

TD 3 de Cryptologie

Lundi 24 octobre 2005

1 Chiffrement par flot

1.1 Génération

On rappelle qu'un polynôme P est primitif sur un corps K si les puissances de X engendrent $(K[X]/P)^*$.
Par exemple $1 + X + X^2$ est primitif sur \mathbb{F}_2 car il génère

$$o(X) = \{X, X^2 = X + 1, X^3 = X^2 + X = 1\} = \mathbb{F}_2[X]/(X^2 + X + 1).$$

1. Montrer que le polynôme irréductible $X^6 + X^3 + 1$ n'est pas primitif sur \mathbb{F}_2 .
2. Montrer que le polynôme $X^5 + X^3 + X^2 + X + 1$ est primitif sur \mathbb{F}_2 . Que dire des autres polynômes irréductibles de degré 5 sur \mathbb{F}_2 ?

1.2 Synchronisation

1. Montrer la re-synchronisation après t caractères transmis suite à une erreur pour le chiffrement par flot auto-synchronisant vu en cours :

$$\begin{aligned}\sigma_{i+1} &= (c_{i-t}, c_{i-t+1}, \dots, c_{i-1}) \\ z_i &= g(\sigma_i, k) \\ c_i &= h(z_i, m_i)\end{aligned}$$

où l'état initial (public) est $\sigma_0 = (c_{-t}, c_{-t+1}, \dots, c_{-1})$, k est la clé, g la fonction produisant le flot de chiffrement z_i , et h la fonction de sortie.

2. On considère le chiffrement par flot suivant :

$$\begin{aligned}\sigma_{i+1} &= (m_{i-t}, m_{i-t+1}, \dots, m_{i-1}) \\ z_i &= g(\sigma_i, k) \\ c_i &= h(z_i, m_i)\end{aligned}$$

Est-il resynchronisant ?

2 Sécurité inconditionnelle

1. Rappeler le principe du chiffrement *one time pad* (OTP), et les conditions sous lesquelles il est jugé sûr.
2. Montrer que le chiffrement OTP est inconditionnellement sûr, c'est-à-dire que l'interception d'un chiffré ne fournit à l'adversaire aucune information concernant le clair.

3 Chiffrements historiques : Vigenère

On considère pour $n \in \mathbb{N}^*$, un alphabet \mathcal{A} de taille finie m noté $\mathcal{A} = [0, 1, \dots, m-1]$ et un mot k de n lettres sur \mathcal{A} le chiffrement suivant :

$$E : \mathcal{A}^* \rightarrow \mathcal{A}^*$$

$$x_0 x_1 x_2 \dots x_u \mapsto (x_0 + k_0 \bmod m) \dots (x_{n-1} + k_{n-1} \bmod m) (x_n + k_0 \bmod m) \dots (x_u + k_{u \bmod n} \bmod m)$$

Exemple :

Clair	C	E	T	D	E	S	T	M	E	R	V	E	I	L	L	E	U	X
Clef	L	A	U	R	E	N	T	L	A	U	R	E	N	T	L	A	U	R
Chiffré	N	D	N	U	I	F	M	X	E	L	M	I	V	E	W	E	O	O

On déchiffre avec la clef "PAGJWNH".

1. Pour une clef de chiffrement comment calculer la clef de déchiffrement ?
2. Pour une clef de taille $m = 1$ ce procédé s'appelle le chiffre de César. Quelle est la difficulté de le casser ?

On suppose qu'on sait que le texte clair est écrit en langue anglaise, et que la distribution des lettres dans le texte en clair est indépendant de la position considérée. On donne la distribution de fréquence observée des lettres en langue anglaise dans la figure 1.

3. Quelle méthode de cryptanalyse du chiffre de César vous inspire la figure 1 ?
4. Si on connaît la taille de la clef de chiffrement de Vigenère, que devient la difficulté de casser le chiffrement sous ces hypothèses ?

Pour deux textes T_1 et T_2 de même longueur on mesure $\kappa(T_1, T_2)$ la fréquence du nombre de coïncidences, c'est-à-dire du nombre de positions dans le texte où la lettre est la même dans les deux textes.

5. Calculer $\kappa(T, T')$ où T et T' sont deux textes aléatoires de longueur m sur un alphabet de taille n .
6. Pour une langue dont les fréquences d'apparitions des lettres p_1, p_2, \dots, p_n sont données, calculer $\kappa_{\text{theorique}}$ de deux textes quelconques écrits dans cette langue.
7. Montrer que pour deux textes T et T' le chiffrement par un chiffre de César avec la même clef ne modifie pas la valeur de κ .
8. En déduire une méthode d'estimation de la longueur de la clef d'un chiffrement de Vigenère à partir d'un texte chiffré suffisamment long.

On donne pour référence $\kappa_{\text{anglais}} = 6.61\%$ et $\kappa_{\text{français}} = 7.78\%$.

Figure 1: Fréquence des lettres en langue anglaise.