Cryptographie - Introduction à DES

Introduction à DES

DES, le chiffrement à clé secrète

Le 15 mai 1973 le **NBS** (*National Bureau of Standards*, aujourd'hui appelé *NIST - National Institute of Standards and Technology*) a lancé un appel dans le *Federal Register* (l'équivalent aux Etats-Unix du *Journal Officiel* en France) pour la création d'un algorithme de chiffrement répondant aux critères suivants :

- posséder un haut niveau de sécurité lié à une clé de petite taille servant au chiffrement et au déchiffrement
- être compréhensible
- ne pas dépendre de la confidentialité de l'algorithme
- être adaptable et économique
- être efficace et exportable

Fin 1974, IBM propose « Lucifer », qui, grâce à la NSA (National Security Agency), est modifié le 23 novembre 1976 pour donner le **DES** (*Data Encryption Standard*). Le DES a finalement été approuvé en 1978 par le NBS. Le DES fut normalisé par l'*ANSI* (*American National Standard Institute*) sous le nom de *ANSI X3.92*, plus connu sous la dénomination *DEA* (*Data Encryption Algorithm*).

Principe du DES

Il s'agit d'un système de chiffrement symétrique par blocs de 64 bits, dont 8 bits (un octet) servent de test de parité (pour vérifier l'intégrité de la clé). Chaque bit de parité de la clé (1 tous les 8 bits) sert à tester un des octets de la clé par parité impaire, c'est-à-dire que chacun des bits de parité est ajusté de façon à avoir un nombre impair de '1' dans l'octet à qui il appartient. La clé possède donc une longueur « utile » de 56 bits, ce qui signifie que seuls 56 bits servent réellement dans l'algorithme.

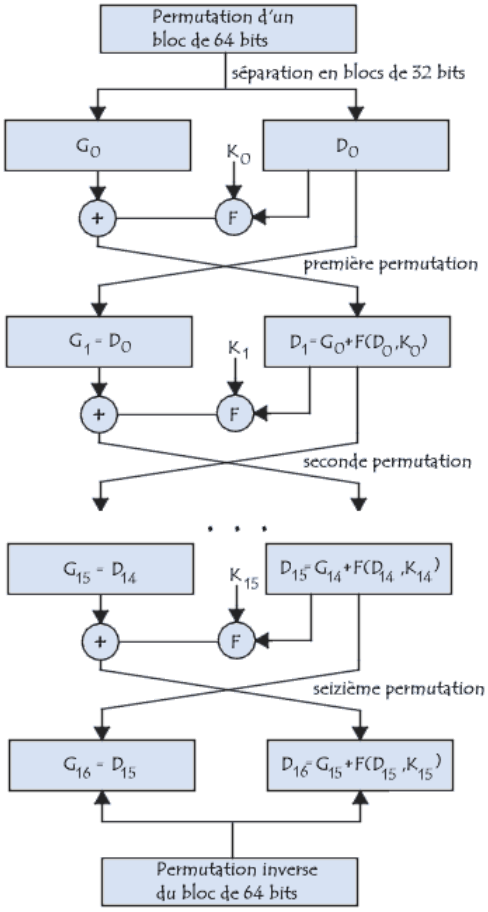
L'algorithme consiste à effectuer des combinaisons, des substitutions et des permutations entre le texte à chiffrer et la clé, en faisant en sorte que les opérations puissent se faire dans les deux sens (pour le déchiffrement). La combinaison entre substitutions et permutations est appelée **code produit** .

La clé est codée sur 64 bits et formée de 16 blocs de 4 bits, généralement notés k_1 à k_{16}. Etant donné que « seuls » 56 bits servent effectivement à chiffrer, il peut exister 2^{56} (soit $7.2*10^{16}$) clés différentes !

L'algorithme du DES

Les grandes lignes de l'algorithme sont les suivantes :

- Fractionnement du texte en blocs de 64 bits (8 octets) ;
- Permutation initiale des blocs ;
- Découpage des blocs en deux parties: gauche et droite, nommées *G* et *D* ;
- Etapes de permutation et de substitution répétées 16 fois (appelées **rondes**) ;
- Recollement des parties gauche et droite puis permutation initiale inverse.



Fractionnement du texte

Permutation initiale

Dans un premier temps, chaque bit d'un bloc est soumis à la permutation initiale, pouvant être représentée par la matrice de permutation initiale (notée *PI*) suivante :

PI	58	50	42	34	26	18	10	2
	60	52	44	36	28	20	12	4
	62	54	46	38	30	22	14	6
	64	56	48	40	32	24	16	8
	57	49	41	33	25	17	9	1
	59	51	43	35	27	19	11	3
	61	53	45	37	29	21	13	5
	63	55	47	39	31	23	15	7

Cette matrice de permutation indique, en parcourant la matrice de gauche à droite puis de haut en bas, que le 58ème bit du bloc de texte de 64 bits se retrouve en première position, le 50ème en seconde position et ainsi de suite.

Scindement en blocs de 32 bits

Une fois la permutation initiale réalisée, le bloc de 64 bits est scindé en deux blocs de 32 bits, notés respectivement **G** et **D** (pour gauche et droite, la notation anglo-saxonne étant *L* et *R* pour *Left and Right*). On note $\mathbf{G}_0$ et $\mathbf{D}_0$ l'état initial de ces deux blocs :

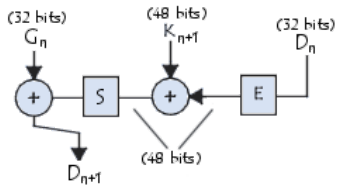
$\mathbf{G}_0$	58	50	42	34	26	18	10	2
	60	52	44	36	28	20	12	4
	62	54	46	38	30	22	14	6
	64	56	48	40	32	24	16	8

$\mathbf{D}_0$	57	49	41	33	25	17	9	1
	59	51	43	35	27	19	11	3
	61	53	45	37	29	21	13	5
	63	55	47	39	31	23	15	7

Il est intéressant de remarquer que $\mathbf{G}_0$ contient tous les bits possédant une position paire dans le message initial, tandis que $\mathbf{D}_0$ contient les bits de position impaire.

Rondes

Les blocs $\mathbf{G}_n$ et $\mathbf{D}_n$ sont soumis à un ensemble de transformation itératives appelées *rondes*, explicitées dans ce schéma, et dont les détails sont donnés plus bas :



Fonction d'expansion

Les 32 bits du bloc $\mathbf{D}_0$ sont étendus à 48 bits grâce à une table (matrice) appelé *table d'expansion* (notée **E**), dans laquelle les 48 bits sont mélangés et 16 d'entre eux sont dupliqués :

E	32	1	2	3	4	5

	4	5	6	7	8	9
	8	9	10	11	12	13
	12	13	14	15	16	17
	16	17	18	19	20	21
	20	21	22	23	24	25
	24	25	26	27	28	29
	28	29	30	31	32	1

Ainsi, le dernier bit de $\mathbf{D}_0$ (c'est-à-dire le 7ème bit du bloc d'origine) devient le premier, le premier devient le second, ...
De plus, les bits 1,4,5,8,9,12,13,16,17,20,21,24,25,28 et 29 de $\mathbf{D}_0$ (respectivement 57, 33, 25, l, 59, 35, 27, 3, 6l, 37, 29, 5, 63, 39, 31 et 7 du bloc d'origine) sont dupliqués et disséminés dans la matrice.

OU exclusif avec la clé

La matrice résultante de 48 bits est appelée $\mathbf{D'}_0$ ou bien $\mathbf{E}[\mathbf{D}_0]$. L'algorithme DES procède ensuite à un *OU exclusif* entre la première clé $\mathbf{K}_1$ et $\mathbf{E}[\mathbf{D}_0]$. Le résultat de ce *OU exclusif* est une matrice de 48 bits que nous appelerons $\mathbf{D}_0$ par commodité (il ne s'agit pas du $\mathbf{D}_0$ de départ!).

Fonction de substitution

$\mathbf{D}_0$ est ensuite scindé en 8 blocs de 6 bits, noté $\mathbf{D}_{0i}$. Chacun de ces blocs passe par des **fonctions de sélection** (appelées parfois *boîtes de substitution* ou *fonctions de compression*), notées généralement $\mathbf{S}_i$.
Les premiers et derniers bits de chaque $\mathbf{D}_{0i}$ détermine (en binaire) la ligne de la fonction de sélection, les autres bits (respectivement 2, 3, 4 et 5) déterminent la colonne. La sélection de la ligne se faisant sur deux bits, il y a 4 possibilités (0,1,2,3). La sélection de la colonne se faisant sur 4 bits, il y a 16 possibilités (0 à 15). Grâce à cette information, la fonction de sélection "sélectionne" une valeur codée sur 4 bits.

Voici la première fonction de substitution, représentée par une matrice de 4 par 16 :

$\mathbf{S}_1$	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Soit $\mathbf{D}_{01}$ égal à *101110*. Les premiers et derniers bits donnent *10*, c'est-à-dire 2 en binaire. Les bits 2,3,4 et 5 donnent *0111*, soit 7 en binaire. Le résultat de la fonction de sélection est donc la valeur situé à la ligne n°2, dans la colonne n°7. Il s'agit de la valeur *11*, soit en binaire *111*.

Chacun des 8 blocs de 6 bits est passé dans la fonction de sélection correspondante, ce qui donne en sortie 8 valeurs de 4 bits chacune. Voici les autres fonctions de sélection :

$\mathbf{S}_2$	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

$\mathbf{S}_3$	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

$\mathbf{S}_4$	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

$\mathbf{S}_5$	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

$\mathbf{S}_6$	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

$\mathbf{S}_7$	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

$\mathbf{S}_8$	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
1	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
1	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Chaque bloc de 6 bits est ainsi substitué en un bloc de 4 bits. Ces bits sont regroupés pour former un bloc de 32 bits.

Permutation

Le bloc de 32 bits obtenu est enfin soumis à une permutation **P** dont voici la table :

P								
	16	7	20	21	29	12	28	17
	1	15	23	26	5	18	31	10
	2	8	24	14	32	27	3	9
	19	13	30	6	22	11	4	25

OU Exclusif

L'ensemble de ces résultats en sortie de **P** est soumis à un *OU Exclusif* avec le $\mathbf{G}_0$ de départ (comme indiqué sur le premier schéma) pour donner D1, tandis que le $\mathbf{D}_0$ initial donne $\mathbf{G}_1$.

Itération

L'ensemble des étapes précédentes (*rondes*) est réitéré 16 fois.

Permutation initiale inverse

A la fin des itérations, les deux blocs $\mathbf{G}_{16}$ et $\mathbf{D}_{16}$ sont "recollés, puis soumis à la permutation initiale inverse :

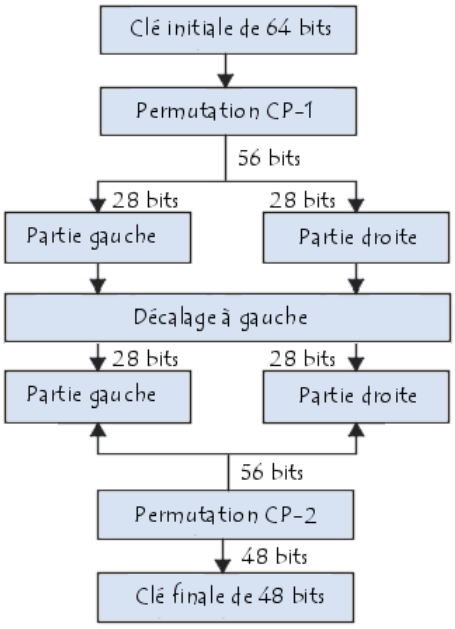
PI-1	40	8	48	16	56	24	64	32
	39	7	47	15	55	23	63	31
	38	6	46	14	54	22	62	30
	37	5	45	13	53	21	61	29
	36	4	44	12	52	20	60	28
	35	3	43	11	51	19	59	27
	34	2	42	10	50	18	58	26
	33	1	41	9	49	17	57	25

Le résultat en sortie est un texte codé de 64 bits !

Génération des clés

Etant donné que l'algorithme du DES présenté ci-dessus est public, toute la sécurité repose sur la complexité des clés de chiffrement.

L'algorithme ci-dessous montre comment obtenir à partir d'une clé de 64 bits (composé de 64 caractères alphanumériques quelconques) 8 clés diversifiées de 48 bits chacune servant dans l'algorithme du DES :



Dans un premier temps les bits de parité de la clé sont éliminés afin d'obtenir une clé d'une longueur utile de 56 bits.

La première étape consiste en une permutation notée **CP-1** dont la matrice est présentée ci-dessous :

CP-1	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

Cette matrice peut en fait s'écrire sous la forme de deux matrice $\mathbf{G}_i$ et $\mathbf{D}_i$ (pour gauche et droite) composées chacune de 28 bits :

$\mathbf{G}_i$	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

$\mathbf{D}_i$	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

On note $\mathbf{G}_0$ et $\mathbf{D}_0$ le résultat de cette première permutation.

Ces deux blocs subissent ensuite une rotation à gauche, de telles façons que les bits en seconde position prennent la première position, ceux en troisième position la seconde, ...
Les bits en première position passent en dernière position.

Les 2 blocs de 28 bits sont ensuite regroupés en un bloc de 56 bits. Celui-ci passe par une permutation, notée **CP-2**, fournissant en sortie un bloc de 48 bits, représentant la clé $\mathbf{K}_i$.

CP-2	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

Des itérations de l'algorithme permettent de donner les 16 clés $\mathbf{K}_1$ à $\mathbf{K}_{16}$ utilisées dans l'algortihme du DES.

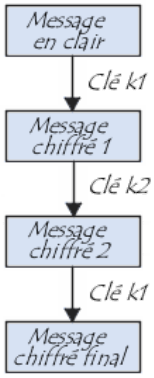
LS	1	2	4	6	8	10	12	14	15	17	19	21	23	25	27	28

TDES, une alternative au DES

En 1990 Eli Biham et Adi Shamir ont mis au point la cryptanalyse différentielle qui recherche des paires de texte en clair et des paires de texte chiffrées. Cette méthode marche jusqu'à un nombre de rondes inférieur à 15, or un nombre de 16 rondes sont présentes dans l'algorithme présenté ci-dessus.

D'autre part, même si une clé de 56 bits donne un nombre énorme de possibilités, de nombreux processeurs permettent de calculer plus de 10^6 clés par seconde, ainsi, utilisés parallèlement sur un très grand nombre de machines, il devient possible pour un grand organisme (un Etat par exemple) de trouver la bonne clé...

Un solution à court terme consiste à chaîner trois chiffrement DES à l'aide de deux clés de 56 bits (ce qui équivait à une clé de 112 bits). Ce procédé est appelé **Triple DES**, noté *TDES* (parfois *3DES* ou *3-DES*).



Le **TDES** permet d'augmenter significativement la sécurité du DES, toutefois il a l'inconvénient majeur de demander également plus de ressources pour les chiffrement et le déchiffrement.

On distingue habituellement plusieurs types de chiffrement triple DES :

- DES-EEE3 : 3 chiffrements DES avec 3 clés différentes ;
- DES-EDE3 : une clé différente pour chacune des 3 opérations DES (chiffrement, déchiffrement, chiffrement) ;
- DES-EEE2 et DES-EDE2 : une clé différente pour la seconde opération (déchiffrement).

En 1997 le *NIST* lança un nouvel appel à projet pour élaborer l'**AES** (*Advanced Encryption Standard*), un algorithme de chiffrement destiné à remplacer le *DES*.

Le système de chiffrement *DES* fût réactualisé tous les 5 ans. En 2000 lors de la dernière révision, après un processus d'évaluation qui a duré 3 années, l'algorithme conçu conjointement par deux candidats belges, Messieurs *Vincent Rijmen* et *Joan Daemen* fût choisi comme nouveau standard par le *NIST*. Ce nouvel algorithme baptisé **RIJNDAEL** par ses inventeurs, remplacera dorénavant le DES.

Plus d'informations

- http://csrc.nist.gov/encryption/tkencryption.html - Spécifications des algorithmes du DES du TDES et de l'AES (site du *NIST*) ;
- RFC 2420 The PPP Triple-DES Encryption Protocol (3DESE)

Trucs & astuces pertinents trouvés dans la base de connaissances

24/01 13h21 Logiciels de Cryptographie & Stéganographie (Cryptographie)
25/09 10h33 PGP était considéré comme une arme (Mythes et légendes)
15/03 09h33 Légalité de la cryptographie en France (Cryptographie)

Plus d'astuces sur « **Cryptographie** »

Discussions pertinentes trouvées dans le forum

14/05 14h19	Nouveau moyen de cryptographie.	Windows	11/12 14h08->LILI for ano...	120
21/08 13h27	[java] cryptographie et fichiers	Développement	26/08 16h21->maily	16
01/06 15h46	Je cherche un algorithme de cryptographie	Développement	12/01 21h37->sebsauvage	8
26/05 11h18	creer logiciel de cryptographie	Développement	26/05 13h22->elrin	8
06/05 15h23	cryptographie	Windows	14/09 12h42->fabienne	6
30/10 21h48	L'évolution des besoins de la cryptographie	Windows	31/10 10h55->sebsauvage	5
22/06 13h54	[cryptographie]	Développement	23/06 18h16->nabilmohcine	4
11/10 13h38	cryptographie	Logiciels/Pilotes	11/10 14h20->lazy	3
10/11 09h39	[SONDAGE] cryptographie et ses conséquences	Virus/Sécurité	10/11 10h28->teebo	3
22/09 10h26	Architecture de Cryptographie Java	Développement	22/09 14h40->Partisan	3

Discussion fermée Problème résolu

Plus de discussions sur « **Cryptographie** »