CHAPITRE 3.

PROTOCOLES DE CRYPTOGRAPHIE À CLÉ SECRÈTE

http://math.univ-lyon1.fr/~roblot/masterpro.html

Propriétés

Clés. La clé de cryptage et la clé de décryptage sont les mêmes et donc doivent être gardées secrètes.

Transformations. Transformations similaires pour codage et décodage (protocoles symétriques).

Avantage. Algorithmes en général très rapides.

Inconvénient. Il faut pouvoir échanger la clé !

Nombres binaires et hexadécimaux

Concaténation. Pour $w_1, w_2 \in \{0,1\}^l$, on définit $w_1 \cdot w_2 \in \{0,1\}^{2l}$

Découpage. Pour $w \in \{0,1\}^{2l}$, on définit $w = (w_1|w_2)$ avec $w_1, w_2 \in \{0,1\}^l$

Ecriture binaire. Tout entier $N \geq 1$ s'écrit de manière unique

$$N = a_k\, 2^k + a_{k-1}\, 2^{k-1} + \cdots + a_1\, 2 + a_0 = (a_k a_{k-1} \cdots a_1 a_0)_2$$

avec $a_0, a_1, \ldots, a_k \in \{0,1\}$ et $k = \lfloor \log_2(N) \rfloor$.

On identifie par la suite les deux ensembles

$$\{0,1\}^l \leftrightarrow \{n \in \mathbb{N} \text{ avec } 0 \leq n < 2^l\}.$$

Exercice. Convertir 453 et 2034 en binaire. Combien vaut 101101_2 ?

Nombres binaires et hexadécimaux

Nombres hexadécimaux. Les chiffres vont de 0 à F et permettent de travailler en base 16 :

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Un chiffre hexadécimal représente 4 bits (car $2^4 = 16$), donc, par exemple, il faut 32 chiffres hexadécimaux pour représenter 128 bits.

0	1	2	3	4	5	6	7
0000	0001	0010	0011	0100	0101	0110	0111

8	9	A	B	C	D	E	F
1000	1001	1010	1011	1100	1101	1110	1111

Exercice. Convertir 6453 en hexadécimal, puis en binaire.

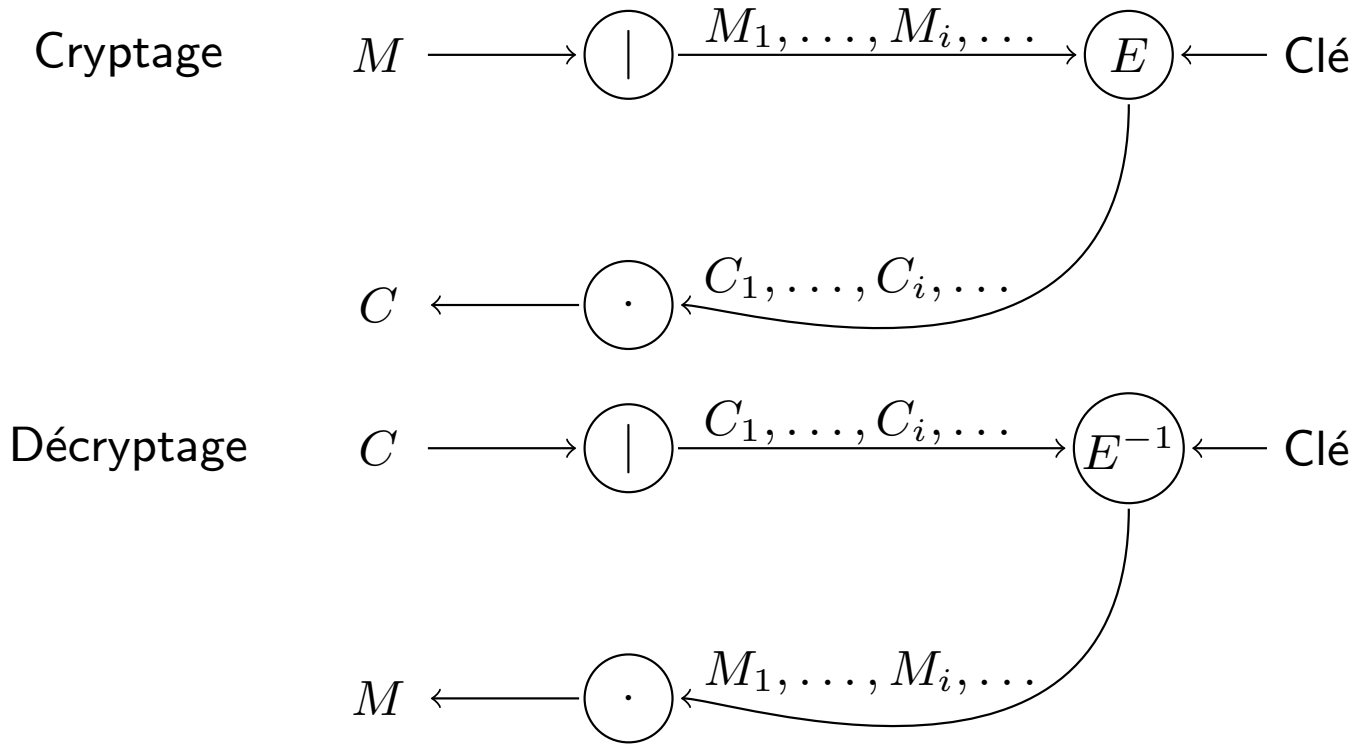
Codage par blocs

Principe. Le message est coupé en blocs de même taille (64 bits, 128 bits, 196 bits, 256 bits...) qui sont encryptés et combinés suivant plusieurs modes d'opération.

4 Modes d'opération.

- ECB (Electronic CodeBook),
- CBC (Cipher-Block Chaining),
- CFB (Cipher Feedback),
- OFB (Output Feedback)

Codage par blocs : mode ECB (Electronic CodeBook)

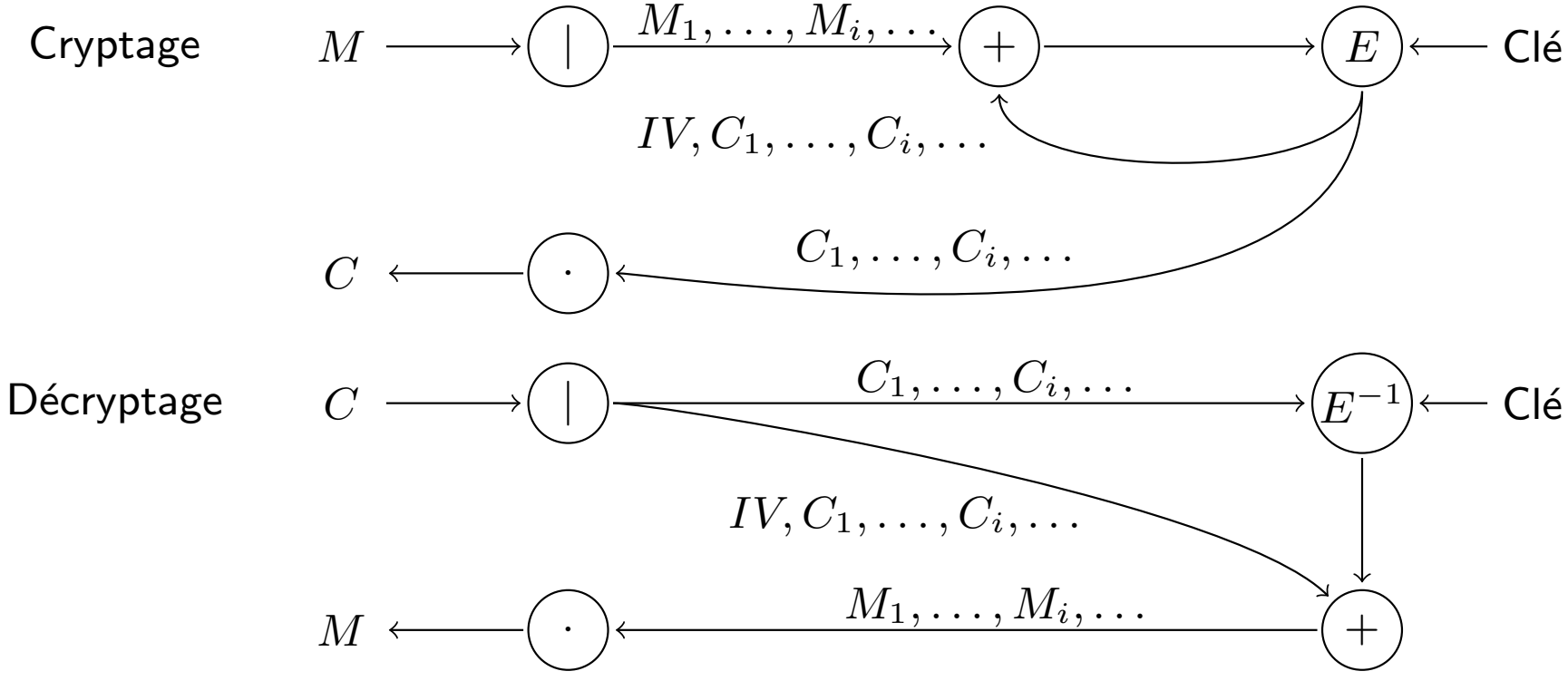


Avantages. Simplicité, intégrité des blocs indépendante.

Inconvénients. Expose le format des données, “usure” de la clé (cf. attaque par le paradoxe des anniversaires).

Exercice. Donner l'expression des blocs C_i.

Codage par blocs : mode CBC (Cipher-Block Chaining)



Remarque. IV vecteur d'initialisation (public)

Avantage. Plus sûr et presque aussi rapide que EBC.

Exercice. Donner l'expression des blocs C_i. En déduire que si C_i est corrompu, les C_j, pour $j > i+1$, peuvent être décodés mais pas C_{i+1}.

D.E.S. (Data Encryption Standard)

Histoire. Créé par IBM. Standard du NIST (National Institute of Standards and Technology) depuis 1976 : norme FIPS 46-3.

Bloc. Codage par blocs de 64 bits.

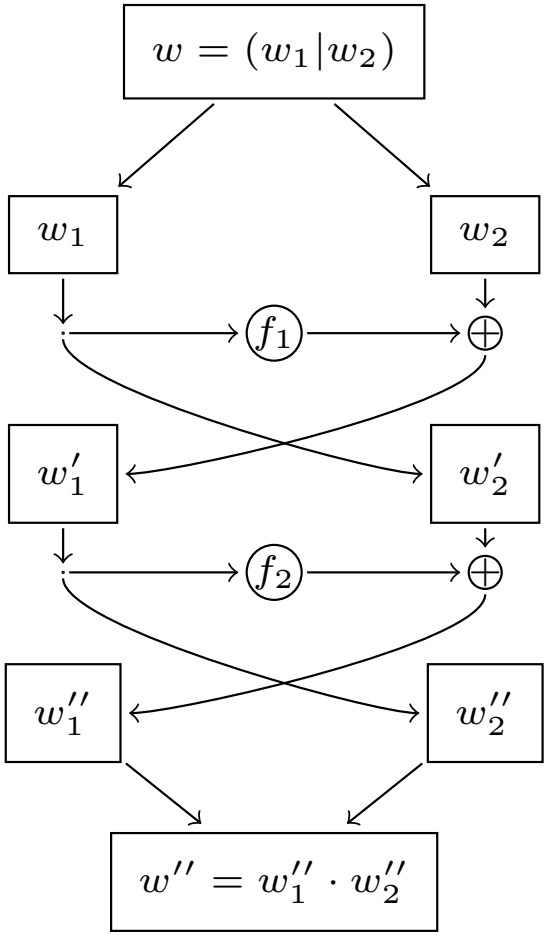
Clé. 64 bits avec 8 bits de parité chaque 8 bits donc 56 bits effectifs (recherche exhaustive divisée par 256).

Spécificités : Plusieurs applications augmentent la sûreté ; vulnérable aux attaques par relations linéaires ; considéré comme obsolète et non sûr car l'espace des clés est trop petit ; repose sur les diagrammes de Feistel.

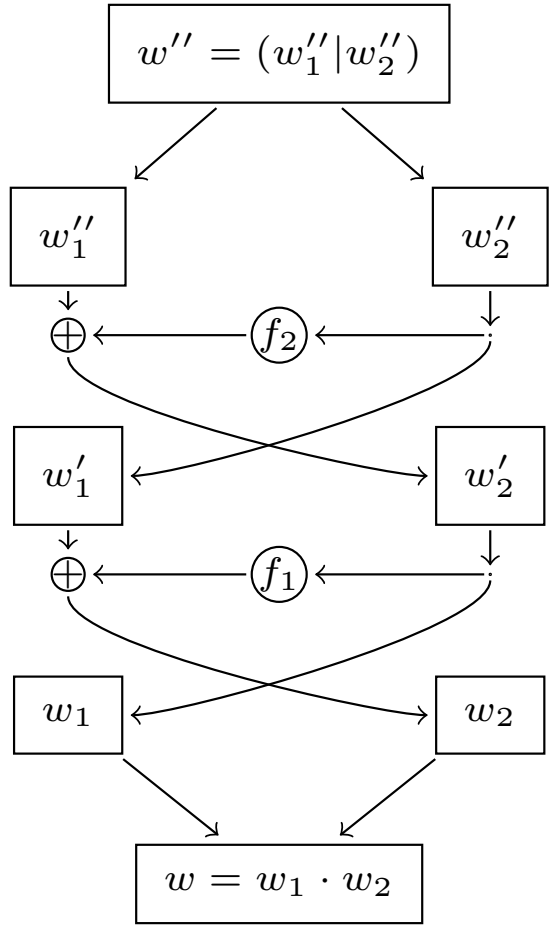
Message à crypter : $w \in \{0,1\}^{2l}$

Clé : deux fonctions $f_1 : \{0,1\}^l \to \{0,1\}^l$ et $f_2 : \{0,1\}^l \to \{0,1\}^l$ (2 rondes)

Cryptage



Décrypage



Exercice sur les diagrammes de Feistel

$w = 101110 \in \{0,1\}^6$

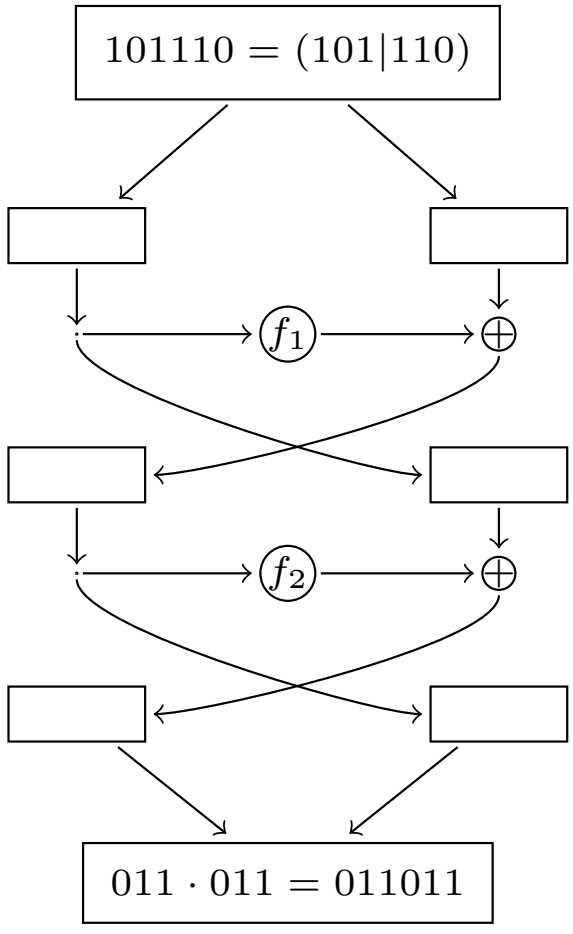
$f_1 : \{0,1\}^3 \to \{0,1\}^3$

$000 \to 101$
$001 \to 100$
$010 \to 111$
$100 \to 000$
$011 \to 001$
$101 \to 101$
$110 \to 010$
$111 \to 110$

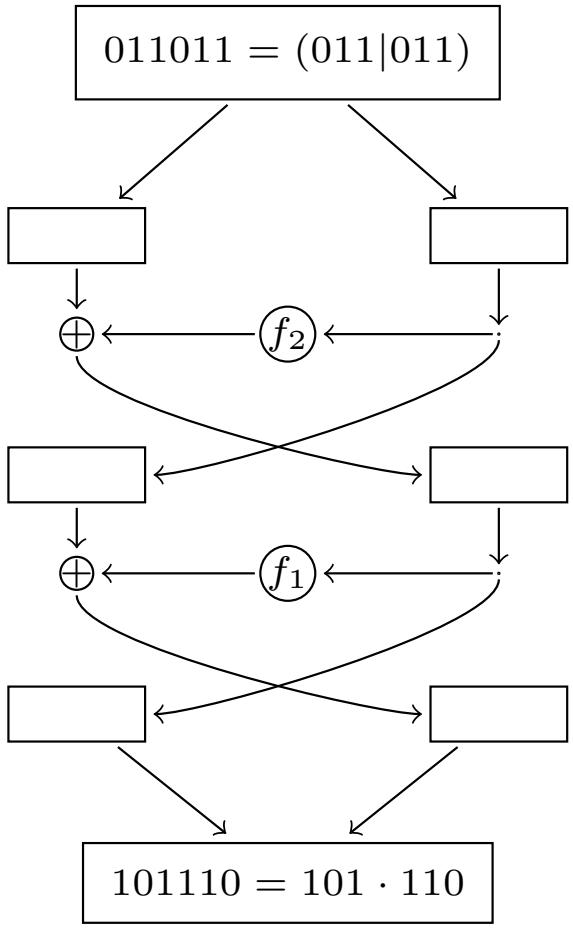
$f_2 : \{0,1\}^3 \to \{0,1\}^3$

$000 \to 010$
$001 \to 001$
$010 \to 110$
$100 \to 111$
$011 \to 110$
$101 \to 011$
$110 \to 001$
$111 \to 100$

Cryptage



Décrypage



Exercice sur les diagrammes de Feistel

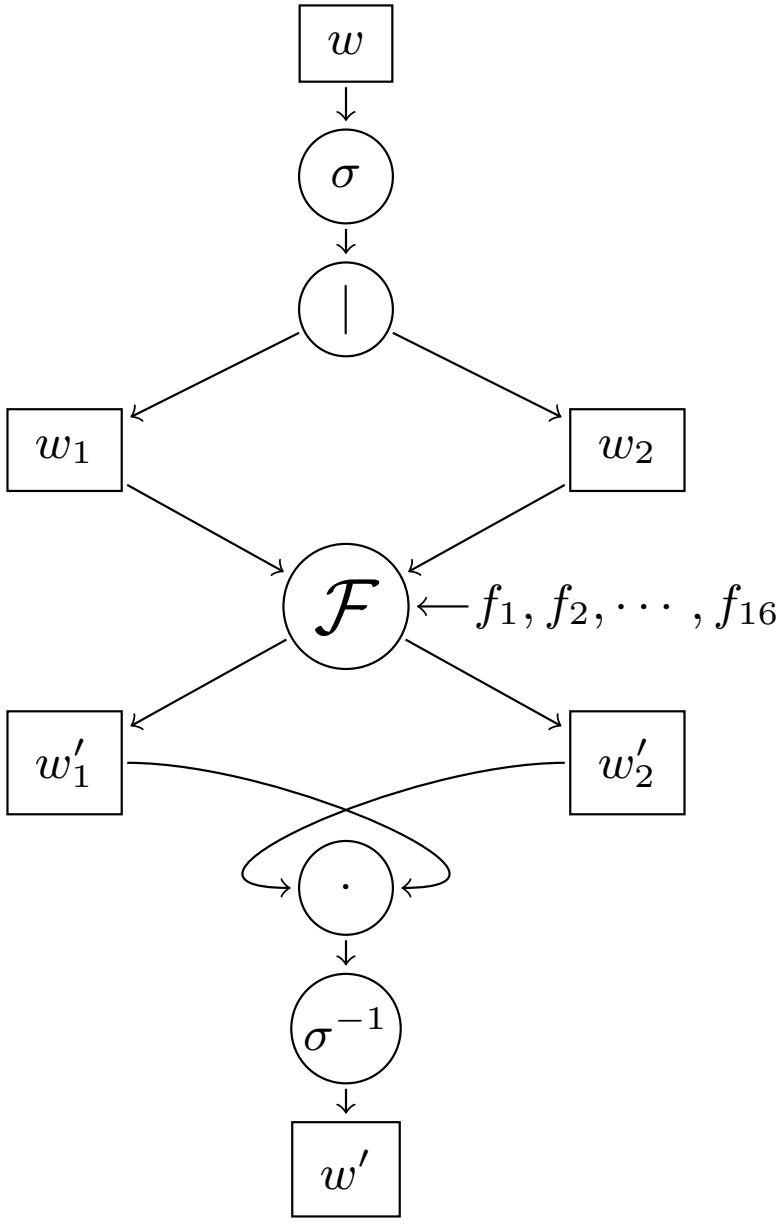
Soit un diagramme de Feistel avec comme entrée $w_1 \cdot w_2$ et sortie $w'_1 \cdot w'_2$. Montrer que

$$w'_1 = f_2(f_1(w_1) \oplus w_2) \oplus w_1 \text{ et } w'_2 = f_1(w_1) \oplus w_2$$

En déduire l'inversibilité des diagrammes de Feistel.

Déterminer les expressions des mots de sorties en fonction des mots d'entrée des diagrammes de Feistel où les fonctions f_1 et f_2 vérifient les propriétés suivantes :

1. f_1 et f_2 sont la fonction nulle.
2. f_1 et f_2 sont la fonction de négation binaire.
3. f_1 et f_2 sont des fonctions linéaires.



CODAGE D.E.S.

1ère étape. Expansion de la clé :

16 fonctions $f_1, \ldots, f_{16}$ sont construites

à partir de la clé par des permutations

sur 4 bits (S-boîtes) et des sous-clés

2ème étape. Chiffrement : application

du diagramme de gauche avec

w un bloc de 64 bits

σ une permutation sur 64 bits *fixée*

Expansion de la clé dans D.E.S.

Les sous-clés K_i de D.E.S. sont au nombre de 16 et de taille 48 bits.

On utilise deux fonctions :

$$\text{PC1} : \{0,1\}^{64} \to \{0,1\}^{28} \times \{0,1\}^{28}$$
$$\text{PC2} : \{0,1\}^{28} \times \{0,1\}^{28} \to \{0,1\}^{48}$$

Puis la procédure suivante :

1. Faire $(C_0, D_0) \leftarrow \text{PC1}(K)$
2. Pour $i = 1, \ldots, 16$, faire
   1. C_i est le mot obtenu à partir de C_{i-1} par une permutation circulaire à gauche de 1 bit si $i = 1, 2, 9$ ou 16, ou 2 bits sinon.
   2. D_i est le mot obtenu à partir de D_{i-1} par une permutation circulaire à gauche de 1 bit si $i = 1, 2, 9$ ou 16, ou 2 bits sinon.
   3. Faire $K_i \leftarrow \text{PC2}(C_i, D_i)$.

Construction des fonctions f_i

Chaque fonction $f_i : \{0,1\}^{32} \to \{0,1\}^{32}$ est construite comme suit

1. On applique $E : \{0,1\}^{32} \to \{0,1\}^{48}$
2. On ajoute (ou exclusif) la sous-clé K_i
3. On coupe le résultat en 8 mots $w_1, \ldots, w_8$ de 6 bits
4. On applique $S_i : \{0,1\}^6 \to \{0,1\}^4$ à chaque w_i $(i = 1, \ldots, 6)$
5. On concatène les mots obtenus en un mot de 32 bits
6. On applique une S-boîte finale

E est construit en permutant les bits de l'entrée avec 16 bits répétés. On part du 32ème bit, on prend 6 bits (avec permutation circulaire), puis on repart de 1 bit en arrière et on recommence :

$$E(x_1 x_2 \cdots x_{32}) = x_{32} x_1 x_2 x_3 x_4 x_5 x_4 x_5 \ldots$$

Exercice sur les clés faibles de D.E.S.

PC1

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

PC2

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

1. Supposons que les sous-clés K_i sont toutes égales.
   Montrer que les bits de C_0 sont tous égaux, ainsi que ceux de D_0.
2. En déduire qu'il existe exactement 4 clés D.E.S. pour lesquelles toutes les sous-clés sont les mêmes.
3. Déterminer ces 4 clés faibles.

Résultats de cryptanalyse contre D.E.S.

Recherche exhaustive. On essaie toutes les clés possibles : 2^{56}, soit 2.3 années/machine pour un test de $1\,000\,000\,000$ clés par seconde. En 1998, le **EFF** (Electronic Frontier Foundation) a construit pour environ \$250,000 une machine spécialisée qui casse une clé D.E.S. en une soixantaine d'heures.

Attaques différentielles. D.E.S. a (sans doute) été conçu pour résister aux attaques différentielles. Néanmoins, on peut casser une clé D.E.S. avec une attaque différentielle avec 2^{46} textes clairs.

Attaques linéaires. D.E.S. n'a pas été conçu pour résister aux attaques linéaires (qui ont été inventées après D.E.S.). Les meilleures attaques permettent de retrouver une clé D.E.S. avec 2^{43} textes clairs.

Triple D.E.S.. Le protocole Triple D.E.S. qui consiste à appliquer trois fois le protocole D.E.S. avec trois clés différentes est considéré comme étant plus sûr puisque l'espace des clés est beaucoup plus grand. Cependant D.E.S. est désormais obsolète avec l'arrivée de A.E.S.

I.D.E.A. (International Data Encryption Algorithm)

Histoire. Développé par Lai et Massey en 1992

Bloc. Codage par blocs de 64 bits

Clé. Clé de 128 bits

Spécificités. Utilise trois structures différentes pour une plus grande sécurité ; pas d'attaque efficace connue autre que la recherche exhaustive (mais certaines clés présentent des faiblesses). Inconvénients : bloc trop petit par rapport à la longueur de la clé ; problèmes de brevet.

Opérations de base de I.D.E.A. (mots de 16 bits)

Ou exclusif : $a \oplus b$

$$0100010101001011 \oplus 0111000011010110 = 0011010110011101$$

Addition modifiée : $a \boxplus b := (a + b) \mod 2^{16}$

Exercice. Montrer que

$$0100010101001011 \boxplus 0111000011010110 = 1011011000100001$$

Multiplication modifiée : $a \otimes b := a \times b \mod (2^{16} + 1)$
(note : $2^{16} + 1$ est premier et $0 \leftrightarrow 2^{16}$)

Exercice. Montrer que

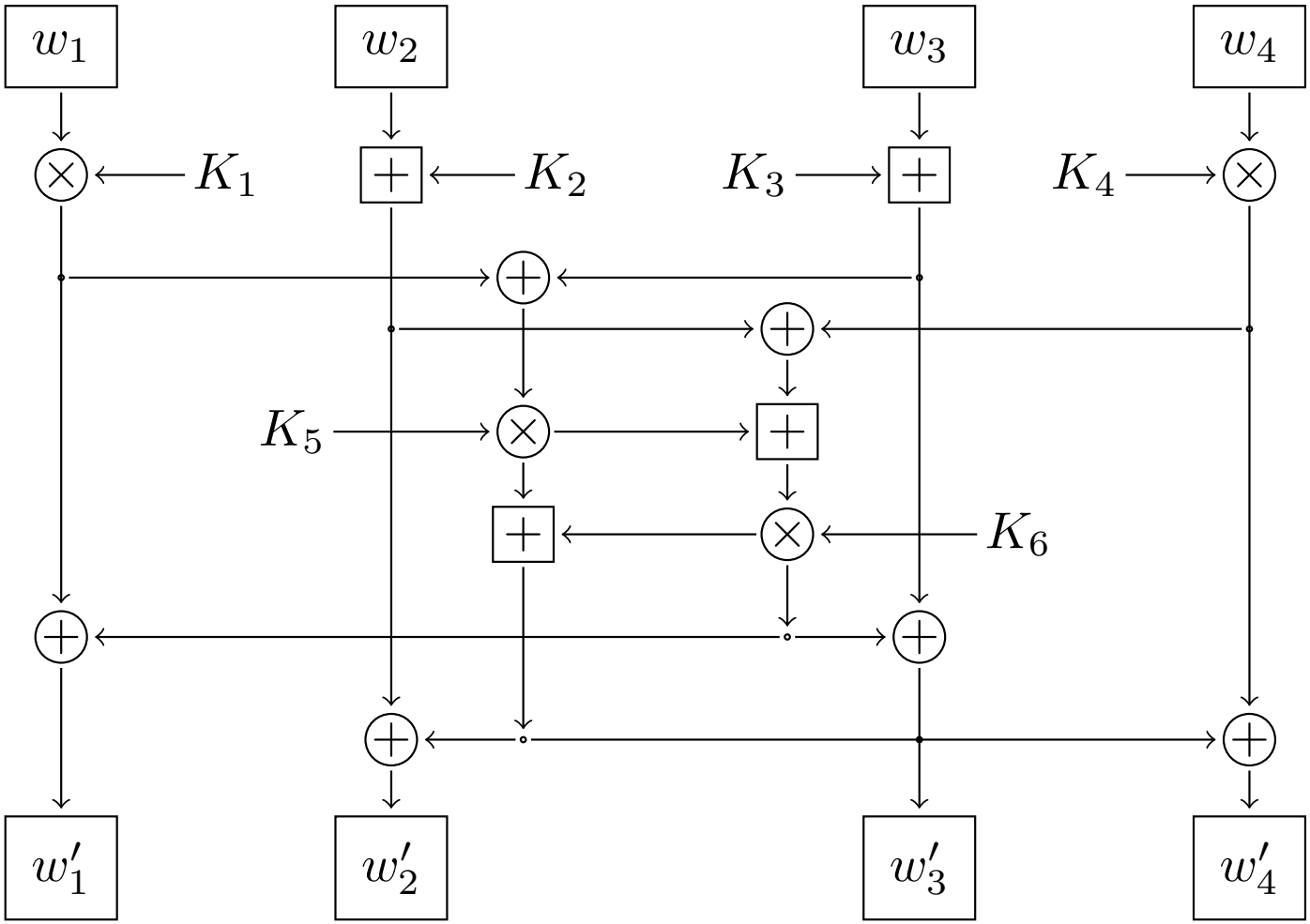
$$0100010101001011 \otimes 0111000011010110 = 1001111000101000$$

Exercice sur les opérations de bases de I.D.E.A.

On travaille sur des mots de 2 bits (au lieu de 16 bits).

1. Donner la description des trois opérations de l'I.D.E.A. pour les mots de 2 bits et un exemple de calcul.
2. Donner les représentations logiques des FBV correspondants aux trois opérations.
3. Calculer les transformées de Fourier et Walsh de ces FBV.
4. Donner les représentations algébriques des FBV correspondants aux trois opérations.
5. En déduire le degré de ces opérations.

Diagramme de base pour I.D.E.A. (une ronde)

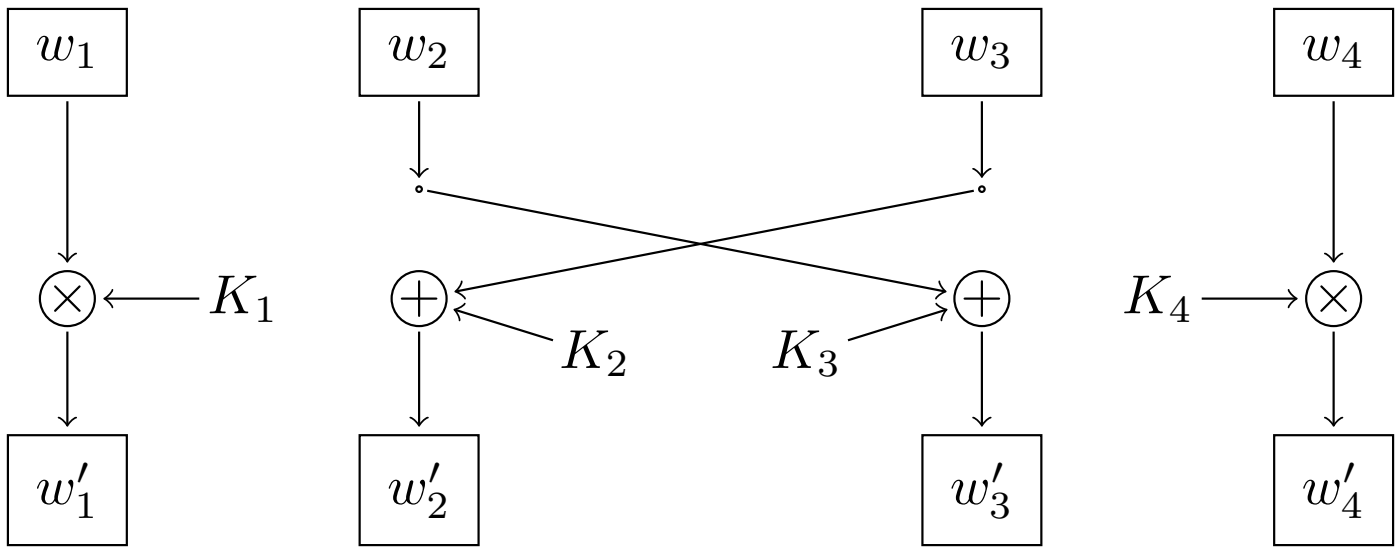


$K_1, K_2, K_3, K_4, K_5, K_6$ sont des sous-clés

Exercice. Donner l'expression des w'_i en fonctions des w_i et des K_i

Les rondes dans I.D.E.A.

Rondes. I.D.E.A. consiste à l'application de 8 rondes successives, nécessitant chacune 6 sous-clés, et d'une dernière demi-ronde comme suit



Il faut donc engendrer $8 \times 6 + 4 = 52$ sous-clés de 16 bits.

Expansion de la clé dans D.E.S.

Sous-clés. On coupe la clé (de 128 bits) en 8 sous-clés de 16 bits. Puis, on décale la clé de 25 bits sur la gauche (de manière circulaire) et découpe le mot obtenu en 8 sous-clés de 16 bits. En faisant cela 7 fois, on obtient 56 sous-clés (les 4 dernières ne sont pas utilisées).

Exercice.

1. Montrer que les sous-clés obtenues sont toutes différentes.
2. Étudier la position d'un même bit dans les différentes sous-clés.
3. Que se passe-t-il si on répète l'opération une 8ième fois ?

Résultats de cryptanalyse contre I.D.E.A.

En 1996, I.D.E.A. était considéré par certains experts comme le meilleur et le plus sûr des protocoles de cryptographie à clé secrète.

Attaques récentes. Les attaques les plus récentes peuvent casser un I.D.E.A. limité à 5 rondes, alors que I.D.E.A. utilise normalement 8 rondes et demi.

Inconvénients. I.D.E.A. souffre d'un manque d'efficacité fasse à des protocoles plus modernes et aussi son utilisation est limitée par de nombreux brevets. La méthode d'expansion des sous-clés est trop simple et donc il existe un nombre anormalement élevé de clés faibles ou semi-faibles.