

Introduction à la cryptographie

Sébastien VARRETTE

Université du Luxembourg - Laboratoire LACS, LUXEMBOURG

CNRS/INPG/INRIA/UJF - Laboratoire LIG-IMAG

Sebastien.Varrette@imag.fr

<http://www-id.imag.fr/~svarrett/>



Cours "Cryptographie & Sécurité Réseau" Master Info
Université de Yaoundé

Déroulement du cours

- Cours intensif ($\sim 40h$) sur 2 semaines (17-28 avril)
- Objectif du cours
 - Introduction/Sensibilisation à la cryptographie
 - Cryptographie à clé secrète
 - Cryptographie à clé publique
 - Programmation en C de divers algorithmes
 - Signatures Electroniques
 - Architectures PKI
 - Sécurité Systèmes & Réseaux
 - Programmation sécurisée en C
 - Manipulation sur machine (environnement Linux)

Quelques références bibliographiques. . . (avant que j'oublie)



MENEZES A. J., VANSTONE S. A. and OORSCHOT P. C. V., *Handbook of Applied Cryptography*, Computer Sciences Applied Mathematics Engineering, CRC Press, Inc., 1st edition, 1996,

<http://www.cacr.math.uwaterloo.ca/hac/>



SCHNEIER B., *"Cryptographie Appliquée"*, Vuibert, Wiley and International Thomson Publishing, NY, 2nd edition, 1997.

<http://www.schneier.com/book-applied.html>



STINSON D.R., *Cryptography : Theory and Practice*, Chapman & Hall/CRC Press, 2nd edition, 2002.

<http://www.cacr.math.uwaterloo.ca/~dstinson/CTAP2/CTAP2.html>



EBRAHIMI T., LEPRÉVOST F. and WARUSFELD Ed., *Cryptographie et Sécurité des systèmes et réseaux*, Hermes/Lavoisier,

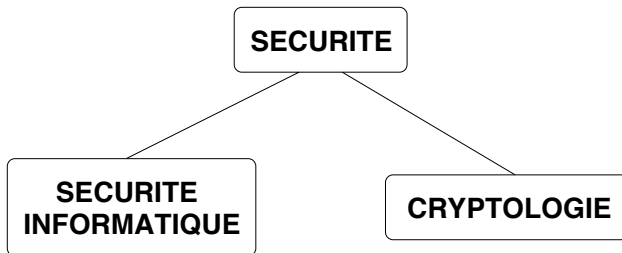
http://www-id.imag.fr/~svarrett/book_secu_mult.html

Plan

- 1 Principes généraux de la cryptographie
- 2 Cryptographie à clé secrète
- 3 Cryptographie à clé publique
- 4 Fonctions de hachage et signatures électroniques

Principes généraux de la cryptographie

Notion de *Sécurité*



Notion de *Cryptologie*

"Science du secret" avec deux composantes complémentaires

- 1 la **cryptographie** : étude et conception des procédés de chiffrement des informations
- 2 la **cryptanalyse** : analyse des textes chiffrés pour retrouver les informations dissimulées

Notion de *Cryptologie*

"Science du secret" avec deux composantes complémentaires

- ① la **cryptographie** : étude et conception des procédés de chiffrement des informations
- ② la **cryptanalyse** : analyse des textes chiffrés pour retrouver les informations dissimulées

- Bien distinguer cryptographie/stéganographie :
 - *cryptographie* : transforme un message clair en cryptogramme
 - *stéganographie* : dissimule l'existence même de l'information secrète (encre sympathique etc...)

Je suis très émue de vous dire que j'ai bien compris, l'autre jour, que vous avez toujours une envie folle de me faire danser. Je garde un souvenir de votre baiser et je voudrais que ce soit là une preuve que je puisse être aimée par vous.[...]

Illustration de l'utilisation de la stéganographie : extrait d'une lettre de Georges Sand.

Un mot sur la stéganographie...

- Information non-chiffrée

Connaissance de l'existence de l'information

=

Connaissance de l'information

- Exemples :

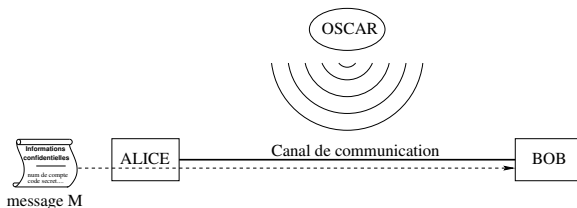
- Message couvert : tablette couverte de cire, crâne du messenger
- Message invisible : encre sympathique (Pline 1er siècle av. JC)
- Message illisible : Micro-film sous forme de point
- Message subliminal : traitement de texte des ministres de M. Thatcher

- Théorie :

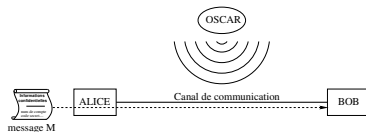
- Faible niveau de sécurité
- En pratique ça marche ...(11/09/2001 ?)
- Utilisée également pour le Watermarking (JPEG, MP3-MPEG, etc)

Terminologie (1)

- Protagonistes traditionnels :
 - **Alice** et **Bob** : souhaitent se transmettre des informations
 - **Oscar** : un opposant qui souhaite espionner Alice et Bob
- Objectif fondamental de la cryptographie
 - permettre à Alice et Bob de communiquer sur un canal peu sûr
 - Oscar ne doit pas comprendre ce qui est échangé.

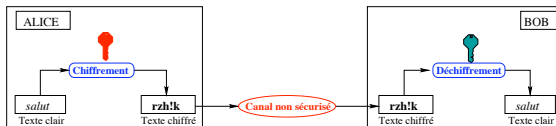


Terminologie (2)



- **Texte clair** : information qu'Alice souhaite transmettre à Bob
 - Ex : texte en français, donnée numérique etc...
- **Chiffrement** : processus de transformation d'un message M de telle manière à le rendre incompréhensible
 - Basé sur une *fonction de chiffrement* E
 - On génère ainsi un **message chiffré** $C = E(M)$
- **Déchiffrement** : processus de reconstruction du message clair à partir du message chiffré
 - Basé sur une fonction de déchiffrement D
 - On a donc $D(C) = D(E(M)) = M$ (D et E sont injectives)

Algorithmes de cryptographie



- Propriétés théoriques nécessaires :

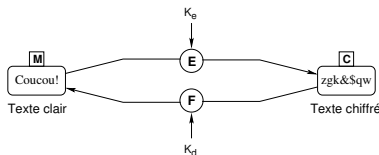
- ① Confusion

- Aucune propriété statistique ne peut être déduite du message chiffré

- ② Diffusion

- Toute modification du message en clair se traduit par une modification complète du chiffré

Relation fondamentale



- En pratique : E et D sont paramétrées par des clés K_e et K_d

$$\begin{cases} E_{K_e}(M) = C \\ D_{K_d}(C) = M \end{cases} \quad (1)$$

- $K_e, K_d \in \{\text{espace des clés}\}$.
- Définit deux catégories de systèmes cryptographiques :
 - 1 Systèmes à clé secrète (ou symétriques) ($K_e = K_d = K$)
 - 2 Systèmes à clé publique (ou asymétriques) ($K_e \neq K_d$)

Exemple : representation mathématique de E et D

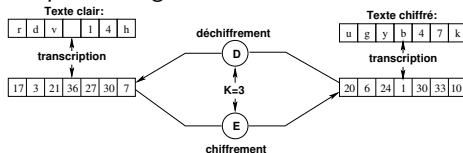
- Pour permettre l'analyse des systèmes cryptographiques
 - représentation mathématique des messages M et C . Ex :

a	b	...	y	z	0	1	...	9	□	.
0	1	...	24	25	26	27	...	35	36	37

Ici : vocabulaire de $n = 38$ caractères. (code ASCII : $n=256$)

- fonctions E et D vues comme des fonctions mathématiques
 - basée le plus souvent sur l'arithmétique modulaire
 - Exemple : chiffrement par décalage

$$\begin{cases} E_K(M) = M + K \mod n \\ D_K(C) = C - K \mod n \end{cases}$$



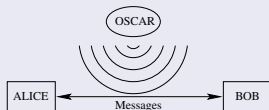
A quoi sert la cryptographie ?

CAIN (Confidentialité - Authentification - Intégrité - Non-répudiation)

- **Confidentialité** des informations stockées/manipulées
 - utilisation d'un algorithme de chiffrement.
 - *empêcher l'accès* aux infos pour ceux qui ne sont pas autorisés.
- **Authentification** d'utilisateurs/de ressources
 - utilisation d'algorithmes d'authentification.
 - Alice s'identifie à Bob en prouvant qu'elle connaît un secret S , (ex : un mot de passe).
- **Intégrité** des informations stockées/manipulées
 - vérifier que les infos transmises n'ont pas subi d'*altérations*
- **Non-répudiation** des informations
 - utilisation d'algorithmes de signatures
 - empêcher un utilisateur de se dédire

Les grands types de menaces

Menaces passives

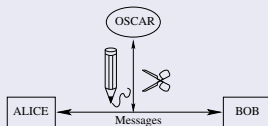


- Oscar ne fait qu'écouter le message.
- menace la *confidentialité*

- une information sensible parvient également à une autre personne que son destinataire légitime.

Les grands types de menaces (2)

Menace active



- Oscar peut modifier le contenu des messages échangés.
- menace l'*intégrité* de l'information.

- Exemple d'attaques actives :

- l'usurpation d'identité (de l'émetteur ou du receuteur)
- l'altération/modification du contenu des messages ;
- la destruction de messages/ le retardement de la transmission ;
- la répétition de messages (jusqu'à engorgement)
- la répudiation de message : l'émetteur nie avoir envoyé le message.

Les attaques sur un chiffrement

- *Cryptanalyse* : étude de la sécurité des procédés de chiffrement utilisés en cryptographie
- Niveaux d'attaques possibles :
 - 1 *Texte chiffré connu* : Seul C est connu d'Oscar
 - 2 *Texte clair connu* : Oscar connaît C et M correspondant
 - 3 *Texte clair choisi* : $\forall M$, Oscar peut obtenir C
 - 4 *Texte chiffré choisi* : $\forall C$, Oscar peut obtenir M
- garantir la confidentialité \implies Oscar ne peut pas :
 - trouver M à partir de $E(M)$
 - trouver la méthode de déchiffrement D à partir d'une séquence $\{M_i, E(M_i)\}$.

Algorithmes d'attaques

1 Attaque brutale

- Énumérer toutes les valeurs possibles de clefs
- 64 bits $\implies 2^{64}$ clefs = $1.844 * 10^{19}$ combinaisons
 - Un milliard de combinaisons/s \Rightarrow 1 an sur 584 machines

2 Attaque par séquences connues

- Deviner la clef si une partie du message est connue
ex : en-têtes de standard de courriels

3 Attaque par séquences forcées

- Faire chiffrer par la victime un bloc dont l'attaquant connaît le contenu, puis on applique l'attaque précédente ...

4 Attaque par analyse différentielle

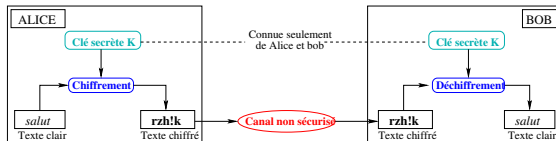
- Utiliser les faibles différences entre plusieurs messages (ex : logs) pour deviner la clef

Bref historique des codes secrets...

- Cryptographie Ancienne
 - Transposition Sparte (5ème siècle av JC)
 - Substitution César (1er siècle av JC), Vigenère (XVI ème)
- Cryptanalyse des codes mono et poly alphabétiques
 - El Kindi (IXème siècle)
 - Babbage/Kasiski (XIXème siècle)
- Mécanisation de la cryptographie et de la cryptanalyse
 - Enigma (1918)
- Vers un chiffrement parfait : Vernam, théorie de l'information
- Standard de chiffrement à clé secrète : DES (1977), AES(2000)
- Cryptographie à clé publique (1976)

Cryptographie à clé secrète

Principe



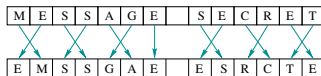
- $K_e = K_d = K$ (clé privée convenue secrètement par Alice et Bob)
 - En pratique : grande efficacité en terme de temps de calculs
 - Inconvénient : la clé K doit rester secrète.
- **Analogie : coffre-fort !**
- Historiquement le premier type de chiffrement utilisé.
- Fourni le seul chiffrement théoriquement indéchiffrable
 - Chiffrement de Vernam (ou one-time password)
 - Démonstration du mathématicien Claude Shannon (1949)

Chiffrement symétrique : outils de base utilisés (1)

- A la base des chiffrements à clé secrète :
 - Substitution : **remplacer** chaque élément par un autre.



- Transposition (ou permutation) : **changer** l'ordre des éléments



Chiffrement symétrique : outils de base utilisés (2)

- Autres opérations utiles :
 - Arithmétique modulaire dans \mathbb{Z}_n $a, b, n \in \mathbb{N}$, avec $n \geq 2$.

$$a \equiv b \pmod{n} \iff n \text{ divise } a - b$$

En pratique : $b =$ reste de la division euclidienne de a par n .

$$5 \equiv 1 \pmod{4} \text{ et } -3 \equiv 125 \pmod{128}$$

- Notions \pm associées : Primalité, Euclide, Th. des restes chinois, Gauss, Euler...
- opération XOR (ou exclusif \oplus)

\oplus	0	1
0	0	1
1	1	0

- Opération bijective (bijection inverse : \oplus !)
- correspond à une addition bit-à-bit modulo 2.

Les premiers procédés

Initialement, le secret échangé était la technique mise en oeuvre

- 400 av JC : esclave envoyé à Aristogoras par Histaius
- Ve av JC : premières transpositions monoalphabétiques
 - Chiffrement de type anagramme : mélange les lettres du message
 - Confusion sur la syntaxe mais chaque lettre conserve sa valeur
 - Clé de chiffrement complexe
 - Principe des scytales spartiate (coms entre chefs des armées)



Les premiers procédés (2)

- IVe : premières substitution
 - Chiffrement par changement d'alphabet
 - Ex : Kama-sutra : recommande aux femmes de maîtriser le *mlecchita-vikalpà* (art de l'écriture secrète)
- 150 avant JC : carré de Polybe

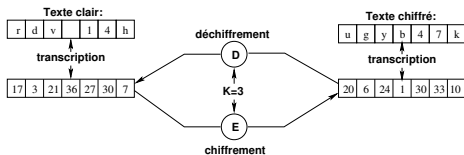
	1	2	3	4	5
1	a	b	c	d	e
2	f	g	h	i	j
3	k	l	m	n	o
4	p	q	r	s	t
5	u	v	x	y	z

- alphabet de 25 lettres (pas de 'w' ou i et j regroupés)
- remplace les lettres par des chiffres
- codage d'une lettre = coordonnée dans le tableau
 - A \longrightarrow 11
 - B \longrightarrow 12 ...

Le chiffrement de César...

- Chiffrement par décalage avec $K = 3$.

$$\begin{cases} E_K(M) = M + K \mod n \\ D_K(C) = C - K \mod n \end{cases}$$



- Seulement n façons différentes de chiffrer un message
 - code très peu sûr (recherche exhaustive facile)
 - avantage de la simplicité
 - employé par les officiers sudistes (guerre de Sécession)
 - réemployé sur les forums de News : ROT-13 ($K = 13$)
- Généralisation : chiffrement affine
 - $E_{(a,b)}(M) = a * M + b \mod n$ (pour $a \in \mathbb{Z}_n^\times$)
 - cf TD

Cryptanalyse des substitutions mono-alphabétique

- Rappel : substitution mono-alphabétique : on remplace chaque lettre par une lettre différente.

M	E	S	S	A	G	E		S	E	C	R	E	T
↓	↓	↓	↓	↓	↓	↓		↓	↓	↓	↓	↓	↓
N	F	T	T	B	H	F		T	F	D	S	F	U

- Nombre de possibilités (alphabet de 26 lettres) ?

Cryptanalyse des substitutions mono-alphabétique

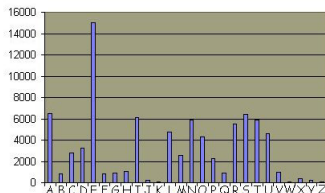
- Rappel : substitution mono-alphabétique : on remplace chaque lettre par une lettre différente.

M	E	S	S	A	G	E		S	E	C	R	E	T
N	F	T	T	B	H	F		T	F	D	S	F	U

- Nombre de possibilités (alphabet de 26 lettres) ?
 - chiffrement de 'A' : 26 possibilités
 - chiffrement de 'B' : 25 possibilités
 - ... $\rightarrow 26! \simeq 4 * 10^{26}$ possibilités
 - Ordre de grandeur de comparaison : plier 50 fois sur elle-même une feuille de papier (épaisseur : 1 dixième de mm)
 - \rightarrow épaisseur de la feuille :
 2^{50} dixième de millimètre $\simeq 1,1 * 10^8$ km
(110 millions de km $\simeq 300$ fois distance Terre/Lune)

Cryptanalyse des substitutions mono-alphabétique

- MAIS ne cache pas la fréquence d'apparition des symboles !



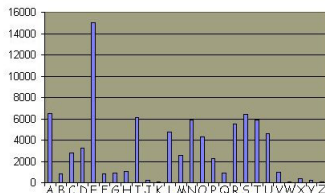
En français, la lettre 'e' apparaît le plus souvent etc...

- Exemple : cryptanalyse du texte suivant :

HQYRBHU GX UHQIRUW DYHF GHV DUPHV

Cryptanalyse des substitutions mono-alphabétique

- MAIS ne cache pas la fréquence d'apparition des symboles !



En français, la lettre 'e' apparaît le plus souvent etc...

- Exemple : cryptanalyse du texte suivant :

HQYRBHU GX UHQIRUW DYHF GHV DUPHV

- Réponse : envoyer du renfort avec des armes
- cryptanalyse proposée par Al Kindi (un savant arabe) au IX^e s.
- cf TD

Exemple de chiffrement par code

Chiffrement par code = chiffrement de mots plutôt que de lettres

- Ex : Code de Marie Stuart (XVI^e s.) (nomenclateur)
 - Décapitée pour avoir utilisé un code trop faible quand elle tenta d'assassiner la Reine d'Angleterre.
 - Illustration d'une attaque active
 - Walsingham fait ajouter un P.S aux lettres interceptées pour obtenir les noms des conspirateurs
 - Trop confiante dans son code, Marie Stuart le fit



a b c d e f g h i k l m n o p q r s t u x y z
o t 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32

Nulles ff. — . — . d .

Dowbleth o

and for with that if but where as of the from by

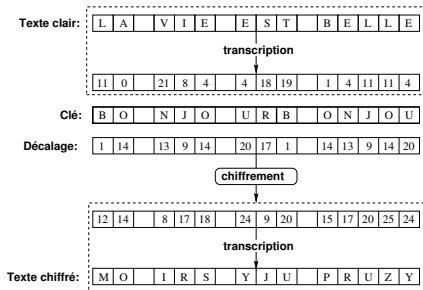
2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32

so not when there this in wich is what say me my wyrt

33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000 1001 1002 1003 1004 1005 1006 1007 1008 1009 1010 1011 1012 1013 1014 1015 1016 1017 1018 1019 1020 1021 1022 1023 1024 1025 1026 1027 1028 1029 1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 1040 1041 1042 1043 1044 1045 1046 1047 1048 1049 1050 1051 1052 1053 1054 1055 1056 1057 1058 1059 1060 1061 1062 1063 1064 1065 1066 1067 1068 1069 1070 1071 1072 1073 1074 1075 1076 1077 1078 1079 1080 1081 1082 1083 1084 1085 1086 1087 1088 1089 1090 1091 1092 1093 1094 1095 1096 1097 1098 1099 1100 1101 1102 1103 1104 1105 1106 1107 1108 1109 1110 1111 1112 1113 1114 1115 1116 1117 1118 1119 1120 1121 1122 1123 1124 1125 1126 1127 1128 1129 1130 1131 1132 1133 1134 1135 1136 1137 1138 1139 1140 1141 1142 1143 1144 1145 1146 1147 1148 1149 1150 1151 1152 1153 1154 1155 1156 1157 1158 1159 1160 1161 1162 1163 1164 1165 1166 1167 1168 1169 1170 1171 1172 1173 1174 1175 1176 1177 1178 1179 1180 1181 1182 1183 1184 1185 1186 1187 1188 1189 1190 1191 1192 1193 1194 1195 1196 1197 1198 1199 1200 1201 1202 1203 1204 1205 1206 1207 1208 1209 1210 1211 1212 1213 1214 1215 1216 1217 1218 1219 1220 1221 1222 1223 1224 1225 1226 1227 1228 1229 1230 1231 1232 1233 1234 1235 1236 1237 1238 1239 1240 1241 1242 1243 1244 1245 1246 1247 1248 1249 1250 1251 1252 1253 1254 1255 1256 1257 1258 1259 1260 1261 1262 1263 1264 1265 1266 1267 1268 1269 1270 1271 1272 1273 1274 1275 1276 1277 1278 1279 1280 1281 1282 1283 1284 1285 1286 1287 1288 1289 1290 1291 1292 1293 1294 1295 1296 1297 1298 1299 1300 1301 1302 1303 1304 1305 1306 1307 1308 1309 1310 1311 1312 1313 1314 1315 1316 1317 1318 1319 1320 1321 1322 1323 1324 1325 1326 1327 1328 1329 1330 1331 1332 1333 1334 1335 1336 1337 1338 1339 1340 1341 1342 1343 1344 1345 1346 1347 1348 1349 1350 1351 1352 1353 1354 1355 1356 1357 1358 1359 1360 1361 1362 1363 1364 1365 1366 1367 1368 1369 1370 1371 1372 1373 1374 1375 1376 1377 1378 1379 1380 1381 1382 1383 1384 1385 1386 1387 1388 1389 1390 1391 1392 1393 1394 1395 1396 1397 1398 1399 1400 1401 1402 1403 1404 1405 1406 1407 1408 1409 1410 1411 1412 1413 1414 1415 1416 1417 1418 1419 1420 1421 1422 1423 1424 1425 1426 1427 1428 1429 1430 1431 1432 1433 1434 1435 1436 1437 1438 1439 1440 1441 1442 1443 1444 1445 1446 1447 1448 1449 1450 1451 1452 1453 1454 1455 1456 1457 1458 1459 1460 1461 1462 1463 1464 1465 1466 1467 1468 1469 1470 1471 1472 1473 1474 1475 1476 1477 1478 1479 1480 1481 1482 1483 1484 1485 1486 1487 1488 1489 1490 1491 1492 1493 1494 1495 1496 1497 1498 1499 1500 1501 1502 1503 1504 1505 1506 1507 1508 1509 1510 1511 1512 1513 1514 1515 1516 1517 1518 1519 1520 1521 1522 1523 1524 1525 1526 1527 1528 1529 1530 1531 1532 1533 1534 1535 1536 1537 1538 1539 1540 1541 1542 1543 1544 1545 1546 1547 1548 1549 1550 1551 1552 1553 1554 1555 1556 1557 1558 1559 1560 1561 1562 1563 1564 1565 1566 1567 1568 1569 1570 1571 1572 1573 1574 1575 1576 1577 1578 1579 1580 1581 1582 1583 1584 1585 1586 1587 1588 1589 1590 1591 1592 1593 1594 1595 1596 1597 1598 1599 1600 1601 1602 1603 1604 1605 1606 1607 1608 1609 1610 1611 1612 1613 1614 1615 1616 1617 1618 1619 1620 1621 1622 1623 1624 1625 1626 1627 1628 1629 1630 1631 1632 1633 1634 1635 1636 1637 1638 1639 1640 1641 1642 1643 1644 1645 1646 1647 1648 1649 1650 1651 1652 1653 1654 1655 1656 1657 1658 1659 1660 1661 1662 1663 1664 1665 1666 1667 1668 1669 1670 1671 1672 1673 1674 1675 1676 1677 1678 1679 1680 1681 1682 1683 1684 1685 1686 1687 1688 1689 1690 1691 1692 1693 1694 1695 1696 1697 1698 1699 1700 1701 1702 1703 1704 1705 1706 1707 1708 1709 1710 1711 1712 1713 1714 1715 1716 1717 1718 1719 1720 1721 1722 1723 1724 1725 1726 1727 1728 1729 1730 1731 1732 1733 1734 1735 1736 1737 1738 1739 1740 1741 1742 1743 1744 1745 1746 1747 1748 1749 1750 1751 1752 1753 1754 1755 1756 1757 1758 1759 1760 1761 1762 1763 1764 1765 1766 1767 1768 1769 1770 1771 1772 1773 1774 1775 1776 1777 1778 1779 1780 1781 1782 1783 1784 1785 1786 1787 1788 1789 1790 1791 1792 1793 1794 1795 1796 1797 1798 1799 1800 1801 1802 1803 1804 1805 1806 1807 1808 1809 1810 1811 1812 1813 1814 1815 1816 1817 1818 1819 1820 1821 1822 1823 1824 1825 1826 1827 1828 1829 1830 1831 1832 1833 1834 1835 1836 1837 1838 1839 1840 1841 1842 1843 1844 1845 1846 1847 1848 1849 1850 1851 1852 1853 1854 1855 1856 1857 1858 1859 1860 1861 1862 1863 1864 1865 1866 1867 1868 1869 1870 1871 1872 1873 1874 1875 1876 1877 1878 1879 1880 1881 1882 1883 1884 1885 1886 1887 1888 1889 1890 1891 1892 1893 1894 1895 1896 1897 1898 1899 1900 1901 1902 1903 1904 1905 1906 1907 1908 1909 1910 1911 1912 1913 1914 1915 1916 1917 1918 1919 1920 1921 1922 1923 1924 1925 1926 1927 1928 1929 1930 1931 1932 1933 1934 1935 1936 1937 1938 1939 1940 1941 1942 1943 1944 1945 1946 1947 1948 1949 1950 1951 1952 1953 1954 1955 1956 1957 1958 1959 1960 1961 1962 1963 1964 1965 1966 1967 1968 1969 1970 1971 1972 1973 1974 1975 1976 1977 1978 1979 1980 1981 1982 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022 2023 2024 2025 2026 2027 2028 2029 2030 2031 2032 2033 2034 2035 2036 2037 2038 2039 2040 2041 2042 2043 2044 2045 2046 2047 2048 2049 2050 2051 2052 2053 2054 2055 2056 2057 2058 2059 2060 2061 2062 2063 2064 2065 2066 2067 2068 2069 2070 2071 2072 2073 2074 2075 2076 2077 2078 2079 2080 2081 2082 2083 2084 2085 2086 2087 2088 2089 2090 2091 2092 2093 2094 2095 2096 2097 2098 2099 2100 2101 2102 2103 2104 2105 2106 2107 2108 2109 2110 2111 2112 2113 2114 2115 2116 2117 2118 2119 2120 2121 2122 2123 2124 2125 2126 2127 2128 2129 2130 2131 2132 2133 2134 2135 2136 2137 2138 2139 2140 2141 2142 2143 2144 2145 2146 2147 2148 2149 2150 2151 2152 2153 2154 2155 2156 2157 2158 2159 2160 2161 2162 2163 2164 2165 2166 2167 2168 2169 2170 2171 2172 2173 2174 2175 2176 2177 2178 2179 2180 2181 2182 2183 2184 2185 2186 2187 2188 2189 2190 2191 2192 2193 2194 2195 2196 2197 2198 2199 2200 2201 2202 2203 2204 2205 2206 2207 2208 2209 2210 2211 2212 2213 2214 2215 2216 2217 2218 2219 2220 2221 2222 2223 2224 2225 2226 2227 2228 2229 2230 2231 2232 2233 2234 2235 2236 2237 2238 2239 2240 2241 2242 2243 2244 2245 2246 2247 2248 2249 2250 2251 2252 2253 2254 2255 2256 2257 2258 2259 2260 2261 2262 2263 2264 2265 2266 2267 2268 2269 2270 2271 2272 2273 2274 2275 2276 2277 2278 2279 2280 2281 2282 2283 2284 2285 2286 2287 2288 2289 2290 2291 2292 2293 2294 2295 2296 2297 2298 2299 2300 2301 2302 2303 2304 2305 2306 2307 2308 2309 2310 2311 2312 2313 2314 2315 2316 2317 2318 2319 2320 2321 2322 2323 2324 2325 2326 2327 2328 2329 2330 2331 2332 2333 2334 2335 2336 2337 2338 2339 2340 2341 2342 2343 2344 2345 2346 2347 2348 2349 2350 2351 2352 2353 2354 2355 2356 2357 2358 2359 2360 2361 2362 2363 2364 2365 2366 2367 2368 2369 2370 2371 2372 2373 2374 2375 2376 2377 2378 2379 2380 2381 2382 2383 2384 2385 2386 2387 2388 2389 2390 2391 2392 2393 2394 2395 2396 2397 2398 2399 2400 2401 2402 2403 2404 2405 2406 2407 2408 2409 2410 2411 2412 2413 2414 2415 2416 2417 2418 2419 2420 2421 2422 2423 2424 2425 2426 2427 2428 2429 2430 2431 2432 2433 2434 2435 2436 2437 2438 2439 2440 2441 2442 2443 2444 2445 2446 2447 2448 2449 2450 2451 2452 2453 2454 2455 2456 2457 2458 2459 2460 2461 2462 2463 2464 2465 2466 2467 2468 2469 2470 2471 2472 2473 2474 2475 2476 2477 2478 2479 2480 2481 2482 2483 2484 2485 2486 2487 2488 2489 2490 2491 2492 2493 2494 2495 2496 2497 2498 2499 2500 2501 2502 2503 2504 2505 2506 2507 2508 2509 2510 2511 2512 2513 2514 2515 2516 2517 2518 2519 2520 2521 2522 2523 2524 2525 2526 2527 2528 2529 2530 2531 2532 2533 2534 2535 2536 2537 2538 25

La cryptographie par substitution polyalphabétique (1)

- Méthode utilisée par Vigenère (1586)
 - la clef définit le décalage pour chaque lettre du message



'A' : décalage de 0

'B' : décalage de 1

...

'Z' : décalage de 25

Ex : chiffrement de
"La vie est belle"
avec la clé "bonjour"

La cryptographie par substitution polyalphabétique (2)

- Procédé de Vigenère résistera jusqu'au milieu du XIX^e s.
 - Cryptanalyse de Babbage (1854) et Kasiski (1863)
 - But : se ramener à la cryptanalyse de substitution simple
 - Exemple :

ENVOYER LA CAVALERIE
CLEFCLE FC LEFCLEFCL
—→ GYZTAPD QC NEACWIWKP

- 1 Déterminer la taille de la clé (méthode de Kasiski) : 4

La cryptographie par substitution polyalphabétique (2)

- Procédé de Vigenère résistera jusqu'au milieu du XIX^e s.
 - Cryptanalyse de Babbage (1854) et Kasiski (1863)
 - But : se ramener à la cryptanalyse de substitution simple
 - Exemple :

ENVOYER LA CAVALERIE
CLEFCLE FC LEFCLEFCL
—→ GYZTAPD QC NEACWIWKP

- 1 Déterminer la taille de la clé (méthode de Kasiski) : 4
- 2 On réarrange le cryptogramme par groupe de 4 lettres :

GYZT
APDQ
CNEA
CWIW
KP

La cryptographie par substitution polyalphabétique (2)

- Procédé de Vigenère résistera jusqu'au milieu du XIX^e s.
 - Cryptanalyse de Babbage (1854) et Kasiski (1863)
 - But : se ramener à la cryptanalyse de substitution simple
 - Exemple :

ENVOYER LA CAVALERIE
CLEFCLE FC LEFCLEFCL
—→ GYZTAPD QC NEACWIWKP

- 1 Déterminer la taille de la clé (méthode de Kasiski) : 4
- 2 On réarrange le cryptogramme par groupe de 4 lettres :

GYZT
APDQ
CNEA
CWIW
KP

- 3 Pour chaque colonne, cryptanalyse de substitution simple

Calculer la longueur d'une clé de Vigenère

- Considérons par exemple le message codé suivant :

CS AZZMEQM, CO XRWf, CS DZRM GFMJECV. X'IMOQJ JC LB NLFMK CC LBM WCCZBM
 KFIMSZJSZ CS URQUIOU. CS ZLPi ECZ RMWWTv, SB KCCJ QMJ FCSOVJ GCI ZI ICCS...

- Idee : une séquence se répète → la distance entre 2 séquences est probablement un multiple de la taille de la clef

Séquence	Position	Distance	Décomposition
COX	11-140	129	3.43
FCS	16-99	83	83
ZRM	20-83	63	3 ² 7
FMJ	24-162	138	2.3.23
CLB	37-46	9	3 ²
KCC	44-92	48	2 ³ 3
WTV	87-133	46	2.23
CCJ	93-126	33	3.11
ICC	110-155	45	3 ² .5
MJI	136-163	27	3 ³

pgcd pour les triplets pertinents : 3

LE SILENCE, LA PAIX, LE VIDE PRESQUE. J'AVAI VU UN FURET OU UNE FOUINE TRAVERSER
 LE MACADAM. LE RUBAN QUI DEFILE, ET TOUS CES RUBANS SUR LA ROUTE...

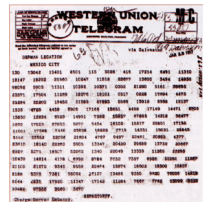
- Méthode moderne (Friedman 1920) : calcul d'indices de coïncidences.

Cryptanalyse classique par
 analyse de fréquence en
 regroupant par paquet de 3

C	S	A
Z	Z	M
E	Q	M
C	O	X
.	.	.
.	.	.
.	.	.

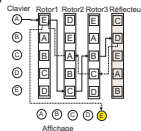
Enjeux de la cryptanalyse : le télégramme de Zimmermann

- 1917 : la guerre s'enlise
- Les Etats-Unis de Wilson sont restés neutre
- L'état-major allemand veut lancer la guerre sous-marine totale
 - Pb : peut déclencher entrée en guerre des USA
 - Idée de Zimmermann (ministre aff. étr.) : occuper les USA avec le Mexique et le Japon
⇒ soutien financier allié à ces insurrections.
 - Z. envoie un télégramme chiffré à l'ambassade d'Allemagne aux USA
- Télégramme déchiffré par le bureau 40 (Montgomery & al.)
- Permet l'entrée en guerre des USA contre l'Allemagne.



Mécanisation de la cryptographie : Enigma

- Machine Enigma (Scherbius 1918)
 - Substitution polyalphabétique
 - 26 orientations pour 3 rotors :
 $26^3 = 17576$ alphabets
 - Réflécteur : cryptage/décryptage : même config
 - Connector/Reflector : substitution



- Brisé par l'équipe polonaise (Marian Rejewski) en 1933.
- Renforcé par les allemands pdt la 2^{eme} guerre (avec 5 rotors)
- Cassé par les "bombes" de Turing (Bletchley) (dico...)

Notion de sécurité inconditionnelle

- Cryptanalyses précédentes utilisent la répétition de la clé

Definition (Sécurité inconditionnelle)

la connaissance du message chiffré n'apporte aucune information sur le message clair.

- seule attaque possible : recherche exhaustive de clé secrète
- la clé secrète doit être au moins aussi longue que le texte clair

Notion de sécurité inconditionnelle

- Cryptanalyses précédentes utilisent la répétition de la clé

Definition (Sécurité inconditionnelle)

la connaissance du message chiffré n'apporte aucune information sur le message clair.

- seule attaque possible : recherche exhaustive de clé secrète
- la clé secrète doit être au moins aussi longue que le texte clair

Existe t il un système cryptographique inconditionnellement sûr ?

Système de Vernam (One time pad) (1917)

- Relation fondamentale :

$$\forall M, K / |M| = |K|, (M \oplus K) \oplus K = M$$

Fonctions de chiffrement/déchiffrement :

$$\begin{cases} E_K(M) = M \oplus K \\ D_K(C) = C \oplus K \end{cases}$$

- Vigenère avec : Longueur mot-clef = longueur message !
 - Confusion totale : chiffrement de "aaaa...aaa" aléatoire
 - Diffusion totale : si le mot -clef n'est jamais réutilisé
- Pour un message M de n bits, clé K de n bits.

$$\begin{array}{rcl} M & = & 1000011 \\ K & = & 1101000 \\ \hline C = M \oplus K & = & 0101011 \end{array}$$

- **Si** K est totalement aléatoire et n'est utilisée une seule fois **alors** Oscar n'obtient aucune information sur M à partir de C .

Systèmes cryptographiques pratiquement sûr

- Vernam : seul système prouvé inconditionnellement sûr
 - MAIS problème du caractère aléatoire et du stockage de K
 - tous les autres systèmes sont théoriquement cassables

Definition (chiffrement pratiquement sûr)

un message chiffré ne permet de retrouver ni la clé secrète ni le message clair *en un temps humainement raisonnable*.

⇒ permet d'utiliser des clés plus petites (56, 128 bits...)

Question : Pourquoi ne pas tester toutes les clés possibles ?

Systèmes cryptographiques pratiquement sûr

- Vernam : seul système prouvé inconditionnellement sûr
 - MAIS problème du caractère aléatoire et du stockage de K
 - tous les autres systèmes sont théoriquement cassables

Definition (chiffrement pratiquement sûr)

un message chiffré ne permet de retrouver ni la clé secrète ni le message clair *en un temps humainement raisonnable*.

⇒ permet d'utiliser des clés plus petites (56, 128 bits...)

Question : Pourquoi ne pas tester toutes les clés possibles ?

Réponse : ce serait trop long a tester sur ordinateur !

Ex : portable 1Ghz → 10^9 op/s ; clé : 128 bits soit
 $2^{128} \simeq 3,4 * 10^{38}$ possibilités ⇒ $3,4 * 10^{29}$ s

Systèmes cryptographiques pratiquement sûr

- Vernam : seul système prouvé inconditionnellement sûr
 - MAIS problème du caractère aléatoire et du stockage de K
 - tous les autres systèmes sont théoriquement cassables

Definition (chiffrement pratiquement sûr)

un message chiffré ne permet de retrouver ni la clé secrète ni le message clair *en un temps humainement raisonnable*.

⇒ permet d'utiliser des clés plus petites (56, 128 bits...)

Question : Pourquoi ne pas tester toutes les clés possibles ?

Réponse : ce serait trop long a tester sur ordinateur !

Ex : portable 1Ghz → 10^9 op/s ; clé : 128 bits soit

$2^{128} \simeq 3,4 * 10^{38}$ possibilités ⇒ $3,4 * 10^{29}$ s

- 10^{22} ordi pdt 1 an (il y a $\simeq 10^9$ PC ds le monde en 2004)
- Age de l'univers : 15 milliard * 365 * 24 * 3600 $\simeq 4,7 * 10^{17}$ s

Cryptographie moderne

Principe de Auguste Kerckhoffs (1883)

- ❶ La sécurité repose sur le secret de la clef et non sur le secret de l'algorithme
 - Canal +, Cartes Bleues
 - Contre-exemple : GSM et surtout CSS (Content Scrambling System - protection des DVD)
 - cf <http://www.lemuria.org/decss/>
- ❷ Le déchiffrement sans la clef doit être impossible
 - en un temps raisonnable
- ❸ Trouver la clef à partir du clair et du chiffré est impossible
 - en un temps raisonnable

Auguste Kerckhoffs, *La cryptographie militaire*, Journal des sciences militaires, vol. IX, pp. 5-83, jan. 1883, pp.

161-191, févr. 1883.

Théorie de l'Information

- Claude Shannon (1948)
- Source d'information $(\mathcal{S}, \mathcal{P})$ sans mémoire
 - $\mathcal{S} = \{s_1, \dots, s_n\}$, $\mathcal{P} = \{p_1, \dots, p_n\}$
 - p_i : probabilité d'occurrence de s_i dans une emission
- Notion d'entropie
 - Quantité d'information de $s_i \in \mathcal{S}$: $I(p_i) = \log_2 \left(\frac{1}{p_i} \right)$
 - Quantité d'information d'une source $(\mathcal{S}, \mathcal{P})$
 - $H(\mathcal{S}) = - \sum_{i=1}^n p_i \log_2(p_i) = \sum_{i=1}^n p_i \log_2 \left(\frac{1}{p_i} \right)$
 - Nombre moyen de question à poser pour déterminer la valeur obtenue
 - Dé non pipé : $H(\mathcal{S}) = \sum_{i=1}^6 \frac{1}{6} \log_2(6) \simeq 2.58$
 - Dé pipé ($p_1 = \frac{1}{2}$; $p_i = \frac{1}{10} \forall i \in [2, 6]$)
 $H(\mathcal{S}) = \frac{1}{2} \log_2(2) + 5 * \frac{1}{10} \log_2(10) \simeq 2,161$
- L'entropie est maximale lorsque toutes les probas sont égales !
 - Ex : Si l'apparition des lettres est exactement aléatoire, il est impossible d'appliquer l'attaque fréquentielle

Théorie de la complexité

- Méthodologie pour analyser la complexité de calcul des algorithmes
 - Complexité en temps (ou en nombre d'opérations)
 - Complexité en mémoire (espace de stockage nécessaire)
- Complexité exprimée comme fonction de la taille du paramètre d'entrée
 - Ex : quicksort : tri moyen en $\mathcal{O}(n \log n)$
- Complexité d'un problème : complexité de l'algorithme permettant de résoudre l'instance la plus difficile
- Classification :
 - Problèmes solubles (en temps polynomial) : classe P
 - Problèmes difficiles (solubles en temps expo) : classe NP
 - Problèmes indécidables

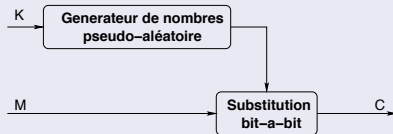
Complexité & Cryptographie

- Niveau de complexité d'une attaque
 - Comparer avec la recherche exhaustive
- Chiffrement idéal (moins que parfait !) - pratiquement sûr
 - L'implémentation est possible : complexité polynomiale au pire
 - Toutes les attaques sont de complexité exponentielle au mieux
- Chiffrement sûr
 - Toutes les attaques connues sont de complexité exponentielle
- Chiffrement pratique
 - Attaquer coûte plus cher (machines,...) que la valeur du secret
 - Attaquer prend plus de temps que la validité du secret
- Attention au *paradoxe des anniversaires* pour les complexités pratique !

Classes de chiffrements symétriques

Chiffrement symétrique par flot (stream cypher - statefull)

- Traitement à la volée ; chiffrement à la one-time pad :
 - $|M| = n$ et avec une petite clé K , générer $K' / |K'| = n$

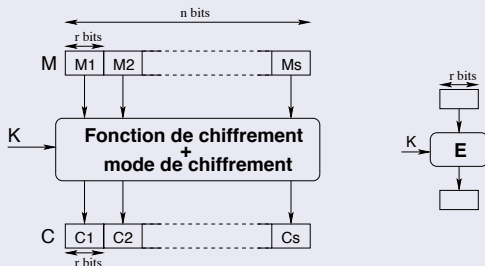


- Sécurité :
 - Substitution rapide (\oplus typiquement)
 - Générateur pseudo aléatoire : "impossible" à prédire
 - Kerckhoffs : la sécurité repose sur le générateur de clé !
- Ex : LFSR, RC4 (Rivest), Py (Biham), E0[Bluetooth], A5/3[GSM]

Classes de chiffrements symétriques (2)

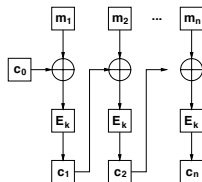
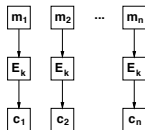
Chiffrement symétrique par bloc (bloc cypher - stateless)

- $M = M_1 \bullet M_2 \bullet \dots \bullet M_s$: s blocs de $r = \frac{n}{s}$ bits



- Sécurité :
 - Pour chaque bloc : $C_i = E_K(M_i)$ dépend de E
 - Pour chaque message : dépend aussi du mode de chiffrement !
- Ex : DES, AES, IDEA, BLOWFISH, RC6

Les modes de chiffrement



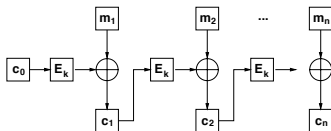
- Mode ECB (Electronic Code Book)

- $C_i = E_K(M_i)$
- $M_i = D_K(C_i)$
- Un bloc est toujours chiffré identiquement
- Aucune sécurité, pas d'utilisation

- Mode CBC (Cipher Block Chaining)

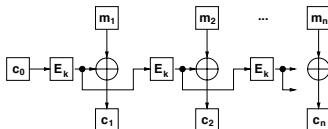
- $C_i = E_K(M_i \oplus C_{i-1})$
- $M_i = C_{i-1} \oplus D_K(C_i)$
- Mode le plus utilisé

Les modes de chiffrement (2)



• Mode CFB (Cipher FeedBack)

- $C_i = M_i \oplus E_K(C_{i-1})$
- $M_i = C_i \oplus E_K(C_{i-1})$
- Pas besoin de D_K !
- Moins sûr, parfois plus rapide
- Utilisé dans les réseaux

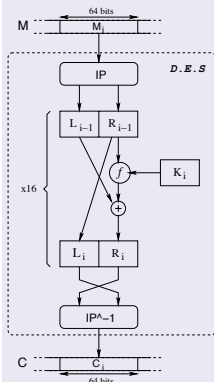


• Mode OFB (Output FeedBack)

- $Z_i = E_K(Z_{i-1})$; $C_i = M_i \oplus Z_i$
- $Z_i = E_K(Z_{i-1})$; $M_i = C_i \oplus Z_i$
- Variante du mode précédent
- Totalement symétrique
- Moins de câblage
- Utilisé dans les satellites

Les standard de chiffrement par bloc

D.E.S (Data Encryption Standard) - 1977



- Standard américain FIPS 46-2
- Chiffrement par blocs de 64 bits
- Clé de 64 bits dont 8 bit de parité :
 - 56 bits effectifs (plaidé par la NSA)
 - Diversification en 16 sous-clés de 48 bits
- Structure générale :
 - Permutation initiale IP
 - 16 rondes "de Feistel" :

$$\begin{cases} L_i = R_{i-1} \\ R_i = L_{i-1} \oplus f(R_{i-1}, K_i) \end{cases}$$

- Permutation finale IP^{-1}

Un mot sur les permutations de DES

IP							
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

IP ⁻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

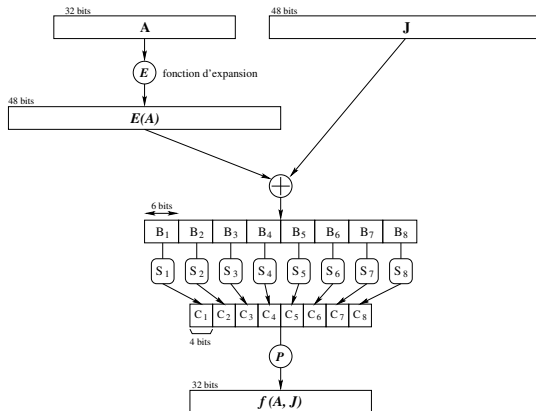
- Signification dans le calcul de $y = IP(x)$:

- le 58^e bit de x est le premier de y
- le 50^e bit de x est le second de y
- etc...

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

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

Détail de la fonction $f(A, J)$ de DES



Détail du calcul autour des S-Box de DES

- 8 "boîtes-S" S_1, S_2, \dots, S_8
 - tableaux 4×16 entiers compris entre 0 et 15

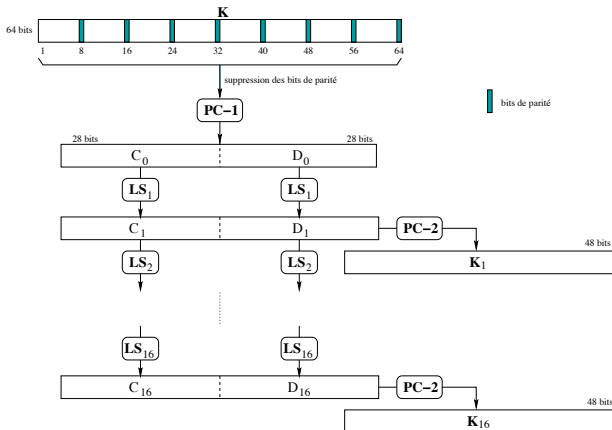
S_1															
14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

- Soit une sous-chaîne de six bits $B_i = b_1.b_2.b_3.b_4.b_5.b_6$.
- Calcul de la chaîne de quatre bits $S_i(B_j)$:
 - $b_1.b_6$ = indice l de S_i à considérer $0 \leq l \leq 3$.
 - $b_2.b_3.b_4.b_5$ = colonne c de S_i à considérer $0 \leq c \leq 15$
 - A l'intersection de l et c : $C_i = S_i(B_i)$!
- On obtient ainsi $C = C_1.C_2 \dots C_8$, chaîne de 32 bits
- On applique la permutation P : $P(C) = f(A, J)$.

Caractéristiques des boîtes S de DES

- Propriétés cryptanalytiques intéressantes
 - Non linéaire (substitution différente de César ou Vernam) : pas d'attaque simple
 - Spécialement conçues pour contrer la cryptanalyse différentielle [Coppersmith 94]
 - Même de très petites modifications des S -box peuvent affaiblir considérablement le chiffre
- A la base de controverse autour de DES
 - cf secret entourant la génération de $\{S_i\}_{1 \leq i \leq 8}$ et de P

Diversification des clés dans DES



Caractéristiques de DES

- Après 5 tours, chaque bit du chiffré dépend de chaque bit du message en clair et de chaque bit de la clef.
- Résultat du chiffrement statistiquement "plat"
- Quelques exemples d'utilisation :
 - Cartes de crédit : *UEPS* (Universal Electronic Payment System)
 - Protocole d'authentification sur réseaux : Kerberos
 - Messagerie électronique : *PEM* (Privacy-Enhanced Mail)
- Implémentation hardware aisée
 - Opération facilement implémentables : cf TP :-)
 - Puce spécifique bas de gamme¹ ($\simeq 60$ euros) : $\simeq 190$ Mo/s.
- Propriété de completion et clés faibles (pas une menace) :

$$DES_K(M) = C \implies DES_{\bar{K}}(\bar{M}) = \bar{C}$$

¹voir par exemple <http://hifn.com/products/7955-7956.html>

Cryptanalyse de DES

- Précalcul exhaustif
 - Stocker le résultat de DES sur un texte choisi $\forall K$
- Recherche exhaustive
 - Chiffrer un texte connu jusqu'à retrouver le chiffrement
 - permet de connaître la clef
- Assez peu de progrès au début - attaques sur 8/16 rondes (1975-1990)
- **Cryptanalyse différentielle** (16 rondes) [Biham-Shamir 1990]
 - Etude des différences de chiffrement entre des textes similaires
 - Permet de sélectionner des clefs probables
- **Cryptanalyse linéaire** (16 rondes) [Matsui 1993]
 - Utiliser des relations linéaires pour interpoler des bits de la clef

Complexité et coût des attaques sur DES

Méthode d'attaque	Texte connu	Texte choisi	Stockage	Calculs
Précalcul exhaustif		1	2^{56}	1 tableau
Recherche exhaustive	1			2^{55}
Crypta. différentielle	2^{47} puis 2^{36}		Textes	2^{47} puis 2^{36}
Crypta. linéaire	2^{55}	2^{47}	Textes	2^{47}

Cout des attaques sur DES en 1996

Attaquant	Budget	Outil	Clé 56 bits
Hacker	300 euro	Soft circuit	38 ans
PME	7500 euro	Circuit	18 mois
Gde Entreprise	225 Keuro	Circuit ASIC	19 j. 3 h
Multinationale	7,5 Meuro	ASIC	6 min
Gouvernement	225 Meuro	ASIC	12 s

Et aujourd'hui ?

- Problème du DES : clé devenu trop petite !
 - cassable en 8h avec 100 PCs ($2^{56} \simeq 7,2 * 10^{16}$) et $\frac{7,2*10^{16}}{10^9*3600*24*100} \simeq 8$ heures
- Solution 1 : double DES ?
 - $C = E_2(E_1(M))$ et $M = D_1(D_2(C))$
 - Pb : cassage effectif le rend seulement 56 fois plus difficile que DES et non 2^{56} fois
- Solution 2 : triple DES ?
 - 3 clefs : $C = E_1(E_2(E_3(M)))$ et $M = D_3(D_2(D_1(C)))$
 - 2 clefs : $C = E_1(D_2(E_1(M)))$ et $M = D_1(E_2(D_1(C)))$
 - retombe sur DES si $K_2 == K_1$
 - Clef moins longue et sécurité effective identique
 - Attaque similaire double DES \Rightarrow clé effective 112 bits
 - CC : Triple DES double seulement la sécurité !

Principe général de la cryptanalyse statistique sur les chiffrements par bloc

- 1 Etude d'une version réduite en ronde
- 2 Etude la propagation de propriété non aléatoire à travers les rondes
- 3 Etre capable de détecter un chiffré d'une permutation aléatoire
- 4 Ajouter quelques rounds en début/fin tout en assurant la découverte de la clé
- 5 Compromis entre complexité des données et temps d'analyse

Illustration sur des études de cas

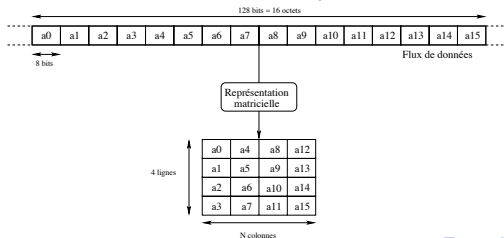
- FEAL-4 (Fast Data Encipherment Algorithm - Miyagushi 87)
 - 4 tours, blocs et clé de 64 bits
 - 88 : 100 à 10000 textes choisis
 - 90 : 20 textes choisis
 - 92 : 5 textes connus
- FEAL-8/FEAL-N/FEAL-NX (Miyagushi 90)
 - 90 : 20.000 textes choisis (Découverte cryptanalyse différentielle !)
 - 92 : 2^{15} textes connus (Découverte cryptanalyse linéaire !)
 - 96 : 12 textes choisis
- IDEA (8 rondes) (Lai, Massey 91)
 - Blocs de 64 bits, clé de 128 bits, 8 rondes
 - 90 : PES ; 91 : PES cassé \implies IPES=IDEA
 - 93 : Cassage sur 2 rondes
 - 97 : Cassage sur 3 rondes
 - 2003 : Cassage sur 5 rondes (2^{24} textes clairs)

Advanced Encryption Standard - AES (2000)

- 1996 : Evaluation DES \implies Il faut un remplaçant !
- 1997 : Appel à candidature international remember Kerckhoffs :-)
 - 15 propositions ; 5 finalistes (1999) :
 - 1 Rijndael (Daemen,Rijmen BE) 10/12/14 rondes
Bloc : 128 bits ; Clé : 128/192/156 bits
 - 2 Serpent (Anderson,Biham,Knudsen UK) 32 rondes
Bloc : 128 bits ; Clé : 128/192/156 bits (en fait :
 $n = 8x \in [0, 2048]$)
 - 3 Twofish (Schneier&al US) 16 rondes
Bloc : 128 bits ; Clé : 128/192/156 bits
 - 4 RC6 (Rivest US) 20 rondes
Bloc : 128 bits ; Clé : 128/192/156 bits (en fait :
 $n = 8x \in [0, 2048]$)
 - 5 MARS (Coppersmith/IBM US) 16 rondes
Bloc : 128 bits ; Clé : 128 \rightarrow 448 bits (128+32k bits)
- 2000 : Standard NIST : AES-Rijndael

Les conventions dans AES

- E/S : blocs de 128 bits ($N_b = 4$)
- Clé : 128, 192 ou 256 bits ($N_k = 4, 6$ ou 8)
- Nb rondes N_r : dépend de N_b et N_k ($N_r \in \{10, 12, 14\}$)
- 1 octet = élément du corps fini à 256 éléments \mathbb{F}_{256}
 - Rappel : p premier, \mathbb{F}_{p^m} isomorphe à $\mathbb{F}_p[X]/g(X)$
 - $g(X)$: polynôme irréd. sur $\mathbb{F}_p[X]$ de degré m
 - Dans AES : $p = 2$, $m = 8$ et $g(X) = X^8 + X^4 + X^3 + X + 1$
- Interprétation matricielle d'un bloc (ex avec 16 octets) :



AES-Rijndael

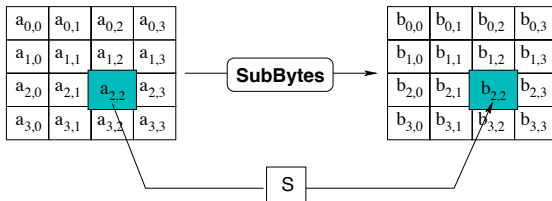
- Standard NIST 2000 ;
- Bloc : 128 bits ; Clé : 128/192/256 bits
- Structure générale :
 - ➊ AddRoundKey Addition initiale de clé
 - ➋ $N_r - 1$ rondes, chacune constituées de 4 étapes :
 - SubBytes : substitution non-linéaire via *S-Box*.
 - ShiftRows : transposition matricielle par décalage à gauche
 - MixColumns : produit matriciel sur colonne
 - AddRoundKey Addition avec les octets des sous-clé
 - ➌ FinalRound : ronde finale (sans MixColumns)

AES : pseudo-code de la fonction de chiffrement

```
AES_Encrypt(State, K) {  
    KeyExpansion(K, RoundKeys);  
    /* Addition initiale */  
    AddRoundKey(State, RoundKeys[0]);  
    /* Les Nr-1 rondes */  
    for (r=1; r<Nr; r++) {  
        SubBytes(State);  
        ShiftRows(State);  
        MixColumns(State);  
        AddRoundKey(State, RoundKeys[r]);  
    }  
    /* FinalRound */  
    SubBytes(State);  
    ShiftRows(State);  
    AddRoundKey(State, RoundKeys[Nr]);  
}
```

Etape SubBytes

- Substitution de chaque élément de la matrice via une SBox
- SBox dérive de la fonction inverse $t : a \longrightarrow a^{-1}$ sur \mathbb{F}_{256} .
 - fonction bien connue pour sa non-linéarité
 - on combine avec une transformation affine inversible f :
- $\text{SBox}[a] = f(t(a)) \forall a \in \mathbb{F}_{256}$
- $\text{SBox}^{-1}[a] = t^{-1}(f^{-1}(a)) = t(f^{-1}(a)) \forall a \in \mathbb{F}_{256}$



Etape Shiftrows

- Opération sur les lignes de matrice
 - La ligne i est décalé de C_i éléments à gauche
 - Le nombre de décalage dépend de N_b (Rijndael) :

N_b	C_0	C_1	C_2	C_3
4	0	1	2	3
5	0	1	2	3
6	0	1	2	3
7	0	1	2	4
8	0	1	3	4

Décalage:

a	b	c	d
e	f	g	h
i	j	k	l
m	n	o	p

ShiftRows

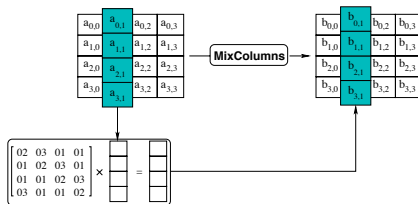
0	a	b	c	d
-1	f	g	h	e
-2	k	l	i	j
-3	p	m	n	o

- Opération inverse : la ligne i est décalée à droite de C_i éléments.

Etape MixColumns

- Operation sur les colonnes de la matrice
 - Considéré un polynôme $a(x)$ de degré 3 dans $\mathbb{F}_{256}[X]$
 - Réalise l'opération : $(03x^3 + x^2 + x + 02) \times a(x) \mod (x^4 + 1)$
 - Matriciellement :

$$b(x) = c(x) \times a(x) \mod (x^4 + 1) \iff \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \times \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$



- Bonne propriétés de diffusion cryptographique

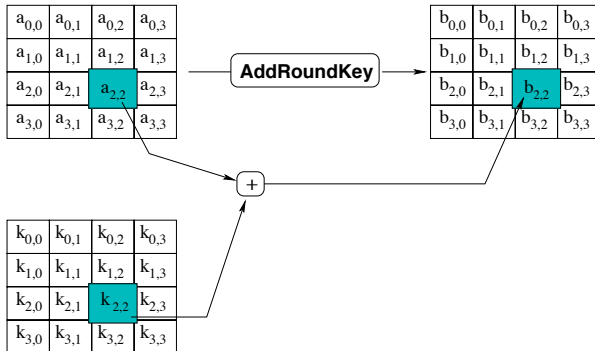
Etape MixColumns⁻¹

- idem mais en utilisant la multiplication par $d(x) = c^{-1}(x)$
 - $(03x^3 + x^2 + x + 02) \times d(x) \equiv 01 \pmod{x^4 + 1}$
 - $d(x) = 0Bx^3 + 0Dx^2 + 09x + 0E$
- Matriciellement :

$$b(x) = d(x) \times a(x) \pmod{x^4 + 1} \iff \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \times \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

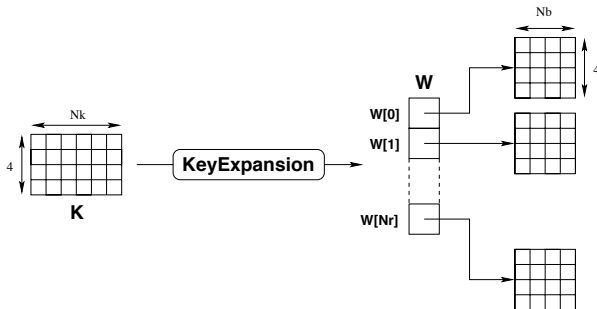
Etape AddRoundKey

- Addition matricielle dans \mathbb{F}_{256} avec une sous-clé



La diversification de la clef KeyExpansion dans AES

- Clé de chiffrement : $4N_k$ octets
- Extension en une clé étendue de $4N_b(N_r + 1)$ octets
 - on dispose ainsi de $N_r + 1$ clés de rondes de $4N_b$ octets



La diversification de la clef KeyExpansion dans AES (3)

```
KeyExpansion(K, W) {  
    /* Recopie directe des Nk premiere colonnes */  
    for (i=0; i<Nk; i++) c[i] = k[i];  
    for (i=Nk; i<Nb*(Nr+1); i++) {  
        tmp = c[i-1];  
        if (i mod Nk == 0)  
            tmp = SubWord(RotWord(tmp)) + Rcon[i/Nk];  
        else if ((Nk > 6) && (i mod Nk == 4)) // Cas Nk > 6  
            tmp = SubWord(tmp);  
        c[i] = c[i-Nk] + tmp;  
    }  
}
```

AES : pseudo-code de la fonction de déchiffrement

- Utilise SubBytes^{-1} , ShiftRows^{-1} , MixColumns^{-1} , et AddRoundKey
- Traitement des clés inchangé

```
AES_Decrypt(State, K) {  
    KeyExpansion(K, RoundKeys);  
    AddRoundKey(State, RoundKeys[Nr]); /* Addition initiale */  
    /* Les Nr-1 rondes */  
    for (r=Nr-1; r>0; r--) {  
        InvShiftRows(State);  
        InvSubBytes(State);  
        AddRoundKey(State, RoundKeys[r]);  
        InvMixColumns(State);  
    }  
    /* FinalRound */  
    InvShiftRows(Out);  
    InvSubBytes(Out);  
    AddRoundKey(Out, RoundKeys[0]);  
}
```

- Ici, séquence des transformations \neq celle du chiffrement
- Il existe une version qui respecte la séquence de transformations du chiffrement

Sécurité de l'AES

- Propriétés cryptanalytiques
 - SBox : sans point fixe ni opposé, ni inverse
 - ShiftRow diffuse les données en séparant les consécutifs
 - MixColumn : chaque bit de sortie dépend de tous les bits en entrée (code correcteur linéaire sur chaque colonne)
- Implémentations "simple" efficace
 - cf TP :-)
 - FPGA : jusqu'à 21.54 Go/s pour le chiffrement
- Cryptanalyse :
 - Aucune attaque significative révélée
 - MAIS seulement 5 ans de recherche
 - (to be continued)

Quelques applications utilisant Rijndael

- SONET (Synchronous Optical NETwork)
- Routeurs Internet
- Switch Ethernet ATM (Asynchronous Transfert Mode)
- Communications Sattelites
- VPN (Réseaux privés virtuels)
- Téléphonie mobile
- Transactions électroniques

Projets de standardisation/Recommandations

- NIST (National Institute of Standards and Technology) (US00)
 - ⇒ AES-Rijndael (Bloc : 128 bits ; Clé : 128/192/256 bits)
25 cycles/octet sur un PIII/Linux.
- KICS (Korean Information and Communication Standards) (Corée01)
 - ⇒ SEED (Bloc : 128 bits ; Clé : 128 bits)
45 cycles/octet sur un PIII.
 - ⇒ ARIA (proposition) (Bloc : 128 bits ; Clé : 128 bits)
37 cycles/octet sur un PIII.

Projets de standardisation/Recommandations (2)

- **NESSIE** (New European Schemes for Signatures, Integrity and Encryption)(EU03)
 - Chiffrement symétriques par bloc :
 - ⇒ MISTY1 (Bloc : 64 bits ; Clé : 128 bits)
47 cycles/octet sur un PIII/Linux.
 - ⇒ AES-Rijndael
 - ⇒ Camellia (Bloc : 128 bits ; Clé : 128/192/256 bits)
35 cycles/octet sur un PIII/Linux.
 - ⇒ SHACAL-2 (Bloc : 256 bits ; Clé : 512 bits)
44 cycles/octet sur un PIII/Linux.
 - Fonctions de hachage à sens unique
 - ⇒ Whirlpool
 - ⇒ SHA-256, SHA-384 et SHA-512 (clairvoyant ! cf SHA-1)
 - Chiffrement asymétriques : PSEC-KEM, RSA-KEM, ACE-KEM
 - Signature électronique : RSA-PSS, ECDSA, SFLASH

Projets de standardisation/Recommandations (3)

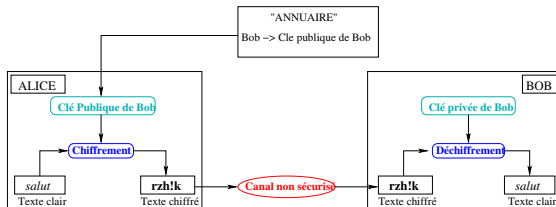
- CRYPTREC (Cryptography Research and Evaluation Committee) (Jap03)
 - Chiffrement symétrique (blocs : 64 bits, clé ≥ 128 bits) :
 - ⇒ CipherUnicorn-E, Hierocrypt-L1, MISTY1, Triple DES
 - Chiffrement symétrique (blocs : 128 bits, clé ≥ 128 bits) :
 - ⇒ AES, Camelia, CipherUnicorn-A, Hierocrypt-3, SC2000
 - Chiffrement symétrique par flot :
 - ⇒ Mugi, Multi-S01, RC4
 - Fonctions de hachage à sens unique
 - ⇒ RIPEMD-160, SHA-1, SHA-256/384/512

Cryptographie à clé publique

Motivations

- Systèmes cryptographiques à clé secrètes
 - pratiquement sûrs
 - efficaces en termes de temps de calcul.
 - Mais nouvelles interrogations :
 - Avant d'utiliser un système de chiffrement à clé secrète, comment convenir d'une clé ?
 - Comment établir une communication sécurisée entre deux entités sans échange préalable de clef ?
- ⇒ Solution apportée par Diffie et Hellman (1976)
- systèmes cryptographiques à clé publique

Principe



Equation fondamentale :

$$\begin{cases} E_{K_e}(M) = C \\ D_{K_d}(C) = M \end{cases}$$

ici : $K_e \neq K_d$,

- K_e publique (connue de tous)
- K_d secrète (connue seulement de Bob)

• Analogie : Boîte aux lettres

- toute personne peut envoyer du courrier à Bob ;
- seul Bob peut lire le courrier déposé dans sa boîte aux lettres.

Quelques pré-requis mathématiques

Theorem (Euclide)

Soit $a, b \in \mathbb{N} / a \leq b$. Soit r le reste de la division euclidienne de a par b . Alors $\text{pgcd}(a,b) = \text{pgcd}(b,r)$.

- Algorithme :
 - Tq $b \neq 0 : (a, b) \longrightarrow (b, a \bmod b)$
 - si $b = 0$ renvoyer a
- Exemple : $\text{pgcd}(42, 30) = 6$

$$(42, 30) \longrightarrow (30, 12)$$

$$(30, 12) \longrightarrow (12, 6)$$

$$(12, 6) \longrightarrow (6, 0)$$

Quelques pré-requis mathématiques (2)

Theorem (Bezout)

Soient $a, b \in \mathbb{Z}$ et $d = \text{pgcd}(a, b)$. Alors $\exists (u, v) \in \mathbb{Z}^2$ tels que

$$au + bv = d$$

Les entiers u et v sont appelés coefficients de Bezout.

- Calcul pratique : Algorithme d'Euclide Etendu

$$(E_0) : \quad 1 \times a \quad + \quad 0 \times b \quad = \quad a$$

$$(E_1) : \quad 0 \times a \quad + \quad 1 \times b \quad = \quad b$$

$$(E_{i+1}) = (E_{i-1}) - q_i(E_i) \quad u_i \times a \quad + \quad v_i \times b \quad = \quad r_i$$

Application sur le calcul d'inverse modulaire

Pb : calcul de $17^{-1} \pmod{50}$

- Calcul des coefficients de Bezout pour $a = 50$ et $b = 17$

$(E_0) :$	1×50	$+$	0×17	$=$	50	
$(E_1) :$	0×50	$+$	1×17	$=$	17	$q_1 = 50/17 = 2$ $r_1 = 50 \% 17 = 16$
$(E_2) = (E_0) - 2 \times (E_1)$	1×50	$+$	$(-2) \times 17$	$=$	16	$q_2 = 17/16 = 1$ $r_2 = 17 \% 16 = 1$
$(E_3) = (E_1) - 1 \times (E_2)$	$(-1) \times 50$	$+$	3×17	$=$	1	$q_3 = 16/1 = 16$ $r_3 = 16 \% 1 = 0$

- Bilan : $(-1) \times 50 + 3 \times 17 = 1 \implies 17^{-1} = 3 \pmod{50}$

Exponentiation rapide modulaire : calcul de $a^e \bmod n$

- Basé sur la remarque suivante :
 - si e est pair, $a^e = (a^{\frac{e}{2}})^2$
 - si e est impair $a^e = (a^{\frac{e}{2}})^2 \cdot a$

Algorithme d'exponentiation rapide modulaire

- 1 Décomposer e en binaire : $e = \sum_{i=0}^k e_i 2^i$
- 2 Calcul de $\{a^{2^i} \bmod n\}_{0 \leq i \leq k}$
 - Utiliser la relation : $a^{2^{i+1}} = (a^{2^i})^2 \bmod n$
- 3 En déduire $a^e = \prod_{i=0}^k (a^{2^i})^{e_i}$

Exponentiation rapide modulaire : exemple

Calcul de $51447^{21} \bmod 17$ (E)

Exponentiation rapide modulaire : exemple

Calcul de $51447^{21} \bmod 17$ (E)

- $51447 = 3026 \times 17 + 5$ donc (E) $\iff 5^{21} \bmod 17$

Exponentiation rapide modulaire : exemple

Calcul de $51447^{21} \bmod 17$ (E)

- $51447 = 3026 \times 17 + 5$ donc $(E) \iff 5^{21} \bmod 17$

① Décomposition de 21 en binaire : $21 = 2^4 + 2^2 + 2^0$

② Calcul de $\{5^{2^i} \bmod 17\}_{0 \leq i \leq 4}$

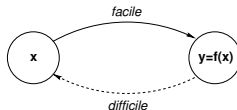
- $i = 0 : 5^{2^0} \equiv 5 \bmod 17$
- $i = 1 : 5^{2^1} = 5^2 = 25 \equiv 8 \bmod 17$
- $i = 2 : 5^{2^2} = 8^2 = 64 \equiv 13 = -4 \bmod 17$
- $i = 3 : 5^{2^3} = (-4)^2 \equiv 16 = -1 \bmod 17$
- $i = 4 : 5^{2^4} = (-1)^2 \equiv 1 \bmod 17$

③ On en déduit :

$$\begin{aligned} 5^{21} &= 5^{2^4} \times 5^{2^2} \times 5^{2^0} \\ &= 1 \times (-4) \times 5 \\ &= -20 \equiv 14 \bmod 17 \end{aligned}$$

Base de la cryptographie à clé publique : fonction a sens unique avec trappe

- Cryptographie à clé publique se base sur des problèmes mathématiques réputés difficiles.
- Fonction à sens unique :



Ex : factorisation d'entiers.

- Fonction à sens unique avec trappe.
 - la connaissance de la trappe (clé) facilite le calcul inverse !

Le cryptosystème RSA

Génération des clefs

- Bob choisit au hasard deux nombres premiers p et q .
 - Bob calcule $n = p.q$
 - Indicatrice d'Euler : $\varphi(n) = (p - 1)(q - 1)$
- Bob choisit au hasard un entier e (impair) tel que
$$\begin{cases} 1 < e < \varphi(n) \\ \text{pgcd}(e, \varphi(n)) = 1 \end{cases}$$
- Bob calcule alors l'entier $1 < d < \varphi(n)$ tel que

$$ed \equiv 1 \pmod{\varphi(n)}.$$

- **Clef publique** : (n, e) (e : exposant RSA ; n : module RSA)
- **Clef secrète** : d .

Le cryptosystème RSA (2)

Chiffrement RSA

- Alice récupère la clef publique (n, e) de Bob
- Pour chiffrer le message M entier tel que $0 \leq M < n$:

$$C = M^e \text{ mod } n$$

- Alice envoie le message chiffré C à Bob.

Le cryptosystème RSA (3)

Déchiffrement RSA

- Pour déchiffrer le message C reçu d'Alice, Bob calcule

$$C^d = M \text{ mod } n$$

En effet, $\exists k \in \mathbb{Z}$ tel que :

$$\begin{aligned} C^d &\equiv M^{e \cdot d} \text{ mod } n \\ &\equiv M^{1+k \cdot \varphi(n)} \text{ mod } n \\ &\equiv M \cdot \left(M^{\varphi(n)} \right)^k \equiv M \text{ mod } n = M \end{aligned}$$

Le cryptosystème RSA : Exemple

Prenons $p = 47$ et $q = 59$.

- On calcule $n = p.q = 47.59 = 2773$
- On choisit e , premier par rapport à $\varphi(n)$. Ex : $e = 17$.
- On calcule alors, par l'algorithme d'Euclide étendu², d tel que $d.e \equiv 1 \pmod{(p-1)(q-1)}$, soit $d = 157$.

Clef publique : $(e, n) = (17, 2773)$

Clef privé : $d = 157$.

- Chiffrement du message $M = 01000010 = 66$:

$$C \equiv M^e \pmod{n} \equiv 66^{17} \pmod{2773} = 872$$

- Déchiffrement de C :

$$C^d \pmod{n} \equiv 872^{157} \pmod{2773} \equiv 66$$

²sous Maple : **igcdex**

Sécurité du cryptosystème RSA

- Le vrai but de l'attaquant : découvrir le texte en clair !
- Calculer d à partir de $(n, e) \iff$ factoriser n .
 - \Leftarrow : trivial (cf génération des clefs)
 - \Rightarrow : Soit $s = \max\{t \in \mathbb{N} : 2^t | ed - 1\}$. On pose $k = \frac{ed-1}{2^s}$. Alors, soit $a \in \mathbb{Z}$ est premier avec n .
 - l'ordre de a^k dans $\mathbb{Z}_n \in \{2^i ; 0 \leq i \leq s\}$ ($a^{\varphi(n)} \equiv 1 \pmod{n}$)
 - si l'ordre de $a^k \pmod{p} \neq$ l'ordre de $a^k \pmod{q}$, alors
$$\exists t \in [0, s[/ 1 < \text{pgcd}(a^{2^t k} - 1, n) < n$$
- On a ainsi trouvé un facteur non trivial de n .
- Algo probabiliste.
- (Coron2004) : algo déterministe presque général($e, d < \varphi(n)$)
 - Toujours d'actualité : Casser RSA aussi dur que factoriser n ?

Sécurité du cryptosystème RSA

- Limites actuelles de factorisation : $\simeq 200$ chiffres
- Record actuel³ : RSA200 (200 chiffres décimaux)
 - Bahr, Boehm, Franke and Kleinjung - 9 mai 2005.
- Si la clef secrète d est petite (de l'ordre de $n^{1/4}$) :
 - attaque utilisant l'algorithme des fractions continues (algorithme LLL)
 - permet de calculer d à partir de n et e .

³<http://www.loria.fr/~zimmerma/records/factor.html>

DLP & Diffie-Hellman

- Autre problème difficile : **Discret Logarithme Problem**

Definition (Logarithme discret)

Soit $G = \langle g \rangle = \{g^i\}_{0 \leq i < n}$ un groupe monogène fini d'ordre n .
Soit $h \in G$. Alors le *logarithme discret* de h en base g , noté $\log_g h$, est l'unique entier x tel que $h = g^x$ ($0 \leq x < n$).

DLP consiste alors à résoudre le problème suivant :

Etant donné G, g, h , trouver $x = \log_g h$.

Exemple : $p = 97$ et $G = \mathbb{Z}/97\mathbb{Z}^* = \{1, 2, \dots, 96\} = \{5^i\}_{0 \leq i < 96}$
 $5^{32} \equiv 35 \pmod{97} \implies \log_5 35 = 32$ dans $\mathbb{Z}/97\mathbb{Z}^*$.

Protocole d'échange de clefs de Diffie-Hellman

Alice et Bob veulent partager une clef secrète K .

On suppose que les données G , $n = |G|$ et g sont publiques.

- Alice choisit un entier $1 \leq a \leq n - 1$ au hasard.
- Alice calcule $A = g^a$ et l'envoie à Bob.
- Bob choisit un entier $1 \leq b \leq n - 1$ au hasard.
- Bob calcule $B = g^b$ et l'envoie à Alice.
- Alice est en mesure de calculer B^a et Bob de calculer A^b .

La clef commune est donc

$$K = g^{ab} = A^b = B^a.$$

Protocole d'échange de clé de Diffie-Hellman

Alice

génère a

$$A = g^a \bmod p$$

Bob

génère b

$$B = g^b \bmod p$$

\xrightarrow{A}

\xleftarrow{B}

(dispose de $[a, A, B, p]$)

Clé secrète : $K = B^a \bmod p$

(dispose de $[b, A, B, p]$)

Clé secrète : $K = A^b \bmod p$

Sécurité de DH

- Problème de DH :
 - connaissant $G, g, A = g^a$ et $B = g^b$, calculer $K = g^{ab}$.
- A l'heure actuelle, résoudre DLP est la seule méthode générale connue pour résoudre DH.
 - MAIS : pas de preuve que résoudre DLP \iff résoudre DH.
- Choix du groupe G : $G = \mathbb{F}_p^*$, $G = E(\mathbb{F}_p)$, etc.
 - Attention au bon choix des paramètres.

Le cryptosystème de El Gamal

Données publiques pré-requise :

- $(G, .) = \langle g \rangle$ un groupe cyclique d'ordre n

Génération des clefs

- Bob choisit $a \in [1, n - 1]$ et calcule $A = g^a$ dans G .
- Clef publique : (G, g, n, A) .
- Clef secrète : a .

Le cryptosystème de El Gamal (2)

Chiffrement

Alice souhaite envoyer le message $M \in G$ à Bob

- Alice récupère la clef publique (G, g, n, A) de Bob.
- Alice choisit au hasard $k \in [1, n - 1]$
- Le message chiffré qu'Alice envoie à Bob est $C = (y_1, y_2)$ avec

$$\begin{cases} y_1 = g^k \\ y_2 = M.A^k \end{cases}$$

Le cryptosystème de El Gamal (3)

Déchiffrement

- Bob reçoit le message chiffré $C = (y_1, y_2)$
- Il lui suffit alors de calculer

$$M = y_2 \cdot (y_1^a)^{-1} = y_2 \cdot y_1^{n-a}$$

En effet :

$$\begin{aligned} y_2 \cdot y_1^{n-a} &= M \cdot A^k \cdot (g^k)^{n-a} \\ &= M \cdot g^{a \cdot k} \cdot g^{k \cdot n} \cdot g^{-ka} \\ &= M \cdot g^{a \cdot k} \cdot (g^n)^k \cdot g^{-ka} \\ &= M \cdot g^{a \cdot k} \cdot g^{-ka} = M \end{aligned}$$

Sécurité du cryptosystème de El Gamal

- Résoudre DLP dans $G \implies$ Casser El Gamal dans G
 - l'attaquant peut alors calculer a à partir de A (public).
- La réciproque n'est pas encore prouvée !

Cas particulier de $G = \mathbb{F}_p^*$:

- utiliser un nombre premier p de 1024 bits choisis uniformément
- permet de résister aux méthodes actuelles de résolution de DLP sur \mathbb{F}_p^*

Fonctions de hachage et signatures électroniques

Notion de fonction de hachage

Definition (Fonction de Hachage)

Une fonction de hachage H est une application facilement calculable qui transforme une chaîne binaire de taille quelconque t en une chaîne binaire de taille fixe n , appelée *empreinte de hachage*.

- En général, $t > n$: H est surjective
- On parle de *collision* entre x et x' lorsque

$$\begin{cases} x \neq x' \\ H(x) = H(x') \end{cases}$$

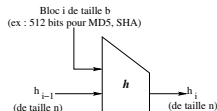
- Si y est tel que $y = H(x)$, alors x est appelé *préimage* de y

Propriétés des fonction de hachage

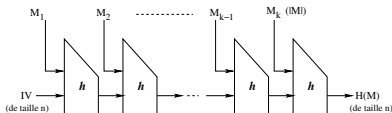
- Propriétés de base : compression et facilité de calcul.
- Propriétés additionnelles :
 - Résistance à la préimage
 - étant donné y , il est difficile de trouver x tel que $y = H(x)$
 - Rrésistance à la seconde préimage
 - étant donné x , il est difficile de trouver $x' \neq x$ tel que $H(x) = H(x')$
 - Résistance à la collision
 - il est difficile de trouver x et x' tels que $H(x) = H(x')$.
- Fonction de Hachage à Sens Unique
 - résistance à la préimage et à la seconde préimage
- Fonction de Hachage résistante aux collisions
 - résistance à la seconde préimage et à la collision

Construction d'une fonction de hachage

- Définir une fonction de compression h



- Pour calculer l'empreinte d'un message M :
 - Application d'un *padding* à M pour que $|M| = k.b$
 - Découpage du message M en blocs de tailles b
 $M = M_1 M_2 \dots M_{k-1} M_k$ avec $|M_i| = b \quad \forall i \in [1, k]$
 - Itération de la fonction h (IV : Initial Value) :



- Exemples connus : MD5, SHA-1, SHA-2, Whirlpool...

Idée générale des signatures électroniques

But des signatures manuscrites :

- prouver l'identité de leur auteur **et/ou**
- l'accord du signataire avec le contenu du document

La signature électronique dépend du signataire **et** du document !

Objectifs d'une signature électronique

- Une signature est authentique.
- Une signature ne peut être falsifiée (imitée).
- Une signature n'est pas réutilisable sur un autre document.
- Un document signé est inaltérable.
- Une signature ne peut pas être reniée.

Idée générale des signatures électroniques (2)

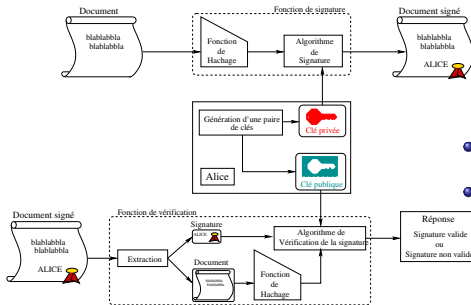
- Réalisation pratique :
 - Cryptosystèmes à clef secrète (et arbitre)
 - Cryptosystèmes à clef publique + fonction de hachage
- On préfère signer le hachage d'un document
 - Taille fixe suffisamment petite pour être utilisée efficacement par un cryptosystème à clé publique

Idée générale des signatures électroniques (3)

- Protocole de signature électronique sûr :
 - Impossible de falsifier la signature $s(M)$ d'un document M
 - sans connaître la clef secrète K (resp. K_d)
 - même en disposant de signatures d'autres documents.
 - Attention : impossibilité *pratique*
- Il existe d'autres conditions nécessaires de sécurité !
 - relève davantage des architectures de sécurité des cryptosystèmes à clef publiques (PKI) ou du secret entourant la clef secrète.

Idée générale des signatures électroniques (4)

- Signature utilisant un cryptosystème à clef publique :



- Alice signe M en utilisant :

- $h_M = H(M)$ le hachage de M
- sa clé secrète K_d .
- la fonction de déchiffrement D .
- Résultat : $s(M) = D_{K_d}(h_M)$

- Document signé : $[M, s(M)]$

- Vérification de $[M, s(M)]$:

- utilise la clé publique K_e d'Alice et la fonction de chiffrement E
- $E_{K_e}(s(M)) = h_M = ? H(M)$
- Seule Alice a pu générer $s(M)$

Signature RSA

Génération des paramètres

Identique à la génération des clefs de RSA !

- Alice choisit au hasard deux nombres premiers p et q .
 - Alice calcule $n = p.q$
 - Indicatrice d'Euler : $\varphi(n) = (p - 1)(q - 1)$
- Alice choisit au hasard un entier e (impair) tel que $1 < e < \varphi(n)$ et $\text{pgcd}(e, \varphi(n)) = 1$
- Alice calcule alors l'entier d tel que $e.d \equiv 1 \pmod{\varphi(n)}$.

Clef publique : (n, e)

Clef secrète : d

On suppose disposer d'une fonction de hachage à sens unique H connue publiquement.

Signature RSA (2)

Génération d'une signature RSA

Alice souhaite signer un document M

- Alice calcule $h_M = H(M)$ (on suppose $0 \leq h_M < n$)
- Signature de M : $s(M) = (h_M)^d \bmod n$
- Le document signé est alors $[M, s(M)]$.

Signature RSA (3)

Vérification d'une signature RSA

- Bob reçoit un document signé $[\tilde{M}, s(M)]$ d'Alice.
 - Ce document est potentiellement altéré/illégitime
- Il récupère la clé publique d'Alice (n, e)
- Il calcule $\tilde{h}_M = H(\tilde{M})$
- Il vérifie l'identité : $s(M)^e \equiv \tilde{h}_M \pmod{n}$

En effet : $s(M)^e \equiv (h_M)^{e \cdot d} \pmod{n} \equiv h_M \pmod{n} = h_M$ et si le document est authentique : $h_M = \tilde{h}_M$.

- La sécurité est donc celle du cryptosystème RSA.
- Présentation simpliste et en l'état sujette à des attaques

Signature El Gamal

Génération des paramètres

- Alice choisit :
 - un nombre premier p
 - g une racine primitive modulo p .
 - un entier $a \in \{1, \dots, p-2\}$ au hasard
- Elle calcule alors $A = g^a \bmod p$.

Clef publique : (p, g, A) .

Clef secrète : a .

On suppose disposer d'une fonction de hachage à sens unique H connue publiquement.

Signature El Gamal (2)

Génération d'une signature El Gamal

Alice souhaite signer un document M

- Alice calcule $h_M = H(M)$ (on suppose $0 \leq h_M < p$)
- Elle choisit au hasard un entier $k \in [1, p-2]$ tel que $\text{pgcd}(k, p-1) = 1$ ($\implies k^{-1} \in \mathbb{Z}_{p-1}$ existe).
- Signature de M : $s(M) = (r, s)$ avec

$$\begin{cases} r = g^k \mod p \\ s = k^{-1}(h_M - a.r) \mod (p-1) \end{cases}$$

- Le document signé est alors $[M, s(M)]$.

Signature El Gamal (3)

Vérification d'une signature El Gamal

- Bob reçoit un document signé $[\tilde{M}, s(M)]$ d'Alice.
 - Rappel : $s(M) = (r, s)$
 - Ce document est potentiellement altéré/illégitime
- Il récupère la clé publique d'Alice (p, g, A)
- Il calcule $h_M = H(\tilde{M})$
- Il vérifie l'identité : $A^r r^s \equiv g^{h_M} \pmod{p}$

En effet,

$$\begin{aligned} A^r r^s &\equiv g^{a \cdot r} \cdot g^{k k^{-1} (h_M - a \cdot r)} \pmod{p} \\ &\equiv g^{h_M} \pmod{p} \end{aligned}$$

Si le document est authentique : $h_M = \tilde{h}_M \Rightarrow g^{h_M} \equiv g^{\tilde{h}_M} \pmod{p}$

Sécurité des signatures El Gamal

- Sécurité intimement liée à DLP dans \mathbb{F}_p^*
 - Résolution de DLP dans \mathbb{F}_p^*
 - \implies possibilité de calculer a à partir de A
 - \implies possibilité d'impersonaliser Alice
- Attention au choix des paramètres.

Le standard DSA

Génération des paramètres

- Alice génère un nb premier q de 160 bits ($2^{159} \leq q < 2^{160}$)
- Elle génère un nb premier p de 512 à 1024 bits vérifiant :

$$\begin{cases} \exists t \in [0, 8] / 2^{511+64t} < p < 2^{512+64t} \\ q | (p - 1) \end{cases} \quad (2)$$

- Soit \tilde{g} une racine primitive modulo p
- Un générateur du sous-groupe de \mathbb{F}_p^* d'ordre q est alors

$$\mathbf{g} = \tilde{g}^{\frac{p-1}{q}} \mod p$$

(2) assure que \mathbb{F}_p^* possède un sous-groupe d'ordre q

Le standard DSA (2)

Génération des paramètres (suite)

Une fois choisis (p, q, g) :

- Alice choisit $a \in \{1, \dots, q-1\}$
- Elle calcule $A = g^a \bmod p$

Clef publique : (p, q, g, A) .

Clef secrète : a

Le problème du logarithme discret sous-jacent se passe dans le groupe d'ordre q .

Le standard DSA (3)

Génération d'une signature DSA

Alice souhaite signer un document M :

- Alice calcule $h_M = H(M)$ (on suppose $1 \leq h_M < q - 1$)
- Elle choisit un entier $k \in \{1, \dots, q - 1\}$
- Signature de M : $s(M) = (r, s)$ avec

$$\begin{cases} r = (g^k \bmod p) \bmod q \\ s = k^{-1}(h_M + a.r) \bmod q \end{cases}$$

- Le document signé est alors $[M, s(M)]$.

Le standard DSA (4)

Vérification d'une signature DSA

- Bob reçoit un document signé $[\tilde{M}, s(M) = (r, s)]$ d'Alice.
- Il récupère la clé publique d'Alice (p, q, g, A)
- Il vérifie que les formats sont respectés : $1 \leq r, s \leq q - 1$
- Il calcule $\tilde{h}_M = H(\tilde{M})$
- Il vérifie l'identité :

$$r \equiv \left[\left(g^{s^{-1} \tilde{h}_M} \bmod q \right) \cdot \left(A^{rs^{-1}} \bmod q \right) \bmod p \right] \bmod q.$$

Ce qu'il nous reste à voir...

- Développement sur les cryptosystèmes à clé publique
 - Primalité
 - Résolution des problèmes sous-jacent : Factorisation, DLP
- Les architectures PKI
- Sécurité informatique
 - Sécurité de la programmation
 - Sécurité des infrastructures