrer. Tous les éléments de la matrice $C_{(j)}$ sont déterminés lorsque le vecteur $v_{(j)}$ est placé dans la colonne de gauche, les colonnes suivantes étant générées par le LFSR.

La matrice inverse S_0^{-1} relative au trinôme caractéristique $P^0 + P^i + P^n$ est donnée ci-après. Les flèches signifient la continuation des éléments 0 ou 1 placés suivant les colonnes, les lignes ou les diagonales.



Matrice inverse S_0^{-1} relative au trinôme caractéristique $P^0 + P^i + P^n$

On obtient ainsi la relation : $v_{(i+j)} = C_{(j)} * (S_0^{-1}) * v_{(i)}$. Le vecteur $v_{(i+j)}$ peut être calculé en une impulsion horloge dans un circuit intégré hardware. Si les bits de la clef écrite dans un développement binaire ont tous une probabilité de ½ d'avoir une valeur 0 ou 1, le chiffrement d'un bloc de 127 bits s'effectuera en moyenne en 127*(1+1/2)*2= 381 impulsions horloge.

Les vitesses des calculs faits en software du SED sont inférieures à celles de l'AES, mais tout de même acceptables. En hardware, avec la réalisation d'un circuit intégré, la vitesse de calcul serait égal en moyenne au tiers de la fréquence horloge, une ronde ayant lieu à chaque impulsion horloge. Cette réalisation hardware pourrait être réalisée dans une clef USB.

Le déchiffrement est obtenu dans l'algorithme SED en effectuant les deux DLM (Discrete Logarithm Multiplier) mais dans l'ordre inverse et en utilisant la clef k' de déchiffrement laquelle peut être calculée par la relation k*k' = 1 modulo 2^{127}-1.

L'étude « *A cryptosystem based on double discrete logarithm* » de Mieczislaw Kula (www.ulb.ac.be/di/scsi/classicsys/sedfrob.pdf) concerne l'algorithme SED, mais la description de l'algorithme ne correspond pas à la réalité. En effet, la fonction "lambda" entrant dans les formules (1) et (2) de l'étude de Kula est mal appropriée pour illustrer le passage de la sortie du 1er DLM à l'entrée du 2sd DLM. Il est préférable de considérer que le vecteur clair à chiffrer appartient à l'espace des 2^n-1 vecteurs, mais qu'il a également un logarithme discret par son appartenance au corps fini défini par le 1er DLM. Le résultat du calcul du 1er DLM crée le vecteur de transition vers le 2sd DLM ce qui

conduit ainsi à la mise en place d'un tout autre logarithme discret appartenant au corps fini du 2sd DLM. Le résultat du chiffrement est obtenu par la multiplication de ce nouveau logarithme discret par la clef k. Le vecteur résultat appartient au corps fini du 2sd DLM et donc aussi à l'espace des 2^n-1 vecteurs. La conclusion donnée par Kula concernant la possibilité de reconstituer la clef par $\sqrt{2^n}$ tests n'a pas pu être vérifiée pour des petites valeurs de n. D'autre part, son estimation de la vitesse de calcul égale à $O(n^3)$ n'est pas à prendre en considération étant donné que le chiffrement d'un bloc de 128 bits peut s'effectuer en moyenne en hardware en 381 impulsions horloge. L'erreur de Kula découle de l'application erronée de relations d'isomorphisme dans les corps finis. Ces relations sont bien réelles dans un seul corps fini, mais ne sont plus applicables lorsqu'on effectue deux exponentiations consécutives car on ne peut pas intervertir l'ordre des facteurs dans un produit matriciel. Les relations d'isomorphisme seraient parfaitement applicables dans le cas de deux exponentiations arithmétiques modulaires. Vu que l'auteur n'a pas donné suite à une demande de dialogue pour clarifier les choses, je considère que les conclusions de son étude doivent être considérées comme nulles et non advenues.

Avec le SED, le choix des éléments mathématiques découle du fait de disposer de plusieurs trinômes caractéristiques différents dans les corps finis 2^{127}-1, et d'adopter les deux trinômes primitifs ayant les exposants des termes intermédiaires qui sont les plus grands. Tout comme avec le RSA où on effectue des exponentiations, on ne pourrait concevoir l'existence d'une porte à la dérobée.

On pourrait se demander quel est l'intérêt de faire appel à un algorithme de chiffrement de troisième génération étant donné que l'AES à été désigné comme le meilleur algorithme par une commission composée des membres les plus éminents au monde dans ce domaine? Certes le SED ne peut pas rivaliser l'AES en ce qui concerne la vitesse de chiffrement en software car le volume des calculs est nettement plus grand, mais il n'en est pas de même en ce qui concerne sa transparence. Le calcul de chacun des 381 états intermédiaires d'un chiffrement nécessite l'application de 16129 portes "AND" et 16509 portes "EXOR". Dans la réalisation d'un circuit intégré, toutes les portes AND et EXOR opèrent à la même impulsion horloge. A première vue, on pourrait trouver que ce système semble bien compliqué, mais c'est le prix à payer pour pouvoir déclarer que le caractère aléatoire du bloc chiffré par rapport au bloc clair, défini par le test universel (voir paragraphe suivant) a atteint un certain niveau d'excellence et qu'il est possible d'en avoir une certitude mathématique.

La réalisation d'un algorithme de chiffrement composé d'une seule boîte de substitution ayant la même grandeur que les données pose un problème particulier: l'algorithme ne peut pas constituer un groupe, au quel cas il pourrait être possible de recalculer la clef à partir des données.

L'algorithme SED peut également réaliser la fonction de hachage et créer, comme résultat, un bloc unique appelé empreinte. Toute modification d'un des bits de l'ensemble des blocs donne lieu à la modification du bloc empreinte, et de plus, il n'est pas possible de créer un autre ensemble de blocs qui donnerait la même bloc empreinte. Tous les blocs de l'ensemble des blocs sont chiffrés en cascade, la clef de chiffrement de chaque bloc étant égale au bloc lui-même, de plus, à chaque chiffrement, le résultat du premier DLM est modifié par l'application XOR du résultat du chiffrement du bloc précédent. Le premier chiffrement (qui n'a pas de chiffrement précédent) ne subit pas de modification XOR et le résultat du dernier chiffrement constitue le bloc empreinte.

Caractère aléatoire du résultat des chiffrements de l'algorithme SED

Il est logique d'admettre que l'algorithme de chiffrement le plus robuste vis-à-vis d'une cryptanalyse linéaire ou différentielle sera celui qui présente le meilleur caractère aléatoire du bloc chiffré par rapport au bloc clair. En fait, le problème de la robustesse d'un algorithme de chiffrement sera résolu par un autre problème: comment évaluer valablement le caractère aléatoire du bloc chiffré par rapport au bloc clair.

Le générateur de nombres pseudoaléatoires mathématiquement parfait n'existe pas, mais il est possible de s'en approcher de très près et de définir les caractéristiques que ce générateur virtuel devrait présenter aux tests d'évaluation. Je voudrais ici introduire un nouveau test inédit, appelé "Test Universel" et qui pourrait remplacer, à lui seul, tous les tests conventionnels comme ceux, par exemple, cités par Donald Knuth. (*The art of computer programming Vol. 2*). L'appellation "Universel" fait référence au fait, que la condition nécessaire et suffisante pour qu'un algorithme de nombres pseudoaléatoires puisse être considéré comme acceptable, est de passer valablement le test universel.

Un nombre composé de *n* bits sera strictement aléatoire si chaque bit présente une probabilité exacte de 1/2 d'être 1 ou 0 lors à chaque tirage. Cette exigence est matériellement impossible à réaliser par un algorithme car tous les systèmes pseudoaléatoires constituent des fonctions déterministes. La probabilité qu'un nombre déterminé de *n* bits soit obtenu lors d'un tirage est de $p = (1/2)^n$ en vertu du fait que les probabilités de chaque bit sont strictement indépendantes les unes des autres. C'est une valeur très petite qui peut être introduite dans la loi binomiale où la probabilité d'un succès est de *p* et celle d'un échec est de (*1-p*). En effectuant *N* ($=2^n$) tirages, on aura une probabilité en moyenne de "*1*" ($= 2^n * 1/2^n$) d'obtenir l'apparition d'un nombre déterminé, mais on aura aussi la possibilité d'obtenir zéro, deux, trois, ...*i*, fois le même nombre en question. Dans ces hypothèses, on peut calculer les probabilités à l'aide de la loi de Poisson pour obtenir pour chacun de ces 2^n nombres qu'il soit choisi: zéro, une, deux, trois, ... *i* fois,... après *N* tirages. En recensant tous les résultats des *N* tirages et en les classant suivant les cas de l'apparition de zéro, de une, de deux, de trois, ... de *i* fois le même nombre, on obtient respectivement les totaux de *N*/e (e = 2.71828...) pour zéro fois, *N*/1!*e pour une fois, *N*/2!*e pour deux fois, *N*/3!*e pour trois fois, ... , *N*/i!*e pour *i* fois. Le nombre de totaux est limité à "*i*", lequel constitue le plus grand nombre pour qu'un même nombre soit obtenu lors de *N* tirages. Chacun de ces totaux expérimentaux varie avec une variance théorique égale à la racine carrée du total théorique en question.

La qualité du générateur pseudoaléatoire expérimental sera mesurée par le rapport de l'écart expérimental vis-à-vis de l'écart quadratique théorique pour chaque total et exprimé en unité sigma. Pour exprimer les "*i*" résultats en un résultat unique, on effectue la moyenne pondérée de tous les écarts expérimentaux par rapport au total *N* et on obtient ainsi l'évaluation de l'écart du générateur en question vis-à-vis du générateur mathématiquement parfait.

L'évaluation de l'équidistribution de tous les bits les uns par rapport aux autres d'un algorithme pseudoaléatoire nécessite une exploration exhaustive de tous les tirages effectués. Si la longueur de la série des nombres aléatoires reste en de ça d'une certaine limite qui permet matériellement d'effectuer une exploration exhaustive, le test universel donne le résultat par un seul coefficient en unité sigma. Par contre, si la longueur de la série est trop grande pour effectuer une exploration exhaustive, on devra se contenter

d'une approche réalisée par un grand nombre de tests pour lesquels on pourra établir une moyenne accompagnée de l'écart quadratique moyen des résultats.

L'algorithme de chiffrement SED est constitué par l'exponentiation au sein de deux corps finis différents, caractérisés par un même nombre *n* de bits par bloc. Il présente l'avantage de pouvoir effectuer une simulation du comportement d'un algorithme avec des blocs plus petits que 127, permettant ainsi d'effectuer une exploration exhaustive et de donner ainsi une tendance. Le nombre *n* peut prendre les valeurs aisées de 7 et 17 bits où nous disposons des trinômes caractéristiques $P^0+P^1+P^7$ et $P^0+P^3+P^7$ pour *n*=7 et $P^0+P^5+P^{17}$ et $P^0+P^6+P^{17}$ pour *n*=17. Pour l'algorithme SED proprement dit, il est fait usage des trinômes $P^0+P^{30}+P^{127}$ et $P^0+P^{63}+P^{127}$, mais dans ce dernier cas, plus aucune exploration exhaustive n'est possible.

Pour n=7 bits, on procède à 126 ($=2^7-1$) tirages et les valeurs des nombres vont de 1 à 126. Théoriquement on trouve une distribution de probabilité suivant la loi de Poisson lorsque la moyenne est égale à *1* comme suit:

- 46,73 (=127/2.71828 (=*e*)) cases vides avec un σ de 6,83 ($= 46,73^{1/2}$);
- 46,73 cases avec un seul nombre et un même σ de 6,83;
- 23,36 (=46,73/2) cases avec 2 nombres et un σ = 4,83;
- 7,79 (=23,36/3) cases avec 3 nombres et un σ = 2,79;
- 1,95 (=7,79/4) cases avec 4 nombres et un σ = 1,40;
- . . .

Expérimentalement, on trouve les résultats suivants: (les écarts par rapport à la moyenne sont exprimées dans l'unité (σ) (écart quadratique) :

- 54 (=46,73+1,27σ) où (1,27=(54-46,73)/6,83) cases vides;
- 41 (=46,73-0,84 σ) où (-0,84=(41-46,73)/6,83)) cases avec 1 nombre;
- 18 (=23,36-1,11 σ) où (-1,11=(18-23,36)/4,83)) cases avec 2 nombres;
- 9 (=7,79+0,43 σ) où (0,43=(9-7,79)/2,79)) cases avec 3 nombres;
- 1 (=1,95-0,68 σ) où (-0,68=(1-1,95)/1,40)) cases avec 4 nombres;
- 1 case avec 5 nombres;
- 1 case avec 6 nombres;
- 1 case avec 7 nombres.

On vérifie que l'on a: (54*0)+(41*1)+(18*2)+(9*3)+(1*4)+(1*5)+(1*6)+(1*7)= 126

Les mêmes calculs expérimentaux ont été réalisés pour *n*=17 et on obtient les résultats expérimentaux suivants: (entre parenthèses, le nombre "*i")*.

48455 *(0)*, 48055 *(1)*, 23927 *(2)*, 8047 *(3)*, 2068 *(4)*, 448 *(5)*, 67 *(6)*, 10 *(7)*, 0 *(8)*, 1 *(9)*, 0 *(10)*, 1 *(11)*, 0 *(12)*, ... , 0 *(15)*, 1 *(16)*. Les valeurs de σ sont données ci-après.

Calculs de σ pour *n*=7		Calculs de σ pour *n*=17	
i = 0;	+1,27 σ	*i* = 0;	+1,08 σ
i = 1;	- 0,84 σ	*i* = 1;	- 0,74 σ
i = 2;	- 1,11 σ	*i* = 2;	- 1,17 σ
i = 3;	+0,43 σ	*i* = 3;	+ 0,12 σ

i = 4;	- 0,68 σ	i = 4;	+ 1,31 σ
i = 5;	. . . . .	i = 5;	+ 2,30 σ
		i = 6;	0,00 σ
		i = 7;	+ 0,14 σ
		i = 8;	. . . . .

Les 5 et 8 valeurs de σ pour n=7 et n=17 peuvent être converties chacune en un résultat unique par le calcul d'une moyenne pondérée des valeurs de σ, ce qui donne comme résultats les valeurs de 1,01 σ (n=7) et 0,90 σ (n=17). Comme il fallait s'attendre, nous constatons que les statistiques s'améliorent en passant pour n de 7 à 17.

Pour le mathématicien, la présence de nombres correspondants aux grandes valeurs de "i" sont dérangeantes car elles sortent des statistiques valables de la loi de Poisson. Pour le cryptographe, ces nombres ne constituent pas une menace pour la sécurité. En effet, le rapport des valeurs résiduelles à (2^n-1) passe de 0,14 (=(5+6+7)/127) (18 nombres sur 127 à exclure) pour n=7 à 0,00027 (=(9+11+16)/131071) (36 nombres à exclure sur 131071) pour n=17 et le diviseur de ce rapport suit une loi exponentielle avec i.

Pour améliorer le caractère aléatoire du bloc chiffré par rapport au bloc clair, nous nous proposons d'utiliser des polynômes caractéristiques composés de cinq termes à la place des trinômes caractéristiques pour la création des deux corps finis.

Pour n = 7, nous avons choisi les deux polynômes caractéristiques $P^7=P^0+P^1+P^2+P^3$ et $P^7=P^0+P^1+P^2+P^5$ ce qui donne comme résultats pour "*i"* allant de 0 à 6 les nombres 48, 43, 29 ,5, 1, 0, 1. Pour les 126 tirages, il y a 6 nombres à exclure. La moyenne pondérée des σ est égale à 0.55 σ, valeur à comparer à 1.01 σ obtenue précédemment. Nous gagnons presque un facteur de 2.

Pour n = 17, nous avons choisi les deux polynômes caractéristiques suivants : $P^{17}=P^0+P^2+P^6+P^7$ et $P^{17}=P^0+P^2+P^3+P^5$ ce qui donne comme résultats pour "i", allant de 0 à 13, les nombres: 48140, 48425, 24029, 7990, 1995, 407, 71, 16, 4, 1, 1, 0, 0, 1. Pour les 131070 tirages, il y a 13 nombres à exclure, alors que précédemment, il y avait 36 nombres à exclure. La moyenne pondérée des σ est égale à 0.52 σ, valeur à comparer à 0.90 σ obtenue précédemment.

Nous avons effectué une simulation du SED pour n = 19 avec les polynômes caractéristiques $P^0+P^3+P^7+P^{10}+P^{12}+P^{13}+P^{14}+P^{15}=P^{19}$ et $P^0+P^2+P^4+P^5+P^7+P^8+P^{14}+P^{15}+P^{19}$ et avons obtenu le résultat de 0,173 σ. Une autre simulation exécutée avec les polynômes $P^{\mathbf{0}}+P^2+P^5+P^7+P^9+P^{11}+P^{13}+P^{15}+P^{16}+P^{17}=P^{19}$ et $P^0+P^2+P^4+P^6+P^8+P^{10}+P^{12}+P^{14}+P^{16}+P^{18}=P^{19}$ où une majorité des exposants pairs et impairs ont été regroupés séparément dans les deux exponentiations, nous avons obtenu le mauvais résultat de 0,936 σ. Nous avons également effectué une simulation avec trois exponentiations comprenant les polynômes $P^0+P^3+P^7+P^{10}+P^{12}+P^{13}+P^{14}+P^{15}=P^{19}$, $P^0+P^2+P^4+P^5+P^7+P^8+P^{14}+P^{15}=P^{19}$ et $P^{\mathbf{0}}+P^2+P^5+P^7+P^9+P^{11}+P^{13}+P^{15}+P^{16}+P^{17}=P^{19}$. Le résultat était de 0,2019 σ, donc moins bon que celui du test avec seulement les deux exponentiations.

Des tests portant sur des séries de 131071 ($=2^{17}$ -1) nombres composés de 17 bits extraits des résultats de chiffrements à l'aide de l'algorithme SED de 128 bits donnent les résultats sigma suivants: 0,301, 0,670, 0,805, 0,450, 0,220, 0,315 et 0,508 ce qui correspond à une moyenne de 0,467 sigma.

De ce qui précède, nous pouvons dire que tout chiffrement effectué avec une même clef peut faire intervenir en moyenne 2^{127} / *e* (*e* = base des logarithmes népériens) boîtes de substitution différentes, $2^{127}/2*e$ boîtes qui se présentent au plus deux fois,.... et $2^{127}/ i! *e$ boîtes qui se présentent *i* fois. En se donnant un couple de données claire et chiffrée, il existe en première approximation une probabilité de $e/2^{127}$ qu'une seule clef réponde à cette hypothèse. Les performances d'un algorithme sont le plus souvent évaluées qu'en ce qui concerne la vitesse de chiffrement. Ne serait-il pas plus judicieux de s'en quérir par un raisonnement mathématique à ce que le caractère aléatoire du bloc chiffré par rapport au bloc d'entrée soit le plus proche du générateur pseudoaléatoire parfait vu le nombre de bits dont se composent les données.

Télécommunications sécurisées avec le logiciel ClassicSys

Les systèmes cryptographiques sont des structures informatiques permettant à deux internautes quelconques de se communiquer en mode chiffré en ayant pour chacun d'eux l'assurance que l'identité du correspondant en ligne soit exacte, et cela dans une convivialité optimum. La problématique d'une utilisation intelligente de la cryptographie nous conduit vers le choix de deux options. Comme tout système cryptographique implique la création et le gardiennage des clefs secrètes, les clefs peuvent être générées, soit par les utilisateurs eux-mêmes, c'est la filière RSA avec hébergement des clefs publiques auprès d'une autorité de certification, soit recourir aux services d'un organisme de confiance tel que le prévoit le standard ANSI X9.17 (Voir *Handbook Applied Cryptography Ch. 13.2.3 voir: http://www.cacr.math.uwaterloo.ca/hac/*) qui distribue les clefs de session. Cette dernière solution comme telle ne s'est jamais bien concrétisée pour diverses raisons: coût du service de l'autorité de certification, obligation de garder en mémoire un nombre trop élevé de clefs de session, absence de logiciels conviviaux, crainte de utilisateurs de l'utilisation malveillante des clefs secrètes. Le système ClassicSys constitue une première ébauche d'un système effectuant les services de l'organisme de confiance dans un nouvel environnement cryptographique. Le texte de l'*Handbook* fait remarquer que si *n* internautes sont affiliés à un organisme de confiance, il y aura n^2 clefs de session à garder en mémoire et ce nombre pourrait devenir excessif pour *n* très grand. Dans le système classicSys, cette contrainte a pu être évitée en donnant à chaque affilié une adresse publique composé de 11 caractères hexadécimaux. La clef de session entre deux affiliés, créée par le serveur de l'Organisme de Confiance, est obtenue par le calcul d'une fonction à sens unique convertissant les deux adresses publiques des deux affiliés en un bloc de 128 bits. Toute demande adressée à l'Organisme de Confiance faisant référence à une clef privée ou une clef de session (mise à jour, certificat, ...) nécessite de recalculer la clef à partir des adresses publiques des deux correspondants.

La comparaison de la sécurité informatique des deux options découle du raisonnement suivant. Dans tout cryptosystème qui assure la gestion des clefs entre les internautes, il y a des éléments publiques associés à des éléments secrets. Dans le cas d'un cryptosystème basé sur l'utilisation de l'algorithme RSA, il y a les clefs secrètes associées aux clefs publiques. La protection de la clef secrète par rapport à la clef publique est obtenue par un pare-feu d'ordre mathématique, c'est la difficulté à factoriser un produit de deux facteurs premiers très grands. La clef secrète sera protégée aussi

longtemps qu'il n'est pas possible de factoriser le produit susmentionné. Pour le crypto-système ClassicSys, l'élément public est l'adresse publique de l'internaute et l'élément secret est en quelque sorte la clef secrète associée à cette adresse publique. Le pare-feu est constitué par le fait que c'est un serveur situé à distance qui détermine la clef secrète et que ce serveur n'accepte de délivrer des informations sensibles en ligne que si toutes les conditions de sécurité sont présentes. Les informations échangées sont chiffrées à l'aide de la clef secrète de l'internaute. La sécurité de ClassicSys sera acquise aussi longtemps que la clef secrète du serveur restera protégée physiquement à l'égard d'un opposant. La crédibilité d'une signature électronique revendiquant l'auteur d'un message sécurisé repose sur deux éléments: la clef secrète de l'auteur de la signature ne peut pas être cassée en connaissant de la clef publique, et d'autre part, l'identité du propriétaire de la clef secrète a été vérifiée à partir d'un document officiel et enregistrée par une procédure manuelle. Les deux cryptosystèmes susmentionnés répondent à ces deux exigences et devraient bénéficier de l'appellation de *token* cryptographique (voir http://fr.wikipedia.org/wiki/Authentification_forte). L'authentification forte nécessite de garantir l'accessibilité, la confidentialité, l'intégrité, la traçabilité, et une possibilité de révocation.

A la procédure de l'affiliation, l'internaute donne son identité et certaines informations communiquées par son FAI afin qu'il puisse recevoir et envoyer des mails chiffrés. Au cours du premier échange d'informations, l'Organisme de Confiance et l'internaute utilisent une clef provisoire déterminée par le processus Diffie-Hellman et au second échange, c'est la clef privée de l'internaute qui est utilisée. Lors de la demande d'une clef de session de l'internaute expéditeur pour correspondre en mode chiffré avec l'internaute destinataire, l'Organisme de Confiance fait parvenir plusieurs blocs chiffrés. Les informations sensibles pour l'internaute expéditeur sont déchiffrées à l'aide de sa clef privée tandis que les informations concernant la clef de l'internaute destinataire transitent par l'internaute expéditeur et sont déchiffrées consécutivement à l'aide des clefs privées de l'expéditeur et du destinataire. Il y a également vérification des blocs servant à l'authentification des adresses publiques de l'expéditeur et du destinataire.

Toute la sécurité concernant la connaissance de l'identité du correspondant dans un échange de mail repose sur le fait que lors de la procédure de l'affiliation, le candidat affilé aura décliné sa réelle identité sans qu'il y ait une possibilité de fraude. La solution à ce défit serait de communiquer l'identité à l'instar de la méthode utilisée par le contribuable qui rédige et envoi sa feuille de contribution par internet. En effet, la carte d'identité belge comporte une puce et le contenu de cette puce peut être lu et communiqué par internet.

A l'envoi d'un émail sécurisé, le texte clair de l'expéditeur fait l'objet du calcul d'un bloc "empreinte" de chiffrements chaînés et le même calcul est aussi effectué à la réception. Si les deux blocs ne sont pas identiques, le logiciel signale que les données sont altérées.

Tout internaute est désireux de voir ses é-mails chiffrés avec une vérification de l'identité des correspondants, mais si 99,99% des mails sont toujours envoyés en clair, c'est qu'il y a une entrave d'ordre financier alors que 99,99% des envois postaux sont effectués sous pli fermé. La création d'un organisme public de confiance pour une distribution gratuite des clefs de session entre internautes serait la solution du bon sens.

Quels sont les critères nécessaires que doit présenter un organisme de confiance? La première exigence est que l'organisme soit un service public non marchant. C'est à l'Etat qu'il incombe de donner une identité à chaque citoyen lors de son enregistrement aux services de l'Etat civil. Avec les développements de l'informatique que l'on connaît

actuellement, les informations relatives à l'identité des citoyens sont toutes informatisées. Ne serait-il pas normal que le prolongement informatique qui lie les identités de deux citoyens par la mise en place d'une clef de session, soit donné gratuitement aux citoyens en question pour qu'ils puissent se communiquer en mode chiffré. A la rigueur, une société telle que Google implantée mondialement et connue pour ses nombreux services gratuits offerts aux internantes pourrait peut-être également convenir.

On pourrait reprocher que les responsables de l'organisme de confiance seraient facilement capables de déchiffrer les mails échangés. Ceci n'est pas exact, tout dépend de la structure des logiciels de l'organisme de confiance et de l'endroit où le serveur de l'organisme de confiance se situe. En fait, toute demande adressée au serveur de l'organisme de confiance doit réponde à certaines exigences de codes et si les codes ne sont pas respectés, la demande est automatiquement rejetée. Dans toutes les démocraties où il existe la séparations des pouvoirs exécutif, législatif et judiciaire, il existe des procédures autorisant l'ouverture de correspondances suspectes concernant le terrorisme, la criminalité, la drogue ou le blanchiment de l'argent. Il est possible de prévoir un code d'accès mis à la disposition des services compétents pour l'ouverture d'une correspondance suspecte. Chaque demande de *key escrow* doit être enregistrée avec un numéro d'ordre non effaçable par les opérateurs, mais pouvoir être examinée par d'autres services afin d'éviter toute dérive. Autre objection contre l'organisme de confiance: en mettant tous les oeufs dans le même panier, on courre le risque en cas de défaillance informatique de provoquer la catastrophe générale mais ceci peut-être évité en mettant plusieurs serveurs opérant en parallèle.

Le cryptosystème expérimental Classicsys se compose des trois éléments:

1) le TA principal (Trust Authority) possède une clef secrète et a pour mission de transformer une information claire reçue du TA serveur et de la retourner en mode chiffré,

2) le TA serveur proprement dit reçoit par internet les messages des l'internautes et dispose d'une liaison chiffrée avec le TA principal et le TA administratif. Ces messages concernent la demande d'une adresse publique, d'une clef secrète, des mises à jour et des certificats d'authenticité.

3) le gestionnaire du TA administratif tient à jour les données non secrètes des internautes et effectue l'enregistrement des affiliés par la vérification de la concordance de l'identité du futur affilié qu'il a donnée lors de son affiliation et celle figurant sur un document officiel. L'affilié obtient ainsi le statut d'affilié enregistré. Le gestionnaire a autorité pour radier un affilié le cas échéant en cas de révocation.

Le TA principal et TA serveur se trouvent dans un local protégé physiquement à l'Université Libre de Bruxelles et opèrent 24h/24h. Le TA administratif se trouve soit au bureau, soit au domicile du gestionnaire. Le logiciel d'affiliation est téléchargeable à l'adresse http://www.ulb.ac.be/di/scsi/classicsys/experim.htm et fonctionne 24h/24h. Dans cette optique, tous les mails envoyés et reçus entre des affiliés sont chiffrés avec signature électronique. La gestion des clefs s'exécute automatiquement sans aucune intervention des utilisateurs. Le TA serveur peut délivrer un certificat d'authentification de la signature électronique de l'affilié expéditeur.

Perspective future: la traçabilité de l'information sur Internet.

Nous avons vu la possibilité d'utiliser le système ClassicSys pour chiffrer les mails sans que les correspondants ne doivent se préoccuper de la gestion des clefs. Ce qui est valable pour les mails le serait également pour toute demande d'informations à une quelconque URL grâce à la traçabilité qui accompagne les informations. Pour concrétiser cette idée, il impératif que l'internaute navigateur et le serveur sollicité aient chacun une adresse publique auprès d'un organisme de confiance et que le serveur soit enregistré. La demande d'une clef de session doit pouvoir être exécutée un temps très court, soit une très petite fraction de seconde et la vitesse de chiffrement doit être suffisamment élevée pour ne pas donner lieu à une majoration du délai de la réception de l'information demandée. Toute la sécurité informatique repose sur le fait que l'ordinateur n'accepte une information venant d'internet que si celle-ci ait été chiffrée avec une clef de session d'un auteur enregistré, en d'autres termes, une information extérieure ne pourra être lue qu'après vérification faite de la traçabilité de cette information.

Le chiffrement des échanges d'informations pourrait s'effectuer par le scénario suivant. La demande de l'URL faite par l'internaute auprès du serveur doit être accompagnée du vecteur estampille. Ce vecteur de 128 bits n'est pas secret et comprend en clair l'adresse publique de l'internaute, la date de son affiliation et le statut du vecteur. Rappelons que le vecteur estampille, chiffré avec la clef secrète du TA principal donne la clef privée de l'internaute. A la réception du vecteur estampille de l'internaute par le serveur sollicité, celui-ci introduit une demande au TA serveur pour recevoir la clef de session allant du serveur sollicité à l'internaute. Cette demande comprend trois vecteurs de 128 bits qui constituent les vecteurs estampilles de l'internaute et du serveur, ainsi que le vecteur complémentaire de estampille du serveur sollicité modifié bit à bit par la fonction XOR et chiffré par la clef secrète de l'organisme de confiance. Ce dernier vecteur aura été donné au serveur sollicité par le TA serveur au moment de son enregistrement et constitue pour ce serveur un moyen rapide pour vérifier de l'enregistrement du serveur sollicité. Avec ces trois vecteurs, l'organisme de confiance peut calculer les clefs de session de chiffrement et de déchiffrement pour le serveur sollicité et l'internaute à l'instar de ce qui se passe lors d'une demande d'une clef de session pour des é-mails. Il importe de constater que le serveur de l'organisme de confiance ne fait que des calculs et n'a pas besoin de rechercher des informations dans sa base de données, ce qui accélère le processus. Avec une réalisation en software de l'organisme de confiance, le temps de ces calculs serait une très petite fraction de seconde.

Conclusions

L'utilisation du "test universel" pour l'évaluation du caractère aléatoire du bloc chiffré par rapport au bloc clair dans l'algorithme SED permet de situer la robustesse de cet algorithme vis-à-vis d'une cryptanalyse par rapport aux autres algorithmes de chiffrement.

L'algorithme RSA, qui permet d'assurer la fonction à sens unique des chiffrements dans les transmissions sécurisées conventionnelles, est remplacé dans le système ClassicSys par le dialogue intelligent entre l'internaute et un organisme de confiance à distance, ce qui permet d'assurer toutes les exigences d'un *token* cryptographique, c'est à dire garantir les fonctions d'accessibilité, de confidentialité, d'intégrité et de traçabilité avec la possibilité de faire appel à la révocation. La version expérimentale de ClassicSys permet de vérifier la faisabilité d'un cryptosystème assurant le rôle d'un *token*

cryptographique. Il est évident que 99,99% du courrier conventionnel postal s'effectue sous pli fermé car cela correspond au désir des gens. Ce même chiffre de 99,99% correspond vraisemblablement au volume des mails qui passent toujours en clair. Cette différence ne peut s'expliquer que par l'entrave financière subie par l'internaute. Il n'existe aucun logiciel gratuit qui permet d'opérer automatiquement en mode de chiffrement avec la clef de session pour un contact défini, avec la possibilité d'une certification du mail, rien qu'en cliquant sur ce contact. Pour mettre en oeuvre ce but, ce n'est plus un problème d'ordre technique à résoudre mais plutôt d'ordre légal. En d'autres termes, il y a lieu d'introduire et de faire accepter le concept d'un l'organisme de confiance dont la gestion soit assurée par un service publique à remplir cette mission.

L'adoption généralisée par les internautes des moyens pour sécuriser toutes les informations reçues et envoyées sur le net par le biais de clefs de session données par un organisme de confiance, permettrait d'effectuer la traçabilité de toutes ces informations entraînant de ce fait l'élimination de tous les virus ou autres spam non sollicités.

Remerciements

Mes premiers remerciements vont à l'Université Libre de Bruxelles pour l'accueil qui m'a été réservé lors de mon premier contact avec le professeur Yves Roggeman ainsi que pour les nombreuses discussions qui ont suivies. Egalement des remerciements pour l'hébergement du système ClassicSys.

Viennent ensuite les remerciements pour Luc Binard, ancien collègue au Centre d'Etude de l'Energie Nucléaire de Mol, qui fut pendant plus de vingt ans a assuré la cheville ouvrière de toute la programmation des logiciels de ClassicSys.

Et pour terminer, je ne peux manquer de citer ma très chère épouse Marcelle Musyck-Quinet qui m'a aidé, épaulé avec patience et compréhension, jour après jour et a consacré pas mal de temps pour corriger de nombreux textes, et bien que n'étant pas une scientifique, elle m'a soutenu, orientant mes réflexions avec son bon sens et sa lucidité.

Emile Musyck