CRYPTOGRAPHIE GENERALE

ANNEXES

Pierre-Louis BAYLE - Benjamin BILLET

Démonstration de la perfection du chiffrement à clé jetable (chiffrement de Vernam)

Imaginons que l'on veuille chiffrer un mot de n lettres avec le chiffrement de Vernam.
On a, si l'on considère que le mot ne contient que les lettres de A à Z :

$$M^* = C^* = K^*$$
$$\left|M^*\right| = \left|C^*\right| = \left|K^*\right| = 26^n$$

Toutes les clés étant, en théorie, équiprobables, la probabilité qu'une clé $k \in K^*$ est de

$$P(K = k) = \frac{1}{26^n}$$

On pose la probabilité conditionnelle d'apparition d'un code *c* en fonction d'un message *m* :

$$P(M = m \mid C = c) = \frac{P(M = m \cap C = c)}{P(C = c)}$$

Les évènements $M = m$ et $C = c$ sont totalement indépendants car *k* est rigoureusement aléatoire et $c = m \oplus k$.
On a donc :

$$\frac{P(M = m \cap C = c)}{P(C = c)} = \frac{P(M = m)P(C = c)}{P(C = c)} = P(M = m)$$

D'où $P(M = m \mid C = c) = P(M = m)$ et donc $H(M \mid C) = H(M)$.

En conclusion, le chiffrement de Vernam est donc bien un chiffrement parfait.

Modes de chiffrement

Dans le concept de chiffrement par bloc (DES, AES, etc…), une suite de n bits (ou bloc), peut être chiffrée de plusieurs façons différentes sans nécessiter l'injection de nouveaux paramètres autre que le bloc et la clé de chiffrement. Ces différentes méthodes d'encodage d'un bloc sont appelés modes de chiffrement.

Mode ECB (Electronic Code Book)

Chaque bloc m_i d'un message *M* est crypté par une fonction (paramétrée avec une clé *k)*, notée E_k telle que $c_i = E_k(m_i)$.

Dans ce mode, qui peut être associé à un chiffrement mono-alphabétique, la sécurité est nulle car $\forall m_i, m_k \in M$, si $m_i = m_k$ alors $c_i = c_k$. Il n'est donc, en principe, jamais exploité en cryptographie.

Mode CBC (Cipher Bloc Chaining)

Ce mode a été crée pour pallier au problème de mono-alphabétisme du mode ECB.
La différence consiste en la réinjection du résultat du précédent chiffrement dans l'encryptage du nouveau bloc.
Ainsi $c_i = E_k(m_i \oplus c_{i-1})$ avec c_0 défini par le système (aléatoirement par exemple).

Poly-alphabétique par nature et suffisamment simple pour être intégré dans des puces spécifiques, ce mode est actuellement le plus utilisé en cryptographie.
Toutefois, le décodage nécessite la connaissance d'une fonction de décodage, inverse de la fonction d'encodage et notée D_k telle que $m_i = c_{i-1} \oplus D_k(c_i)$.

Mode CFB (Cipher FeedBack)

Ce mode a été conçu pour permettre de se passer de la fonction de décodage D_k nécessaire pour le mode CBC.

Très proche, l'encodage s'effectue de cette façon : $c_i = m_i \oplus E_k(c_{i-1})$. Ainsi, $m_i = c_i \oplus E_k(c_{i-1})$, permettant de ne pas implémenter la fonction D_k.
De par le fait, ce mode est moins sur que le CBC, mais en devient plus simple à mettre en œuvre. Pour cela, il est, entre autres, utilisé pour les cryptages réseaux.

Mode OFB (Output FeedBack)

Dans ce mode, on fait intervenir un élément supplémentaire, noté *a*, interne au système qui sera réintroduit à la place du précédent bloc encodé.

$$\begin{cases} a_i = E_k(a_{i-1}) \text{ avec } a_0 \text{ défini par le système.} \\ c_i = m_i \oplus a_i \end{cases}$$

Dès lors, le mode est totalement symétrique car il suffit, pour décoder, d'effectuer $m_i = c_i \oplus a_i$ avec a_i défini de la même façon à l'encodage.
L'avantage majeur de ce mode se situe dans le fait que, lors d'une transmission de données, si un bloc c_i est erroné cela ne perturbera pas le déchiffrement du bloc c_{i+1}. En effet, en mode CBC et CFB, si une erreur de transmission survient, elle entraînera exactement deux erreurs de déchiffrement.

Mode CTR (Counter-mode encryption)

Ce mode est extrêmement proche du précédent. Le changement réside dans le fait que a est remplacé par un compteur incrémental T indépendant tel que $c_i = m_i \oplus E_k(T + i)$.

Ainsi, chaque encodage de bloc est indépendant (à l'image du mode ECB), à la différence qu'un même bloc n'est, en principe[1], jamais codé de la même façon.
Tout comme OFB, le chiffrement est totalement symétrique (sous réserve que les compteurs soient identiques de chaque côté), le déchiffrement s'opérant ainsi : $m_i = c_i \oplus E_k(T + i)$.
Enfin, l'utilisation d'un compteur dont les itérations sont connues à l'avance permet de paralléliser l'encodage et le décodage (encoder ou décoder plusieurs blocs en même temps). De ce fait, ce mode est particulièrement adapté pour les architectures multiprocesseurs ou pour les systèmes de transmission à canaux multiples.

[1] En effet, comme le compteur T est toujours incrémenté de la même façon, un comportement récurrent peut apparaître si la fonction d'encodage n'a pas été correctement conçue (ou insuffisamment testée).

DES - Fonctions de substitutions S2 à S8

S_2

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5
1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11
2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2
3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14

S_3

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2
1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15
2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14
3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2

S_4

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4
1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14
2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8
3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2

S_5

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14
1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8
2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0
3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5

S_6

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5
1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3
2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11
3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8

S_7

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6
1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8
2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9
3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3

S_8															
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12
2	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9
3	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5
4	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6

Algorithme d'Euclide étendu – Exemple

Cherchons le pgcd et les nombres de Bezout pour a = 3 et b = 5.

1: a = 3, b = 5
q = 0, r = 3

2: a = 5, b = 3
q = 1, r = 2

3: a = 3, b = 2
q = 1, r = 1

4: a = 2, b = 1
q = 2, r = 0
u = 0, v = 1

d = 1
u = 1, v = -1

d = 1
u = -1, v = 2

d = 1
u = 2
v = -1

Au final nous obtenons : $2a - b = d$ où a = 3, b = 5 et d = 1.

Chiffrement RSA – Exemple

Pour l'exemple nous prendrons deux nombres premiers simples : p = 47 et q = 59

- $n = 47 \times 59 = 2773$ et $\varphi(n) = (47-1) \times (59-1) = 2668$
- Nous prendrons e = 17, premier avec $\varphi(n)$.
- Nous calculons l'inverse de e grâce à l'algorithme d'Euclide étendu avec en paramètre (e, $\varphi(n)$). Le *u* ainsi obtenu (157 dans notre cas) est l'inverse de e, noté d.
- Nous obtenons donc la clé publique (2773, 17) et la clé privée (2773, 157)

Observons le chiffrement et le déchiffrement en pratique :
Pour l'exemple, la lettre A, codée en ASCII par le code 0100 0001 soit 65 en décimal.
$65 \in [\![0,...,2772]\!]$ donc est chiffrable par l'algorithme.

Dès lors :

$$\begin{cases} 65^{17} \pmod{2773} = 332 \\ 332^{157} \pmod{2773} = 65 \end{cases}$$

Chiffrement El Gamal – Exemple

Pour l'exemple nous prendrons deux un petit nombre premier : p = 1009
D'après le mémoire de Korkine, nous apprenons que 11 est un générateur de $\mathbb{Z}/1009\mathbb{Z}^*$.

- p = 1009 et g = 11
- Nous choisissons s aléatoirement entre 1 et p - 1. s = 984 et β = 433.
- Nous obtenons donc la clé publique (1009, 11, 433) et la clé privée (984)

Observons le chiffrement et le déchiffrement en pratique :
Pour l'exemple, la lettre A, codée en ASCII par le code 0100 0001 soit 65 en décimal.
$65 \in \mathbb{Z}/1009\mathbb{Z}^*$ donc A est chiffrable par l'algorithme.

Dès lors $C = (y_1, y_2)$ avec $\begin{vmatrix} y_1 = 487 \\ y_2 = 667 \end{vmatrix}$

Pour déchiffrer, il suffit d'appliquer $M = 667 \times 487^{24} \pmod{n} = 65$.