Chiffrement par injection de bruit aléatoire

JAKJOUD Abdeslam

23 mars 2010

Table des matières

1 Contexte du document

Dans le cadre de l'élaboration d'un outil de télé-assistance nous travaillons sur une nouvelle alternative de chiffrement des données transférées à travers Internet. Le résultat de ce travail est la proposition d'un algorithme de chiffrement par insertion de caractères invalides et non significatifs dans le cryptogramme, de façon aléatoire. Ce document décrit l'approche de chiffrement et de déchiffrement en vue d'évaluer la résistance du crypto-système aux attaques.

2 Idée de base

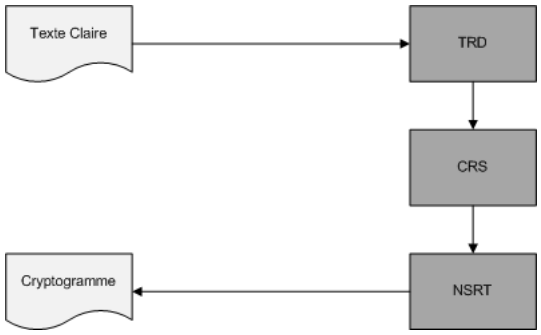
L'idée est de trouver un moyen pour générer des caractères in-intelligibles qui, une fois inversés, n'auront aucune signification car ils ne sont pas le résultat d'un chiffrement.

Ces caractères (bruit) doivent être de valeurs et de positions aléatoires. Ce qui signifi que même la personne qui chiffre le texte d'origine ne sera en mesur de prédire l'emplacement des caractères bruits.

Cela est interessant pour pouvoir amortir certains types d'attaques se basant sur la detection de la longueur de la clé avant des éléments qui la composent, aussi bien que les attaques stochastiques modernes qui prennent de plus en plus de valeur dans le domaine de la cryptanalyse, vu que la présence du bruit permettra de limiter l'efficacité de certaines méthodes qui essayent de vérifier l'influence des modification du texte en claire sur le cryptogramme.

3 Chiffrement

Le chiffrement suivra l'organigramme ci-dessous, l'explication qui suit permettra de mieux comprendre son fonctionnement.

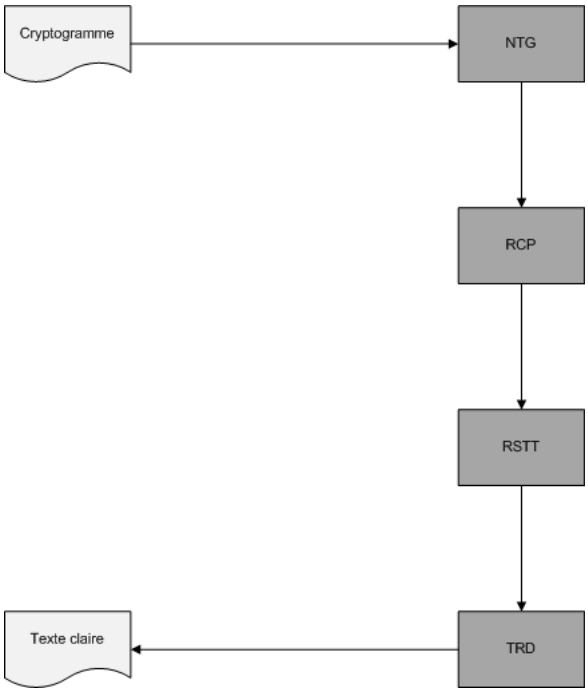


Explication :

Etape	**Description**
TRD	La traduction permet de récuperer le texte du message en claire sous format numérique (par exemple traduir chaque caractère par son code ASCII)
CRS	Le croisement se fait en utilisant la donnée de la première clé K Soit C le vecteur des entiers représentant le texte en clairee et n son cardinal Soient N et R deux vecteurs entiers de la même taille que C tels que : $\forall i \in [1, n] : C_i = K.N_i + R_i$ Après on croise les deux vecteurs pour former le vecteur M dont la taille est : $n' = 2.n$ $M = (N_1, R_n, N_2, R_{n-1}, \ldots, N_n, R_1)$
NSRT	L'insertion est l'étape la plus importante dans l'algorithme Elle concerne l'injection de bruit selon le principe suivant Soient : P le vecteur des nombres premiers qui forment la seconde clé n_p la taille de P f la fréquence d'apparition du bruit et n_b la taille du bruit qui est aussi un paramétre d'entrée Soit S le vecteur du cryptogramme de taille n' : $\forall i \in [1, n'] : S_i = \begin{cases} \rho & \rho \neq 0[P_h] \\ P_h \times M_i \end{cases}$ Notons que ρ est un nombre aléatoire généré à l'emplacement i par une probabilité de $\frac{f}{n_b}$ et que $h = (i + n_p - 1).[n_p] + 1$

4 Déchiffrement

Tout comme le chiffrement, le déchiffrement suivra un organigramme et sera expliqué par la suite.



Explication :

Etape	Description
NTG	Le netoyage permettra de reconnaître le bruit des éléments déchiffrables selon la formule suivante : $S'_i = E[\frac{S_i}{P_h}].E[1 - (\frac{S_i}{P_h} - E[\frac{S_i}{P_h}])]$ Le résultat sera le remplacement de tout le bruit par des zeros
RCP	La réduction consiste à enlever tous les zeros pour reccuperer le vecteur M de longueur n' : $S' = (x_1, 0, 0, x_2, x_3, 0, x_4, \ldots, x_{n'}, 0, 0, 0)$ $\longrightarrow M = (x_1, x_2, x_3, x_4, \ldots, x_{n'})$
RSTT	La restitution permettra d'inverser la division euclidienne : Soient N et R deux vecteurs de taille $n = \frac{n'}{2}$ tels que $N_i = (M_i, \exists k \in \mathbf{N}$ tel que $i = 2.k + 1)$ et $R_i = (M_i, \exists k \in \mathbf{N}$ tel que $i = 2.k)$ $\forall i \in [1, n] : C_i = K.N_i + R_i$
TRD	La traduction conssiste à récupérer les caractères à partir de leurs codes dans C

5 Cryptanalyse

Pour la partie cryptanalyse, les algorithmes classiques comme l'analyse fréquentielle et l'indice de coïncidence sont impuissantes vu que le mécanisme de croisement fait perdre la consistence des caractères et l'injection de bruit permet de donner de fausses pistes.
Aussi l'injection de bruit permet de fausser les attaques stochastiques comme la cryptanalyse linéaire et différentielle. En ce qui concerne l'attaque par force

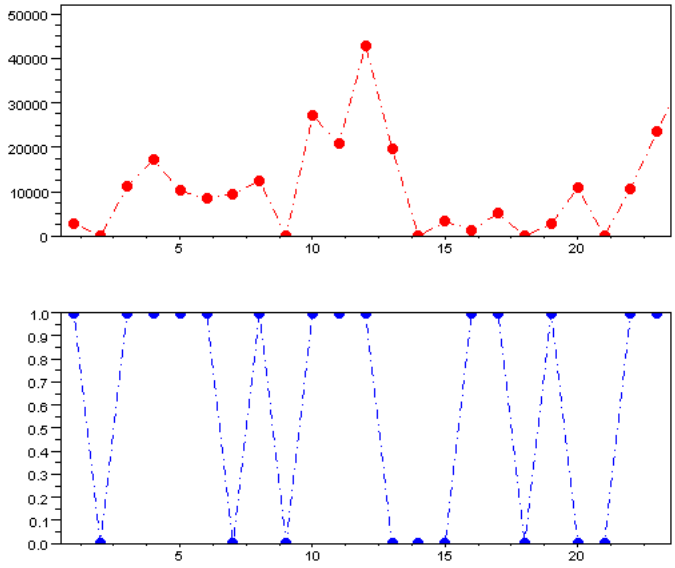
brute en ne connaissant pas la taille de la seconde clé la retrouver relève d'une probabilité de :

$$n^{\frac{n_p.(n_p+1)}{2}}$$

tels que :

- n est le nombre d'entiers que peuvent prendre les valeurs de la seconde clé ;
- n_p est la taille maximale de la seconde clé.

Pour aussi vérifier si le bruit prend des valeurs différentes de celles du cryptogramme une multiplication de ρ par le codage maximal (256 pour l'ASCII) permet de donner des valeurs de grandeur identique que les éléments intelligibles du cryptogramme comme le montre la figure suivante :



Les points en rouge représentent le cryptogramme et ceux en bleu indiquent la présence du bruit