

Cryptage pyramidale Quelques explications

Cet article sert à détailler la méthode pyramidale pour ceux qui trouverait l'article de base un peu difficile à lire (en particulier à cause des indices très nombreux).

Il est basé sur un exemple qui n'en est un qu'à moitié, car on ne fera pas tous les calculs (qui sont inintéressants au possible). On travaillera avec $p=5$ c'est à dire dans F_5 (ce sont les nombres de 0 à 4 pour lesquels tous les calculs sont réalisé modulo 5). On prendra $n=8=2^3$.

1- Passage dans F_5

Pour utiliser le cryptage il faut convertir un texte fait d'octet (i.e. nombres de 0 à 256) en élément de F_5 . Il y a forcément un peu de perte. Nous ne chercherons pas à optimiser ici.

De façon très simple on va coder le mot 'NO' en conversion ascii-binaire,

On a 'N'= b01001110 et 'O'= b01001111 . On passe dans F_5 très simplement en injectant les nombres par groupement de deux bits. Le 4 de F_5 n'est jamais utilisé par cette méthode, ça ne gêne en rien pour le cryptage. Par ce groupement le codon de base à coder sera :

$t_1=^b01=1$, $t_2=^b00=0$, $t_3=^b11=3$, $t_4=^b10=2$, $t_5=^b01=1$, $t_6=^b00=0$, $t_7=^b11=3$, $t_8=^b11=3$
d'où en résumé $T=(1,0,3,2,1,0,3,3)$

2-Schéma de base

Le schéma de base est $A(x,y)=\sum_{i,j=0..4} a_{i,j} x^i y^j$, lorsqu'on entre 2 nombres x et y il ressort

un nombre $A(x,y)$ (tout dans F_5). La clé de cryptage rendu publique contiendra les 25 $a_{i,j}$.

Mais les $a_{i,j}$ ne sont pas choisis au hasard, car $A(x,y)=h(f(x).g(y))$.

C'est f,g,h qui sont choisis au hasard (avec quelques restrictions : f,g dans I^M et h dans I^C)

Le hasard porte seulement sur les 15 variables des coefficients de f,g,h (et encore, il y a les restrictions). Mais pour cacher cette structure, on publie les 25 $a_{i,j}$ (qui sont liés entre eux).

Exemple :

Si $f(x)=(x^4+3x^2+3)$, $g(y)=2y^3+3y^2+2y+2$ et $h(z)=z^4+3z^3+4z^2+z$

Alors $f(x)g(y)=$
 $1+0x+1x^2+0x^3+2x^4+$
 $1y+0yx+1yx^2+0yx^3+2yx^4+$
 $4y^2+0y^2x+4y^2x^2+0y^2x^3+3y^2x^4+$
 $1y^3+0y^3x+1y^3x^2+0y^3x^3+2y^3x^4+$
 $0y^4+0y^4x+0y^4x^2+0y^4x^3+0y^4x^4$

Mais ces 25 coefficients ne sont pas les $a_{i,j}$, il faut injecter cette grosse somme dans le calcul du polynôme h et c'est seulement là qu'on obtient les $a_{i,j}$. (Je passe les calculs car c'est vraiment un peu long à la main ...)

Rappelons que $x^5 = x$, ce qui permet la stabilité des formes $\sum_{i,j=0..4} a_{i,j} x^i y^j$ quand on les

compose par un polynome.

3-Pyramide

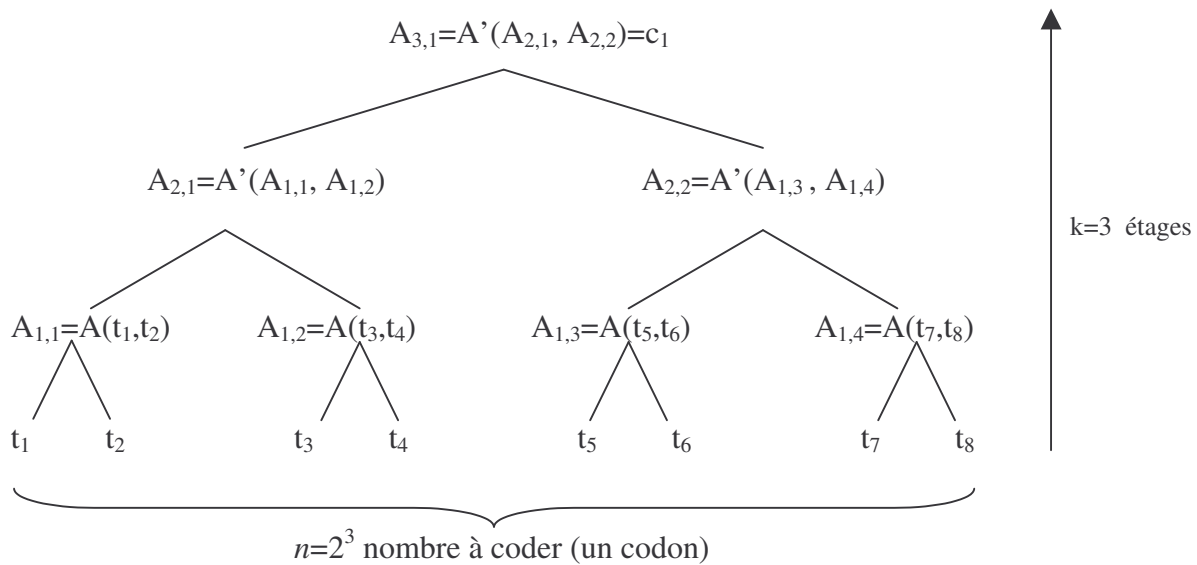
La méthode pyramidale consiste à intégrer toutes les variables dans une somme finale.

Parce qu'on aura besoin de simplifier la fonction h , on crée le schéma de base A' comme pour A , mais avec une particularité : on ajoute h^{-1} autour des variables.

$A'(x,y)=h(f(h^{-1}(x)) \cdot g(h^{-1}(y)))$. Ainsi $A'(A(x,y),A(z,t))=h(f(x)g(y)f(z)g(t))$

Comme pour A , A' sera publié sous forme développée, il possède donc la même apparence (développée) que A .

On construit donc la pyramide ainsi :



Au résultat remarquons que $A_{3,1}$ est un nombre de F_5 qui dépend de toutes les variables : t_1, t_2, \dots, t_8

Cette pyramide est le calcul que doit effectuer l'ordinateur pour crypter une seule valeur dans F_5 . On peut avoir l'impression que le nombre de calcul est énorme, mais en précalculant de nombreuses choses le résultat peut-être très rapide.

(On pourrait se passer de A et se contenter de A' en remplaçant le polynôme f par $f'(x) = f(h^{-1}(x))$)

Intérêt

Lorsqu'on essaie d'exprimer le résultat $A_{3,1}$ à partir des variables (t_i) , on s'aperçoit qu'il est

de la forme $\sum_{\alpha \in \{0..4\}^8} a_\alpha \cdot t_1^{\alpha_1} \cdot t_2^{\alpha_2} \dots t_8^{\alpha_8}$. L'intérêt mathématique de cette forme réside dans le fait

que toute application de $(F_5)^8$ dans F_5 s'écrit par cette forme

(Il y a une bijection entre les applications de $(F_5)^8$ dans F_5 et les coefficients a_α).

Autrement dit on est dans le cas le plus général des cryptages possibles dans F_5 . Il n'est pas possible de faire plus complexe que cette forme qui recouvre toutes les méthodes possibles. Cette remarque me semble très intéressante pour le cryptage.

Bien sûr, les fonctions $A(x,y)$ sont structurées et ne permettent donc pas de couvrir toutes les possibilités de cryptage. Mais vu que tout est donné sous forme développée, ce cadre développé est plus ou moins la cadre logique de résolution. Ce qui rend extrêmement difficile la découverte la structure particulière sous-jacente (pour p un peu plus grand que 5 évidemment). C'est là un fondement de la sécurité de cette méthode :

Pour casser la méthode une des principal méthode consiste à découvrir f, g et h à partir de la forme développée $A(x,y)$. Or si on utilise le cadre général, la forme développée possède p^n termes, ce qui est inaccessible dès que p et n deviennent assez grands.

Il faut donc trouver f, g, h en résolvant $f(x) \cdot g(y) = h^{-1}(A(x,y))$. C'est une équation diophantienne sur F_p de degré 2 avec $3p$ inconnues et $p^2 + p$ termes, ce qui semble plutôt difficile. Et c'est justement parce que les équations diophantiennes sont difficiles à résoudre que ce schéma de

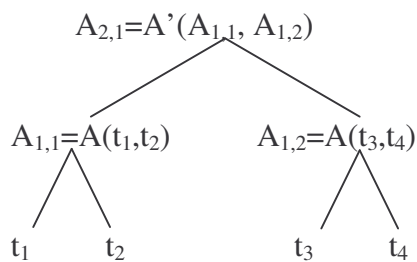
base a été choisi. (Si par hasard on savait résoudre le second degré il est facile de passer au 3^{ème} degré). Casser cette équation diophantienne est inaccessible à la force brute, si p est assez grand. Les tests sont au nombre de $(3p)^p$.

Existe-t-il une autre méthode que la force brute pour ce type d'équation ?

4-Méthode de cryptage

Remarque

Nous l'avons soulevé, mais il faut bien comprendre la particularité du schéma de base. En effet on a : $h^{-1}(h(f.g))=f.g$. Or A' en composant les variables par h^{-1} a été conçu spécialement pour détruire h à chaque étapes. Ainsi dans le schéma suivant



comme $A(x,y)=h(f(x).g(y))$ et $A'(x,y)=h(f(h^{-1}(x)).g(h^{-1}(y)))$, il en résulte que $A_{2,1}=h(f(t_1)g(t_2)f(t_3)g(t_4))$. Autrement dit les variables ne sont pas aussi mélangés qu'elle peuvent paraître par leur forme développée, elles restent sous forme très structurées. Et cela jusqu'en haut de la pyramide. En effet on a $A_{3,1}=h(f(t_1)g(t_2)f(t_3)g(t_4)f(t_5)g(t_6)f(t_7)g(t_8))$ Mais cette structure est cachée et inaccessible. Publiquement il ne paraît que la structure développée. C'est cette structure cachée qui va permettre la fabrication de clés asymétriques.

Pour bien comprendre la méthode pyramidale, il faut toujours avoir à l'esprit la distinction entre la forme structurée et la forme développée (la forme publique). Toutes les explications servant à comprendre le mécanisme sont donnés sous la forme structurée, mais tous les calculs effectués par la machine se font par la formes développées.

1ère étape - le premier cryptage : C(T)

Il faut maintenant construire le procédé asymétrique : pour l'instant nous n'avons crypté qu'une seule valeur dans F_5 . Nous allons tout simplement crypter n valeurs par le même méthode pyramidale. En répétant n fois la fabrication de valeurs $A_{3,1}$ par le procédé de la pyramide, on utilisera des f et g différents (noté f_i, g_i). Ces valeurs final (les $A_{3,1}$) seront notées les c_i

Au total, on obtient un texte crypté, $C(T)=(c_1,c_2,\dots,c_8)$ dans $(F_5)^8$

C'est un élément de $(F_5)^8$, par exemple on pourrait avoir $C(T)=(3,2,3,0,1,0,2,1)$.

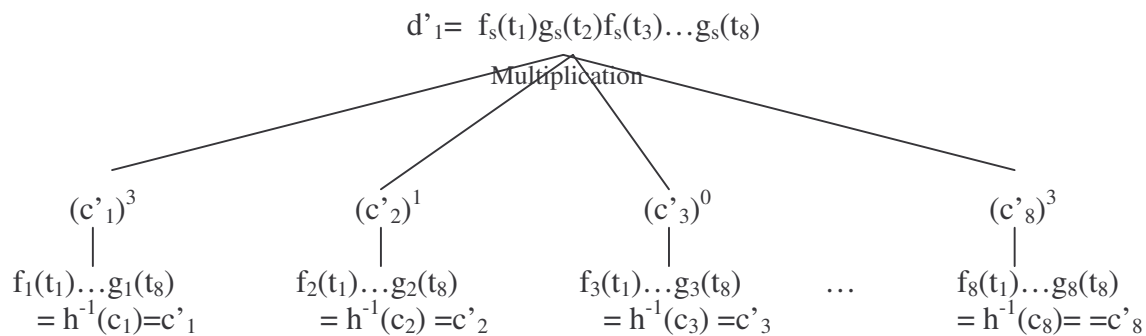
Observons la structure finale des c_i : on remarque que si on passe ces c_i dans h^{-1} , on obtient des expressions de la forme $f(t_1)g(t_2)f(t_3)\dots g(t_8)$ (on note ce nombre c'_i)

La technique consistera à mélanger les valeurs c_i en conservant cette structure stable. (Il existe plusieurs méthodes pour y arriver. Nous présentons ici une méthode non optimisée mais plus accessible)

2ème étape - le second cryptage D(C(T))

Comme les $c'_i=f(t_1)g(t_2)f(t_3)\dots g(t_8)$, on peut les élever à une puissance quelconque puis les multiplier entre eux sans que cela ne change la forme générale. Cela revient à faire une

'combinaison linéaire multiplicative'. Au résultat, on obtient un nombre d'_1 qui est de la même forme structurée que les précédents $f_s(t_1)g_s(t_2)f_s(t_3)...g_s(t_8)$.
Voici schématiquement la production de d'_1 :



En partant des mêmes c_i (c.à.d du texte crypté $C(T)$), on construit n fois de tels nombres d'_i ; mais avec des puissances différentes. (Ces puissances formeront la clé de décryptage.)

Si on compose ces nombres d'_i par h (notons ce nombre d_i , $d_i = h(d'_i)$), on s'aperçoit que d_i possède exactement la même forme que les c_i , une forme du type

$$h(f(t_1)g(t_2)f(t_3)g(t_4)f(t_5)g(t_6)f(t_7)g(t_8))$$

On appelle D cet ensemble de combinaisons linéaires multiplicatives précédées de h^{-1} et suivit de h . On a alors : $R(T) = D(C(T)) = (d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8)$, c'est le deuxième texte crypté que l'on note $R(T)$.

C 'est un élément de $(F_5)^8$, par exemple on pourrait avoir $D(T) = (1, 1, 2, 0, 4, 2, 3, 3)$.

Méthode de Cryptage-décryptage

On obtient alors un résultat tout à fait particulier, il existe deux façons de construire $R(T)$:

- on peut construire $R(T)$ à partir de $C(T)$, en passant les c_i dans h^{-1} , puis dans les combinaisons linéaires multiplicatives qu'on a appelé D . C'est la méthode qu'on vient d'exposer.
- Mais on peut aussi construire $R(T)$ (i.e. les d_i) par le même procédé que l'on a construit $C(T)$, c'est à dire à partir d'une pyramide de fonctions à double entrée du type $A(x,y)$. Cela se fait très simplement car les d_i ont la même forme que les c_i (on remarque que f_s et g_s sont donnés quand on a choisit D . Il n'y a qu'à choisir h et la pyramide est construite, on peut d'ailleurs reprendre le même que pour $C(T)$). Et c'est par cette seconde méthode que l'on va construire $R(T)$ en donnant publiquement les coefficients développés de la pyramide qui le produit.

Au résultat le cryptage se fait par le calcul de deux pyramides sous forme étagée et développée. C'est terminé, il ne reste plus qu'à expliciter le cryptage.

Le cryptage est réalisé par le couple $F(T) = (C(T), T - R(T))$

On note aussi (T', T'') un texte crypté.

Dans les exemples on avait :

$$T = (1, 0, 3, 2, 1, 0, 3, 3), \quad T' = C(T) = (3, 2, 3, 0, 1, 0, 2, 1) \quad \text{et} \quad T'' = R(T) = (1, 1, 2, 0, 4, 2, 3, 3)$$

$$\text{Le cryptage sera donc donné par } F(T) = ((3, 2, 3, 0, 1, 0, 2, 1), (0, 4, 1, 2, 2, 3, 0, 0)) = (T', T'')$$

Et le décryptage se fait tout simplement par le calcul suivant $T'' + D(T')$ où D est l'ensemble de ces combinaisons linéaires précédées de h^{-1} et suivit de h .

Si l'on fait ce calcul sur T' et T'' , on retrouve $T = (1, 0, 3, 2, 1, 0, 3, 3)$.

Commentaire

On peut penser que le nombre de calcul sera très grand pour un p satisfaisant ($p > 16$), mais de nombreuses astuces et choix adaptés permettent de les réduire considérablement. On réalisera notamment des économies en réutilisant plusieurs fois une bonne partie des calculs effectués par exemple en stockant dans des tableaux précalculés à double entrée tous les calculs $A(x,y)$.

Trouver D semble impossible, car la taille des systèmes à manipuler est inaccessible pour p et n suffisant (inversion d'une matrice de $p^n \times p^n$).

5-Retour

Pour écrire le texte codé $F(T)$ dans un fichier sous forme d'octet, il y aura à nouveau un peu de perte. On procède comme tout à l'heure, en regroupant cette fois ci les bits par 8 pour former des octets.

(Notons qu'il faut 3 bits pour coder un élément de F_5 donc cette méthode simple produit une perte de 1 bit pour 2 bit de texte à coder. Sachant de plus qu'on double la place, au total il faut 5 octets pour coder 2 octets. On peut perdre moins de place par une méthode plus fine).