

Sécurité des systèmes informatiques
Le chiffrement symétrique

Olivier Markowitch

Définitions

- Si a , b et m sont des entiers et si $m > 0$, nous écrivons $a \equiv b \pmod{m}$ si m divise $b - a$ (nous disons a est congru à b modulo m , et m est appelé le *modulus*)
- Une *fonction* (ou *transformation*) est définie par deux ensembles X et Y et par une règle f qui assigne chaque élément de X à précisément un élément de Y . Nous notons cela $f : X \longrightarrow Y$
- L'ensemble X est appelé le *domaine* et l'ensemble Y le *codomaine*
- Si $x \in X$ est tel que $y = f(x) \in Y$, alors y est appelé l'*image de x*
- La *préimage de $y \in Y$* est un $x \in X$ tel que $f(x) = y$

Définitions (suite)

- L'ensemble des éléments de Y qui ont au-moins une préimage est appelé *image de f*
- Une fonction f est une *injection* si chaque élément du codomaine Y est l'image d'au plus un élément du domaine X
- Une fonction f est une *surjection* si chaque élément du codomaine Y est l'image d'au moins un élément du domaine
- Une fonction f est une *bijection* si cette fonction est injective et surjective

Définitions (suite)

- Une fonction f est une *fonction à sens unique* si $f(x)$ est « facile » à calculer pour tous les éléments de X mais pour un y appartenant à l'image de f et choisi au hasard, il est calculatoirement infaisable de trouver un $x \in X$ tel que $f(x) = y$
- Une fonction f à sens unique est dite *fonction trappe* (« trapdoor function ») si étant donné une information appelée *information trappe* il devient possible de trouver, pour tout y de l'image de f , un x tel que $f(x) = y$
- Soit S un ensemble fini d'éléments, une *permutation p sur S* est une bijection de S sur lui-même

Définitions (suite)

- Soit A un ensemble fini, appelé l'*alphabet de définition*
- Soit M un ensemble appelé l'*espace des messages clairs* où chaque message est un string de symboles d'un alphabet de définition
- Soit C un ensemble appelé l'*espace des messages chiffrés* où chaque message chiffré est un string de symboles d'un alphabet de définition
- Soit K un ensemble appelé *espace des clés*
- Chaque $k \in K$, détermine une injection E_k de M vers C , appelée *fonction de chiffrement*, k est appelé *clé de chiffrement*

Définitions (suite)

- A chaque k est associé un $k' \in K$, tel que $D_{k'}$ dénote une injection de C vers M , appelée *fonction de déchiffrement*, k' est appelé *clé de déchiffrement*
- Le processus consistant à appliquer une transformation E paramétrisée par k et notée E_k à un message $m \in M$ est appelé *chiffrement de m*
- Le processus consistant à appliquer une transformation D paramétrisée par k' et notée $D_{k'}$ à un message chiffré $c \in C$ est appelé *déchiffrement de c*
- Un *schéma de chiffrement* consiste en un quintuplet (M, C, K, E, D) tel que $\forall k$ et $k' \in K$, il existe E_k et $D_{k'} : \forall x \in M : D_{k'}(E_k(x)) = x$. Notons que si $M = C$, alors notre chiffrement est une permutation

Définitions (suite)

- Un *canal* est le moyen utilisé pour faire parvenir de l'information d'une *entité* à une autre
- Un schéma de chiffrement est *cassable* si une tierce partie adverse, sans connaître k et k' peut systématiquement retrouver le message clair depuis un message chiffré, dans un temps raisonnable
- Un *groupe* $(G, *)$ consiste en un ensemble G avec une opération binaire $*$ sur G tel que cette opération est interne, associative, possède un neutre et est symétrique. Un groupe dont l'opération $*$ est de plus commutative est appelé un *abélien*. Par exemple \mathbb{Z} associé à l'addition forme un groupe

Définitions (suite)

- Un *groupe fini* est un groupe composé d'un nombre fini d'éléments ($|G|$ est fini). Ce nombre d'éléments est appelé *l'ordre* du groupe. Par exemple l'ensemble \mathbb{Z}_n avec comme opération l'addition modulo n forme un groupe fini d'ordre n
- Un sous-ensemble H non vide de G est un *sous-groupe* de G si H est lui-même un groupe pour l'opération associée à G
- Un groupe G est *cyclique* s'il existe un élément $\alpha \in G$ tel que pour chaque $b \in G$, il existe un entier i pour lequel nous avons $b = \alpha^i$. Un tel élément α est appelé un *générateur* de G

Définitions (suite)

- Si G est un groupe et $a \in G$, l'*ordre de a* est le plus petit entier $t > 0$ tel que $a^t = 1$. Si un tel t n'existe pas, l'ordre est dit infini
- Un *anneau* $(R, +, \times)$ consiste en un ensemble R associé à deux opérations binaires sur R tel que $(R, +)$ est un groupe abélien, l'opération \times est interne, associative, possède un neutre et l'opération \times est distributive par rapport à $+$. Si R associé à \times est de plus commutative, l'anneau est dit *anneau commutatif*. Par exemple \mathbb{Z} et \mathbb{Z}_n associés à l'addition et la multiplication (modulo n pour \mathbb{Z}_n) sont deux anneaux commutatifs
- Un élément a d'un anneau est dit *inversible* s'il existe un élément $b \in G$ tel que $a \times b = 1$

Définitions (suite)

- Un *champ* est un anneau commutatif dans lequel tous les éléments, excepté le neutre pour la première loi, possèdent un élément symétrique pour la seconde loi
- Un sous-ensemble F d'un champ E est un *sous-champ de E* si F est lui-même un champ en regard des opérations associées à E
- Un *champ fini* est un champ qui contient un nombre fini d'éléments. L'*ordre* de ce champ est le nombre d'éléments du champ. Si F est un champ fini de q éléments, il est alors noté $GF(q)$ ou F_q

Le chiffrement symétrique

Définitions

Un schéma de chiffrement est dit *symétrique* (ou à *clés secrètes*) si pour chaque paire de clés k et k' , il est « facile » de déterminer k connaissant k' et réciproquement

Définition

Un schéma de *chiffrement par blocs* est un schéma de chiffrement qui découpe les messages clairs en blocs (strings) de taille fixe t et chiffre un bloc à la fois

La cryptanalyse

Principe de *Kerckhoff* : le système de chiffrement utilisé est connu

Les attaques les plus habituelles sont les :

- attaques à **texte chiffré connu** où l'opposant ne connaît que les chiffrés (*known ciphertext attack*)
- les attaques à **textes clairs connus** où l'opposant dispose de textes clairs correspondant à des textes chiffrés (*known plaintext attack*)
- les attaques à **textes clairs choisis** où l'opposant peut choisir le texte clair et en obtenir le texte chiffré correspondant (*chosen plaintext attack*)
- les attaques à **textes chiffrés choisis** où l'opposant peut choisir le texte chiffré et en obtenir le texte clair correspondant (*chosen ciphertext attack*)

Chiffrement par décalage

Soit $M = C = K = \mathbb{Z}_{26}$. Soit $0 \leq k \leq 25$. Et soit x et $y \in \mathbb{Z}_{26}$

$$E_k(x) = x + k \pmod{26}$$

et

$$D_k(y) = y - k \pmod{26}$$

Chaque lettre de l'alphabet est représenté par une valeur comprise entre 0 et 25

Pour la valeur de clé $k = 3$ nous retrouvons le chiffrement de César

Exemple : Avec $k = 3$ et le texte clair est « CESAR », nous obtenons le chiffré « FHVDU »

Cette méthode est facilement cassable par recherche exhaustive (26 clés possibles)

Chiffrement par substitution

Soit $M = C = \mathbb{Z}_{26}$. K est l'ensemble des permutations sur $\{0, \dots, 25\}$. Pour chaque permutation $k \in K$, nous définissons :

$$E_k(x) = k(x)$$

et

$$D_{k^{-1}}(y) = k^{-1}(y)$$

où x et $y \in \mathbb{Z}_{26}$ et k^{-1} est la permutation inverse de k

cette méthode est aussi appelée *chiffrement par substitution mono-alphabétique*. Chacune des lettres est remplacée par une autre lettre de l'alphabet

Le chiffrement réalisant donc une permutation de l'ensemble des lettres de l'alphabet, le nombre de clés possibles est $26! > 4 \cdot 10^{26}$, une recherche exhaustive n'est donc pas possible

Chiffrement par substitution (suite)

Avec la permutation proposée, la phrase :

« chiffrementparpermutation »

devient :

« YGZPPCHTHSMLXCLHCTUMXMZFS »

Attaque : la fréquence des lettres est préservée (par exemple la lettre 'e' est la lettre statistiquement la plus utilisée en français et en anglais. Donc la lettre apparaissant le plus dans le chiffré a beaucoup de chance d'être un 'e'. Puis on étudie les digrammes et trigrammes (dont on connaît aussi les probabilités d'apparition) et on retrouve ainsi des parties importantes du message clair par simple correspondance

Chiffrement par fonction affine

Soit $M = C = \mathbb{Z}_{26}$

Soit $K = \{(a, b) \in \mathbb{Z}_{26} \times \mathbb{Z}_{26} : \text{pgcd}(a, 26) = 1\}$.

Pour $k = (a, b) \in K$, nous définissons :

$$E_k(x) = a \cdot x + b \pmod{26}$$

et

$$D_k(y) = a^{-1} \cdot (y - b) \pmod{26}$$

où x et $y \in \mathbb{Z}_{26}$

Chiffrement par fonction affine (suite)

Si $k = (7, 3)$, nous avons $7^{-1} \bmod 26 = 15$. Nous obtenons :

$$E_k(x) = 7x + 3$$

et

$$D_k(y) = 15(y - 3) = 15y - 19$$

Si nous voulons chiffrer « hello », nous le transposons en nombre : 7 4 11 11 14, que nous chiffons : 0 5 2 2 23 ce qui nous donne « AFCCX »

Le nombre de clés possibles est 26 fois le nombre d'éléments premiers avec 26

Il y a 12 valeurs ≤ 26 qui sont premiers avec 26 : 1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23 et 25

Le nombre de clés possibles $26 \cdot 12 = 312$

Chiffrement par fonction affine (suite)

Attaque : dans un premier temps de la même manière que précédemment par comptage du nombre d'occurrences de chaque élément dans le chiffré

Réalisation d'une correspondance avec la table de fréquences des lettres

Les hypothèses réalisées lors de la correspondance sont vérifiables : si on pense que le chiffré de r_1 est s_1 et que le chiffré de r_2 est s_2 , en résolvant le système composé par $r_1 \cdot a + b = s_1$ et $r_2 \cdot a + b = s_2$ nous trouvons une solution unique pour a et b dans \mathbb{Z}_{26} , si le $\text{pgcd}(a, 26) \neq 1$ on sait que la correspondance est mauvaise et on essaie une autre correspondance (toujours sur base de la table de fréquence des lettres)

Substitution polyalphabétique

Un schéma de *chiffrement par substitution polyalphabétique* est un chiffrement par blocs de longueur t sur un alphabet A ayant les propriétés suivantes :

- E consiste en tous les ensembles de t permutations où chaque permutation est définie sur l'ensemble A
- chaque clé $k \in K$ définit un ensemble de t permutations (p_1, \dots, p_t)
- le chiffrement d'un message $x = x_1 \dots x_t$ avec la clé k est donnée par :

$$E_k(x) = p_1(x_1) \dots p_t(x_t)$$

- la clé de déchiffrement k' définit D_k l'ensemble des t permutations inverses à celles de $E_k : (p_1^{-1}, \dots, p_t^{-1})$

Chiffrement de Vigenère

Méthode inventée par Blaise Vigenère au XVI^{me} siècle

Soit m un entier strictement positif, soit $M = C = K = (\mathbb{Z}_{26})^m$. Pour toute clé $k = (k_1, \dots, k_m)$, nous définissons :

$$E_k(x) = E_k(x_1, \dots, x_m) = (x_1 + k_1, \dots, x_m + k_m)$$

et

$$D_k(y) = D_k(y_1, \dots, y_m) = (y_1 - k_1, \dots, y_m - k_m)$$

où x et $y \in \mathbb{Z}_{26}$ et où les opérations sont réalisées dans \mathbb{Z}_{26}

Chiffrement de Vigenère (suite)

En pratique pour chiffrer on utilise comme clé un string de longueur m appelé *mot-clé* que l'on convertit en nombre

Le nombre de mot-clés possibles de longueur m est 26^m , et donc une recherche exhaustive est impossible

La cryptanalyse qui est un peu plus complexe permet de retrouver la taille du mot clé (grâce au test de Kasiski), puis sa valeur (grâce à l'indice de coïncidence mutuel)

Chiffrement de Vigenère (suite)

Exemple : on utilise comme mot clé le mot « hello » qui convertit en nombre nous donne 7 4 11 11 14. Pour chiffrer la phrase « rendezvousahuitheure » nous réalisons les additions :

17	04	13	03	04	25	21	14	20	18
07	04	11	11	14	07	04	11	11	14
--	--	--	--	--	--	--	--	--	--
24	08	24	14	18	06	25	25	05	06

00	07	20	08	19	07	04	20	17	04
07	04	11	11	14	07	04	11	11	14
--	--	--	--	--	--	--	--	--	--
07	11	05	19	07	14	08	05	02	18

Le texte chiffré est donc :

« YIYOSGZZFGHLFTHOIFCS »

Chiffrement par permutation

Soit m un entier strictement positif

soit $M = C = \{0, \dots, 25\}^m$

soit K l'ensemble des permutations de $\{1, \dots, m\}$

Pour toute clé $k \in K$ qui est une permutation, nous définissons :

$$E_k(x) = E_k(x_1, \dots, x_m) = (x_{k(1)}, \dots, x_{k(m)})$$

et

$$D_{k'}(y) = D_{k'}(y_1, \dots, y_m) = (y_{k^{-1}(1)}, \dots, y_{k^{-1}(m)})$$

où x et $y \in \mathbb{Z}_{26}$ et k^{-1} est la permutation inverse de k

Chiffrement par permutation (suite)

Par exemple : supposons que nous avons la permutation suivante :

$1 \rightarrow 3, 2 \rightarrow 5, 3 \rightarrow 1, 4 \rightarrow 6, 5 \rightarrow 4, 6 \rightarrow 2$

La permutation inverse est :

$1 \rightarrow 3, 2 \rightarrow 6, 3 \rightarrow 1, 4 \rightarrow 5, 5 \rightarrow 2, 6 \rightarrow 4$

Supposons que nous avons le texte clair suivant et des blocs de 6 lettres :

« annulerlelancement »

que nous regroupons donc en :

« annule rlelan cement »

et qui est chiffré en :

« NEALNU ENRALL MTCNEE »

Chiffrement par permutation (suite)

Un chiffrement par permutation, appelé aussi chiffrement par transposition, préserve les occurrences de chaque symbole dans le bloc après chiffrement et peut donc être facilement cryptanalysé

Les chiffrements par substitution et par transposition étant séparément facilement cryptanalysable, il est classique de les combiner dans les méthodes modernes de chiffrements symétriques obtenant ainsi une robustesse accrue

La scytale

La scytale, ou bâton de Plutarque, consiste en un bâton autour duquel on enroule une bande de tissu, le message est écrit sur le tissu ainsi enroulé en écrivant une lettre sur chaque circonvolution ; la bande est ensuite alors déroulée. Pour déchiffrer, il faut donc utiliser un bâton de même diamètre que le bâton original et y enrouler à nouveau la bande de tissu

Chiffrements historiques

En résumé :

- Chiffrement par substitution
 - par décalage : $y = x + k \pmod{26}$
 - par substitution : $y = k(x)$
 - par fonction affine : $y = ax + b \pmod{26}$
 - Vigenère : $y_i = x_i + k_i$
- Par transposition (anagramme)
 - par permutation

Enigma

Enigma est une famille de machines électro-mécaniques composées d'un clavier, d'au moins trois rangées de disques rotatifs (rotors), et d'un affichage. A chaque pression sur une touche, un ou plusieurs rotors tournent

Pour chiffrer puis déchiffrer un message, il faut que l'émetteur et le receveur configurent leur machine de la même façon (c'est la clé secrète), à savoir, entre autres, la position initiale des rotors, l'ordre dans lequel on place les rotors (les rotors sont tous différents)

Le schéma de Vernam

Le chiffrement de Vernam appelé aussi *one-time pad* est défini sur $M = 0, 1$

Un message binaire $x_1 \dots x_t$ est modifié par une clé binaire $k_1 \dots k_t$ de même taille, de la manière suivante :

$$E_k(x) = y = x_i \oplus k_i, 1 \leq i \leq t$$

et

$$D_k(m) = x = y_i \oplus k_i, 1 \leq i \leq t$$

Si la clé est choisie aléatoirement le chiffré est aléatoire et si cette même clé n'est plus jamais utilisée alors le chiffrement est incassable

Le schéma de Vernam (suite)

Par contre si une même clé est utilisée deux fois alors les chiffrés y et y' produits en appliquant cette clé à x et x' sont tels que $x_i \oplus x'_i = y_i \oplus y'_i$ et donc par analyse du xor des deux chiffrés on enlève l'aléatoire introduit par la clé et on peut analyser statistiquement cette somme binaire pour en extraire les deux messages clairs x et x'

Il a été prouvé qu'un cryptosystème incassable doit (au-moins) avoir une clé aussi longue que les messages clairs. Ce qui est peu pratique. Pourtant c'est (à peu près) le mécanisme qui aurait été utilisé comme téléphone rouge entre les USA et l'URSS (les clés étant échangées par des personnes de confiance)

Chiffrements en chaîne

SEAL et RC4 sont des cryptosystèmes réalisant le chiffrement en chaîne

Phillip Rogaway et Don Coppersmith *A Software-Optimized Encryption Algorithm*, Fast Software Encryption, Lecture Notes in Computer Science 809, Springer-Verlag, 1994

ou

Phillip Rogaway et Don Coppersmith *A Software-Optimized Encryption Algorithm*, Journal of Cryptology, volume 11, number 4, Springer International 1998

Ron Rivest *The RC4 Encryption Algorithm*, RSA Data Security, 1992

Chiffrements par produit et itératif

Définition

Un *chiffrement par produit* combine deux (ou plus de deux) transformations de manière à ce que le chiffrement résultant soit plus sûr que les transformations individuelles

Définition

Un *réseau de substitutions et transpositions* est un chiffrement par produit composé d'étapes impliquant des substitutions et des transpositions

Définition

Un *chiffrement itératif par blocs* est un chiffrement par blocs impliquant une répétition séquentielle d'une fonction. Ce chiffrement est paramétrisé par son nombre r de tours (nombre d'itérations), la taille n des blocs et la taille k de la clé dont on dérive r sous-clés k_i . Ces sous-clés k_i paramétriseront, à leur tour, la fonction à chaque itération

Chiffrement de Feistel

Définition

Un *chiffrement de Feistel* est un chiffrement itératif appliquant un message clair de $2t$ bits (avec t bits de gauche L_0 et t bits de droite R_0) vers un message chiffré de même taille (L_r et R_r) après r tours où $r \geq 1$. Pour $1 \leq i \leq r$, le tour i applique L_{i-1} et R_{i-1} vers L_i et R_i en utilisant la sous-clé k_i de la manière suivante :

$$L_i = R_{i-1}$$

et

$$R_i = L_{i-1} \oplus f(R_{i-1}, k_i)$$

Le déchiffrement se réalise en effectuant le même procédé mais en utilisant les sous-clés en ordre inverse (de k_r à k_1)

DES : l'algorithme

DES chiffre un bloc de texte clair de 64 bits en utilisant une clé de 56 bits pour obtenir un bloc de texte chiffré de 64 bits

1. A partir d'un bloc de texte clair x , une chaîne de bits x_0 est construite en changeant l'ordre des bits de x suivant une permutation initiale IP : $x_0 = IP(x) = L_0R_0$ où L_0 contient les 32 premiers bits de x_0 et R_0 les 32 suivants
2. L'algorithme opère 16 itérations d'une fonction f où $L_i = R_{i-1}$ et $R_i = L_{i-1} \oplus f(R_{i-1}, k_i)$ où k_1, \dots, k_{16} sont des sous-clés de 48 bits composées à partir de la clé principale k
3. Une permutation inverse IP^{-1} est appliquée à $R_{16}L_{16}$ pour obtenir le texte chiffré y de 64 bits

DES : la fonction f et les SBoxes

La fonction f accepte une chaîne, A , de 32 bits ainsi qu'une sous-clé, J , de 48 bits, le résultat de cette fonction est sur 32 bits. La fonction f réalise les 4 étapes suivantes :

1. A est *augmenté* pour passer de 32 à 48 bits au moyen d'une fonction d'expansion qui modifie l'ordre des bits de A et qui en dédouble certain. Nous notons le résultat $E(A)$
2. $B = E(A) \oplus J$ est calculé, puis ce résultat B est décomposé en 8 sous-chaînes, B_i , de 6 bits chacune

DES : f et SBoxes (suite)

3. Chacun des 8 B_i est modifié par une « sbox » différente : $C_i = S_i(B_i)$, $1 \leq i \leq 8$ où C_i contient 4 bits. Chaque sbox est un tableau 4×16 d'entiers compris entre 0 et 15. Les bits b_1b_6 de B_i forment le numéro (en binaire) de l'indice de ligne de la table et les bits $b_2b_3b_4b_5$ forment l'indice de colonne de la table. A l'endroit indiqué se trouve les 4 bits résultats. Ces boîtes sont des fonctions de substitution
4. Les 8 C_i forment la chaîne C qui est réordonnée suivant la permutation P

DES : la clé et les sous-clés

Au départ la clé k est codée sur 64 bits, mais seul 56 bits sont significatifs. 8 bits sont présents dans la clé en tant que bits de détection d'erreur

Les bits 8, 16, 24, 32, 40, 48, 56 et 64 sont des bits de parité positionnés de manière à ce que l'octet auquel ils appartiennent contiennent un nombre impair de 1

Ces 8 bits ne sont plus considérés par la suite

DES : la clé et les sous-clés (suite)

La diversification de la clé se réalise ainsi :

1. Les 56 bits de la clé sont ordonnés suivant la permutation $PC1$: $PC1(k) = C_0D_0$ où C_0 est composé des 28 premiers bits et D_0 des 28 suivants

2. Pour tout i entre 1 et 16, on calcule :

$C_i = LS_i(C_{i-1})$ et $D_i = LS_i(D_{i-1})$ où LS_i est une rotation circulaire gauche d'une position si $i = 1, 2, 9$ ou 16 et une rotation circulaire gauche de deux positions sinon

Puis $k_i = PC2(C_iD_i)$ où $PC2$ est une autre permutation qui fournit un résultat sur 48 bits

DES : clés faibles

DES possède 4 clés faibles, à savoir qui sont telles que $\forall x \in M$ nous avons $E_k(E_k(x)) = x$

DES possède 6 paires de clés semi-faibles, à savoir des paires (k_1, k_2) qui sont telles que $\forall x \in M$ nous avons $E_{k_1}(E_{k_2}(x)) = x$

De plus pour les 4 clés faibles de DES, il existe 2^{32} messages $x \in M$ qui sont tels que $E_k(x) = x$

Caractéristiques des bloc ciphers

Altérer un bit d'un bloc de texte clair doit altérer chaque bit du bloc de texte chiffré correspondant avec un probabilité égale à $\frac{1}{2}$

Altérer un bit d'un bloc de texte chiffré doit altérer chaque bit du bloc de texte clair déchiffré correspondant avec un probabilité égale à $\frac{1}{2}$

Altérer un bit de la clé doit altérer chaque bit des blocs du texte chiffré correspondant avec un probabilité égale à $\frac{1}{2}$

Empiriquement DES vérifie ces propriétés

DES : la cryptanalyse différentielle

Méthode de cryptanalyse inventée par Biham et Shamir. Cette attaque est à texte clair choisi

Cette cryptanalyse s'intéresse à la comparaison entre le ou-exclusif de deux textes clairs et le ou-exclusif de deux textes chiffrés correspondants

Nous notons L_0R_0 le premier texte clair et $L_0^*R_0^*$ le deuxième. Nous notons aussi : $L'_0R'_0 = L_0R_0 \oplus L_0^*R_0^*$

Nous généralisons ces notations avec $L_i, L_i^*, L'_i \dots$ correspondant à chaque itération de l'algorithme

Cryptanalyse différentielle (suite)

La méthode d'attaque utilise des caractéristiques qui sont des L'_i et R'_i associés à une probabilité d'existence connaissant L'_{i-1} et R'_{i-1}

L'évolution des caractéristiques à travers l'algorithme permet de déterminer de manière probabiliste des bits de la clé

E. Biham et A. Shamir *Differential cryptanalysis of the Data Encryption Standard*, Springer-Verlag, 1993

Autres

Cryptanalyse linéaire de DES

Mitsuru Matsui *Linear cryptanalysis method for DES cipher*, Eurocrypt'93, Lecture notes in computer science, volume 765, Springer-Verlag 1994

IDEA

X. Lai et J. Massey *A proposal for a new block encryption standard*, Eurocrypt'90, Lecture notes in computer science, volume 473, Springer-Verlag 1991

RC5

R. Rivest *The RC5 encryption algorithm*, Fast Software Encryption, Lecture note in computer science, volume 1008, Springer-Verlag 1995

AES

Advanced Encryption Standard

Sur les 10 candidats au départ, il en reste 5 :

- RC6 (RSA Lab, USA)
- Rijndael (UE)
- Twofish (USA)
- Serpent (UE)
- MARS (IBM, USA)

<http://csrc.nist.gov/encryption/aes>