Cryptosystèmes symétriques : le DES et l'AES

Christophe Guyeux

IUT de Belfort-Montbéliard

Cours, Mathématiques et Informatique, 2008

Avant-propos

- On étudie en détail un système cryptographique à clé privée.
- On présente dans le détail le standard cryptographique DES : son algorithme, sa cryptanalyse. On parle aussi, un peu, du Triple-DES et de l'AES.

Plan

Le DES : présentation générale

Le DES : présentation générale

Histoire

Histoire du DES

- En 1973, le National Bureau of Standards (NBS) américain lança un appel d'offres pour un système de chiffrement.

Histoire du DES

- En 1973, le National Bureau of Standards (NBS) américain lança un appel d'offres pour un système de chiffrement.
- Comme IBM disposait déjà d'un algorithme de chiffrement symétrique, nommé *Lucifer*, elle le proposa au concours.

Histoire du DES

- En 1973, le National Bureau of Standards (NBS) américain lança un appel d'offres pour un système de chiffrement.
- Comme IBM disposait déjà d'un algorithme de chiffrement symétrique, nommé *Lucifer*, elle le proposa au concours.
- Il fut retenu et modifié, pour devenir le code DES : Data Encryption System. Il a notamment été utilisé dans le système de mots de passe Unix.

Histoire du DES

- En 1973, le National Bureau of Standards (NBS) américain lança un appel d'offres pour un système de chiffrement.
- Comme IBM disposait déjà d'un algorithme de chiffrement symétrique, nommé *Lucifer*, elle le proposa au concours.
- Il fut retenu et modifié, pour devenir le code DES : Data Encryption System. Il a notamment été utilisé dans le système de mots de passe Unix.
- Il fut remplacé en 2002 par l'AES (Advanced Encryption System, quand des attaques ont montré sa vulnérabilité.

Le DES : présentation générale

Le DES : présentation générale

L'algorithme

Algorithme du DES

- Le message à chiffrer est tronçonné en mots de 64 bits.

Algorithme du DES

- Le message à chiffrer est tronçonné en mots de 64 bits.
- Chaque bloc sera crypté séparément.

Algorithme du DES

- Le message à chiffrer est tronçonné en mots de 64 bits.
- Chaque bloc sera crypté séparément.
- Voyons le cryptage d'un bloc en détail...

Schéma du cryptage d'un bloc

- Le cryptage d'un bloc s'effectue en trois étapes :

Schéma du cryptage d'un bloc

- Le cryptage d'un bloc s'effectue en trois étapes :
  1. Le mot subit une permutation initiale.

Schéma du cryptage d'un bloc

- Le cryptage d'un bloc s'effectue en trois étapes :
  1. Le mot subit une permutation initiale.
  2. Le mot obtenu est scindé en sa partie gauche G_0 et sa partie droite D_0, chacune de 32 bits, et est transformé en $M_1 = G_1 D_1$, où
     - $G_1 = D_0$
     - $D_1 = G_0 + f(D_0, K_0)$

Schéma du cryptage d'un bloc

- Le cryptage d'un bloc s'effectue en trois étapes :
  1. Le mot subit une permutation initiale.
  2. Le mot obtenu est scindé en sa partie gauche G_0 et sa partie droite D_0, chacune de 32 bits, et est transformé en $M_1 = G_1 D_1$, où
     - $G_1 = D_0$
     - $D_1 = G_0 + f(D_0, K_0)$
  3. On itère ce procédé pour obtenir $M_2 = G_2 D_2$ à partir de $M_1 = G_1 D_1$, etc. jusqu'à $M_{16} = G_{16} D_{16}$.

Schéma du cryptage d'un bloc

- Le cryptage d'un bloc s'effectue en trois étapes :
  1. Le mot subit une permutation initiale.
  2. Le mot obtenu est scindé en sa partie gauche G_0 et sa partie droite D_0, chacune de 32 bits, et est transformé en $M_1 = G_1 D_1$, où
     - $G_1 = D_0$
     - $D_1 = G_0 + f(D_0, K_0)$
  3. On itère ce procédé pour obtenir $M_2 = G_2 D_2$ à partir de $M_1 = G_1 D_1$, etc. jusqu'à $M_{16} = G_{16} D_{16}$.
  4. On fait subir la permutation inverse de la permutation initiale à $D_{16} G_{16}$.

Les permutations du DES

La permutation initiale

La permutation initiale

La permutation initiale consiste à placer le cinquante-huitième bit en première position, le cinquantième en seconde, etc. :

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

Les permutations du DES

Les permutations du DES

La permutation finale

La permutation finale

- La permutation inverse est alors donnée par le tableau :

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

La permutation finale

- La permutation inverse est alors donnée par le tableau :

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

- Ces permutations n'ont aucun rôle dans la sécurité de l'algorithme.

La fonction $f(D, K)$

La fonction $f(D, K)$

L'expansion initiale

Les opérations de $f(D, K)$

L'action de $f(D, K)$ est constituée de différentes étapes :

1. L'argument de gauche (D, de 32 bits) est expansé en un mot de 48 bits.

L'expansion initiale

- On place le 32^e bit en première position, le premier en seconde, etc.

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

La fonction $f(D, K)$

La fonction $f(D, K)$

Utilisation de la clé secrète

Les opérations de $f(D, K)$

L'action de $f(D, K)$ est constituée de différentes étapes :

1. L'argument de gauche (D, de 32 bits) est expansé en un mot de 48 bits.

Les opérations de $f(D, K)$

L'action de $f(D, K)$ est constituée de différentes étapes :

1. L'argument de gauche (D, de 32 bits) est expansé en un mot de 48 bits.
2. Le mot obtenu est additionné (bit à bit) à K.

La clé secrète, ses bits de contrôle

- La clé secrète est une clé de 56 bits, plus 8 bits de parité ou de contrôle.

La clé secrète, ses bits de contrôle

- La clé secrète est une clé de 56 bits, plus 8 bits de parité ou de contrôle.
- Les bits de contrôle occupent les positions 8, 16, 24, 32, etc.

La clé secrète, ses bits de contrôle

- La clé secrète est une clé de 56 bits, plus 8 bits de parité ou de contrôle.
- Les bits de contrôle occupent les positions 8, 16, 24, 32, etc.
- Ainsi, si les 7 premiers bits sont 0011010, le 8^e sera 1, pour qu'il y ait un nombre pair de 1.

La clé secrète, ses bits de contrôle

- La clé secrète est une clé de 56 bits, plus 8 bits de parité ou de contrôle.
- Les bits de contrôle occupent les positions 8, 16, 24, 32, etc.
- Ainsi, si les 7 premiers bits sont 0011010, le 8^e sera 1, pour qu'il y ait un nombre pair de 1.
- A partir de ces 56+8 bits, il nous faut générer $K_1, K_2, \ldots, K_{16}$, par le *principe de diversification de la clé* suivant...

Principe de diversification de la clé secrète

1. On applique une permutation PC_1 à la clé K (voir ci-après).

Principe de diversification de la clé secrète

1. On applique une permutation PC_1 à la clé K (voir ci-après).
2. On réitère 16 fois de suite :

Principe de diversification de la clé secrète

1. On applique une permutation PC_1 à la clé K (voir ci-après).
2. On réitère 16 fois de suite :
   - A chaque moitié du mot de 56 bits, on effectue un décalage à gauche :
     - d'un cran, aux étapes 1, 2, 9 et 16,
     - de deux crans, aux autres étapes.

Principe de diversification de la clé secrète

1. On applique une permutation PC_1 à la clé K (voir ci-après).
2. On réitère 16 fois de suite :
   - A chaque moitié du mot de 56 bits, on effectue un décalage à gauche :
     - d'un cran, aux étapes 1, 2, 9 et 16,
     - de deux crans, aux autres étapes.
   - A chacune de ces étapes i, on obtient une clé partielle K_i de 48 bits, en appliquant la règle d'extraction PC_2.

La permutation PC_1

- La permutation PC_1 consiste à placer le 57^e bit en première position, etc. :

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

- Où les 56 bits sont numérotés de 1 à 64, en sautant les multiples de 8.

La règle d'extraction PC_2

- De même, la règle d'extraction PC_2 est définie par :

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

- Où les 56 bits sont numérotés de 1 à 64, en sautant les multiples de 8.

La fonction $f(D, K)$

La fonction $f(D, K)$

Les boîtes de substitution

Les opérations de $f(D, K)$

L'action de $f(D, K)$ est constituée de différentes étapes :

1. L'argument de gauche (D, de 32 bits) est expansé en un mot de 48 bits.

Les opérations de $f(D, K)$

L'action de $f(D, K)$ est constituée de différentes étapes :

1. L'argument de gauche (D, de 32 bits) est expansé en un mot de 48 bits.
2. Le mot obtenu est additionné (bit à bit) à K.

Les opérations de $f(D, K)$

L'action de $f(D, K)$ est constituée de différentes étapes :

1. L'argument de gauche (D, de 32 bits) est expansé en un mot de 48 bits.
2. Le mot obtenu est additionné (bit à bit) à K.
3. On subdivise le mot obtenu en 8 mots de 6 bits. Chacun est transformé en mot de 4 bits en utilisant des « boîtes de substitution ».

Les boîtes de substitution

- On dispose de 8 boîtes de substitution différentes.

Les boîtes de substitution

- On dispose de 8 boîtes de substitution différentes.
- En entrée, chacune prend un mot de 6 bits, pour fournir en sortie un mot de 4 bits.

Les boîtes de substitution

- On dispose de 8 boîtes de substitution différentes.
- En entrée, chacune prend un mot de 6 bits, pour fournir en sortie un mot de 4 bits.
- Le premier et le dernier bit du mot d'entrée donnent la ligne à considérer (de 0 à 3).

Les boîtes de substitution

- On dispose de 8 boîtes de substitution différentes.
- En entrée, chacune prend un mot de 6 bits, pour fournir en sortie un mot de 4 bits.
- Le premier et le dernier bit du mot d'entrée donnent la ligne à considérer (de 0 à 3).
- Les 4 autres bits en donne la colonne (de 0 à 15).

Les boîtes de substitution

- On dispose de 8 boîtes de substitution différentes.
- En entrée, chacune prend un mot de 6 bits, pour fournir en sortie un mot de 4 bits.
- Le premier et le dernier bit du mot d'entrée donnent la ligne à considérer (de 0 à 3).
- Les 4 autres bits en donne la colonne (de 0 à 15).
- L'élément à l'intersection de cette ligne et de cette colonne fournit la valeur de retour.

La première des boîtes de substitution

Voici la première des 8 boîtes de substitution :

La première des boîtes de substitution

Voici la première des 8 boîtes de substitution :

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

La première des boîtes de substitution

Voici la première des 8 boîtes de substitution :

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

Si le mot d'entrée de cette boîte est 010101, le mot de sortie sera 12, soit 1100.

La fonction $f(D, K)$

La fonction $f(D, K)$

La dernière permutation

Les opérations de $f(D, K)$

L'action de $f(D, K)$ est constituée de différentes étapes :

1. L'argument de gauche (D, de 32 bits) est expansé en un mot de 48 bits.

Les opérations de $f(D, K)$

L'action de $f(D, K)$ est constituée de différentes étapes :

1. L'argument de gauche (D, de 32 bits) est expansé en un mot de 48 bits.
2. Le mot obtenu est additionné (bit à bit) à K.

Les opérations de $f(D, K)$

L'action de $f(D, K)$ est constituée de différentes étapes :

1. L'argument de gauche (D, de 32 bits) est expansé en un mot de 48 bits.
2. Le mot obtenu est additionné (bit à bit) à K.
3. On subdivise le mot obtenu en 8 mots de 6 bits. Chacun est transformé en mot de 4 bits en utilisant des « boîtes de substitution ».

Les opérations de $f(D, K)$

L'action de $f(D, K)$ est constituée de différentes étapes :

1. L'argument de gauche (D, de 32 bits) est expansé en un mot de 48 bits.
2. Le mot obtenu est additionné (bit à bit) à K.
3. On subdivise le mot obtenu en 8 mots de 6 bits. Chacun est transformé en mot de 4 bits en utilisant des « boîtes de substitution ».
4. Enfin, le mot concaténé résultant (de 32 bits), subit une dernière permutation.

La dernière permutation

- On place le 16^e bit en première position, le septième en seconde, etc.

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

Attaques et contre-attaques

Attaques et contre-attaques

Les attaques

Utilisation du DES

- DES est une méthode de chiffrement utilisant des clés de 56 bits.

Utilisation du DES

- DES est une méthode de chiffrement utilisant des clés de 56 bits.
- Son emploi n'est plus recommandé aujourd'hui :
  - A cause de sa lenteur à l'exécution
  - Du fait de son espace de clés trop petit permettant une attaque systématique en un temps raisonnable.

Utilisation du DES

- DES est une méthode de chiffrement utilisant des clés de 56 bits.
- Son emploi n'est plus recommandé aujourd'hui :
  - A cause de sa lenteur à l'exécution
  - Du fait de son espace de clés trop petit permettant une attaque systématique en un temps raisonnable.
- Quand il est encore utilisé c'est généralement en Triple DES, ce qui ne fait rien pour améliorer ses performances.

Attaques

- Plusieurs attaques ont été découvertes sur DES.

Attaques

- Plusieurs attaques ont été découvertes sur DES.
- Elles permettent de diminuer les coûts d'une recherche exhaustive des clés qui se monte à 2^{55} opérations en moyenne.

Attaque par cryptanalyse différentielle

- La cryptanalyse différentielle, découverte par Eli Biham et Adi Shamir en 1991.

Attaque par cryptanalyse différentielle

- La cryptanalyse différentielle, découverte par Eli Biham et Adi Shamir en 1991.
- Elle permet de trouver la clé en utilisant 2^{47} textes clairs.

Attaque par cryptanalyse différentielle

- La cryptanalyse différentielle, découverte par Eli Biham et Adi Shamir en 1991.
- Elle permet de trouver la clé en utilisant 2^{47} textes clairs.
- Le principe est de disposer d'un DES implémenté dans une boîte noire hermétique avec une clé secrète à l'intérieur.

Attaque par cryptanalyse différentielle

- La cryptanalyse différentielle, découverte par Eli Biham et Adi Shamir en 1991.
- Elle permet de trouver la clé en utilisant 2^{47} textes clairs.
- Le principe est de disposer d'un DES implémenté dans une boîte noire hermétique avec une clé secrète à l'intérieur.
- En fournissant suffisamment de texte en entrée, on peut statistiquement analyser le comportement des sorties selon les entrées et retrouver la clé.

Attaque par cryptanalyse différentielle

- La cryptanalyse différentielle, découverte par Eli Biham et Adi Shamir en 1991.
- Elle permet de trouver la clé en utilisant 2^{47} textes clairs.
- Le principe est de disposer d'un DES implémenté dans une boîte noire hermétique avec une clé secrète à l'intérieur.
- En fournissant suffisamment de texte en entrée, on peut statistiquement analyser le comportement des sorties selon les entrées et retrouver la clé.
- Les entrées utilisées pour cette attaque doivent mutuellement présenter une légère différence.

Attaque par cryptanalyse différentielle

- La cryptanalyse différentielle, découverte par Eli Biham et Adi Shamir en 1991.
- Elle permet de trouver la clé en utilisant 2^{47} textes clairs.
- Le principe est de disposer d'un DES implémenté dans une boîte noire hermétique avec une clé secrète à l'intérieur.
- En fournissant suffisamment de texte en entrée, on peut statistiquement analyser le comportement des sorties selon les entrées et retrouver la clé.
- Les entrées utilisées pour cette attaque doivent mutuellement présenter une légère différence.
- En regardant comment la différence affecte la sortie, on peut établir des statistiques...

L'attaque-T (Tickling attack)

- L'attaque-T (Tickling attack), variante de la cryptanalyse différentielle.

L'attaque-T (Tickling attack)

- L'attaque-T (Tickling attack), variante de la cryptanalyse différentielle.
- Elle a été découverte lors de la conception du DES par les chercheurs d'IBM...

L'attaque-T (Tickling attack)

- L'attaque-T (Tickling attack), variante de la cryptanalyse différentielle.
- Elle a été découverte lors de la conception du DES par les chercheurs d'IBM...
- ...et révélée par Don Coppersmith au milieu des années 90.

L'attaque-T (Tickling attack)

- L'attaque-T (Tickling attack), variante de la cryptanalyse différentielle.
- Elle a été découverte lors de la conception du DES par les chercheurs d'IBM...
- ...et révélée par Don Coppersmith au milieu des années 90.
- Pendant une vingtaine d'années, le silence a été complet sur cette découverte.

L'attaque-T (Tickling attack)

- L'attaque-T (Tickling attack), variante de la cryptanalyse différentielle.
- Elle a été découverte lors de la conception du DES par les chercheurs d'IBM...
- ...et révélée par Don Coppersmith au milieu des années 90.
- Pendant une vingtaine d'années, le silence a été complet sur cette découverte.
- À l'époque, elle avait incité les concepteurs de DES à renforcer le contenu des tables de substitution (au lieu de l'affaiblir comme la rumeur le laissait entendre)

Le compromis temps-mémoire

- Le compromis temps-mémoire, concept inventé par Martin Hellman au début des années 80.

Le compromis temps-mémoire

- Le compromis temps-mémoire, concept inventé par Martin Hellman au début des années 80.
- Principe : le même message va être chiffré plusieurs fois avec des clés différentes, (calcul d'une immense table qui contient toutes les versions chiffrées de ce message).

Le compromis temps-mémoire

- Le compromis temps-mémoire, concept inventé par Martin Hellman au début des années 80.
- Principe : le même message va être chiffré plusieurs fois avec des clés différentes, (calcul d'une immense table qui contient toutes les versions chiffrées de ce message).
- Lorsque l'on intercepte un message chiffré, on peut le retrouver dans la table, et obtenir la clé qui avait été utilisée pour le coder.

Le compromis temps-mémoire

- Le compromis temps-mémoire, concept inventé par Martin Hellman au début des années 80.
- Principe : le même message va être chiffré plusieurs fois avec des clés différentes, (calcul d'une immense table qui contient toutes les versions chiffrées de ce message).
- Lorsque l'on intercepte un message chiffré, on peut le retrouver dans la table, et obtenir la clé qui avait été utilisée pour le coder.
- Cette attaque n'est bien sûr pas faisable : besoin d'une table de l'ordre du milliard de GB.

Le compromis temps-mémoire

- Le compromis temps-mémoire, concept inventé par Martin Hellman au début des années 80.
- Principe : le même message va être chiffré plusieurs fois avec des clés différentes, (calcul d'une immense table qui contient toutes les versions chiffrées de ce message).
- Lorsque l'on intercepte un message chiffré, on peut le retrouver dans la table, et obtenir la clé qui avait été utilisée pour le coder.
- Cette attaque n'est bien sûr pas faisable : besoin d'une table de l'ordre du milliard de GB.
- Hellman a trouvé un moyen pour réduire cette table à 1 téraoctet, ce qui est faisable de nos jours.

Autres attaques, spécifiques

- D'autres attaques sont spécifiques à des implémentations et ne sont pas forcément spécifiques à DES.

Autres attaques, spécifiques

- D'autres attaques sont spécifiques à des implémentations et ne sont pas forcément spécifiques à DES.
- Dans le cas d'un DES implémenté dans du matériel, on pourrait analyser la consommation électrique et déduire certaines informations sur la clé (une consommation accrue indique des bits actifs).

Autres attaques, spécifiques

- D'autres attaques sont spécifiques à des implémentations et ne sont pas forcément spécifiques à DES.
- Dans le cas d'un DES implémenté dans du matériel, on pourrait analyser la consommation électrique et déduire certaines informations sur la clé (une consommation accrue indique des bits actifs).
- Le même style d'attaque peut aussi être employé sur un ordinateur en calculant le temps mis pour chiffrer avec des textes différents ou en analysant la mémoire utilisée.

Autres attaques, spécifiques

- Toutes les autres attaques sur DES visent à réduire le temps de calcul d'une recherche exhaustive en utilisant des machines spécifiquement conçues pour la tâche.

Autres attaques, spécifiques

- Toutes les autres attaques sur DES visent à réduire le temps de calcul d'une recherche exhaustive en utilisant des machines spécifiquement conçues pour la tâche.
- Une telle machine a été construite en 1998.
- *Deep Crack* a coûté environ 200 000 dollars et pouvait casser la clé en moins d'une semaine.

Autres attaques, spécifiques

- Toutes les autres attaques sur DES visent à réduire le temps de calcul d'une recherche exhaustive en utilisant des machines spécifiquement conçues pour la tâche.
- Une telle machine a été construite en 1998.
- *Deep Crack* a coûté environ 200 000 dollars et pouvait casser la clé en moins d'une semaine.

Autres attaques, spécifiques

- Toutes les autres attaques sur DES visent à réduire le temps de calcul d'une recherche exhaustive en utilisant des machines spécifiquement conçues pour la tâche.
- Une telle machine a été construite en 1998.
- *Deep Crack* a coûté environ 200 000 dollars et pouvait casser la clé en moins d'une semaine.
- Le calcul distribué en utilisant les ordinateurs des particuliers a prouvé son efficacité en cassant une clé en moins de 24 heures.

Attaques et contre-attaques

Attaques et contre-attaques

Les contre-attaques

Le Triple-DES

- Pour pallier la faiblesse du DES, des chercheurs du projet DES chez IBM ont développé nu algorithme de chiffrement qui enchaîne trois DES sur le même bloc de 64 bits.

Le Triple-DES

- Pour pallier la faiblesse du DES, des chercheurs du projet DES chez IBM ont développé nu algorithme de chiffrement qui enchaîne trois DES sur le même bloc de 64 bits.
- On peut utiliser, pour cela, deux ou trois clés différentes.

Le Triple-DES

- Pour pallier la faiblesse du DES, des chercheurs du projet DES chez IBM ont développé nu algorithme de chiffrement qui enchaîne trois DES sur le même bloc de 64 bits.
- On peut utiliser, pour cela, deux ou trois clés différentes.
- La version la plus sûre utilise un chiffrement, suivi d'un déchiffrement pour se conclure à nouveau par un chiffrement.

Le Triple-DES

- Le Triple-DES est généralement utilisé avec deux clés différentes.

Le Triple-DES

- Le Triple-DES est généralement utilisé avec deux clés différentes.
- Il peut s'écrire formellement

$$Triple - DES_{k_1,k_2} = DES_{k_1} \circ DES_{k_2^{-1}} \circ DES_{k_1}$$

Le Triple-DES

- Le Triple-DES est généralement utilisé avec deux clés différentes.
- Il peut s'écrire formellement

$$Triple - DES_{k_1,k_2} = DES_{k_1} \circ DES_{k_2^{-1}} \circ DES_{k_1}$$

- Une clé triple DES est donc composée de deux clés DES, et fait 112 bits.

Le Triple-DES

- Le Triple-DES est généralement utilisé avec deux clés différentes.
- Il peut s'écrire formellement

$$Triple - DES_{k_1,k_2} = DES_{k_1} \circ DES_{k_2^{-1}} \circ DES_{k_1}$$

- Une clé triple DES est donc composée de deux clés DES, et fait 112 bits.
- Cela met donc Triple-DES hors de portée d'une attaque exhaustive.

L'AES

- Cependant cet algorithme, assez simple à mettre en œuvre, est lent.

L'AES

- Cependant cet algorithme, assez simple à mettre en œuvre, est lent.
- Il a laissé la place à AES, dont les principes restent analogues.