

Eléments de Cryptoanalyse



**Roumen Andonov, Nadia Bennani,
Didier Donsez**

Université de Valenciennes

Institut des Sciences et Techniques de Valenciennes

e-mail : {andonov,nbennani,donsez}@univ-valenciennes.fr

Sommaire

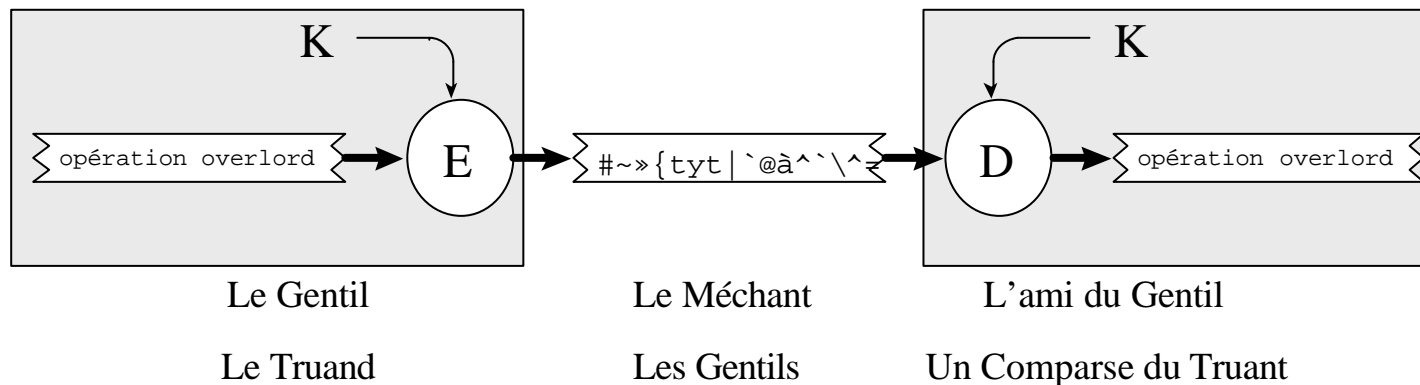
- Rappel sur la Cryptographie
- Chiffrement à clé symétrique (clé secrète) - DES
- Chiffrement à clé asymétrique (clé publique) - RSA
- Éléments de cryptanalyse
 - Fonction de hachage à sens unique
 - Attaque d'un chiffrement
 - longueur des clés secrètes
 - longueur des clés publiques
- Rudiments mathématiques - théorie de la complexité

le Chiffrage (Cryptage)

- **algorithme public**
 - connu de tous
 - le secret est maintenue
tant que la clé n'est pas connu
 - qui peut être propriétaire : royalties
- **chiffrage à clé symétrique**
 - (clé secrète)
- **chiffrage à clé asymétrique**
 - (clé publique / clé privée)

le Chiffrement à clé symétrique (clé secrète)

- 1 seule clé pour chiffrer et déchiffrer



- DES (Data Encryption Standard - IBM 1977), IDEA

DES

(Decryption Encryption Standard)

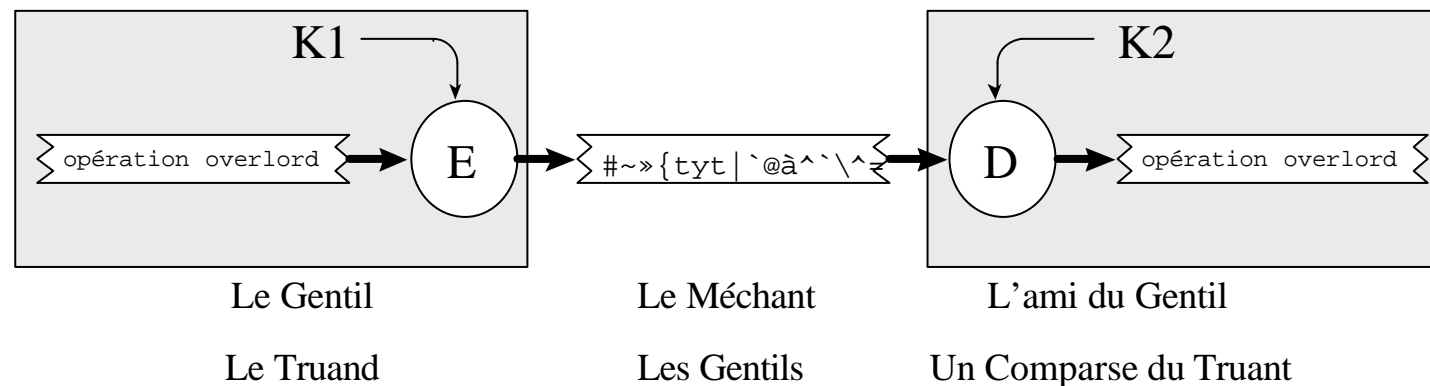
- Principe
 - Succession de Rouleaux de Permutation
 - Machine ENIGMA
- DES
 - 56 bits
 - Triple-DES (3*56 bits)
- Limite de DES
 - DES56 « cassable »
 - Juillet 98 : 56 heures par une machine à 250 000 \$
<http://www.cdt.org/crypto/>
 - Appel d 'offre du NIST pour un remplaçant
 - AES (Advanced Encryption Standard)

le Chiffrement à clé asymétrique (clé publique / clé privée)

■ 2 clés K1 et K2

- si chiffrement par K1, déchiffrement par K2
- si chiffrement par K2, déchiffrement par K1

Remarque : on ne peut pas trouver une clé à partir de l'autre



■ RSA (Rivest Shamir Adelman)

RSA (Rivest Shamir Adelman)

Génération des Clés de B

B sélectionne 2 nb. premiers p et q

B tire la clé publique K_{pub}

tq $\text{pgcd}(K_{pub}, (p-1)(q-1))=1$

B calcule la clé privée K_{priv}

$K_{priv} * K_{pub} = 1 \pmod{(p-1)(q-1)}$

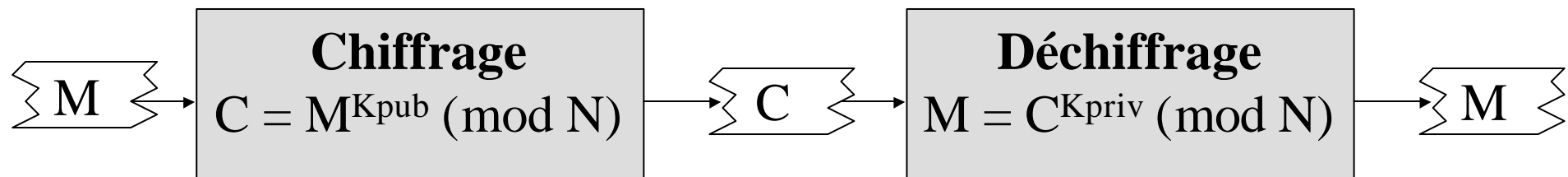
(par l'alg. d'Euclide étendu)

Privé

- p, q nombres premiers
- K_{priv} une clé secrète

Public

- $N = p * q$
- K_{pub} une clé publique



RSA - Exemple

Génération des Clés de B

B sélectionne $p=47$ et $q=71$, alors $N=pq=3337$

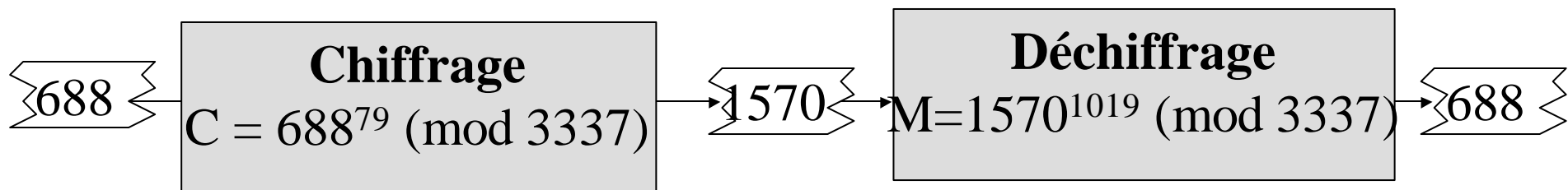
$$(p-1)(q-1)=46*70=3220$$

B tire aléatoirement la clé publique $K_{pub}=79$

B calcule la clé privée K_{priv}

$$K_{priv} = 79^{-1} \pmod{3220} \Rightarrow K_{priv}=1019$$

B publie n et K_{pub} , garde K_{priv} secret et jette p et q



Chiffrements par bloc et en continu

■ Chiffrement par bloc

- Principe:
 - le message est découpé en blocs (de 1,8,32 ou 64 bits)
 - chaque bloc est chiffré indépendamment de la valeur des autres blocs
- Algo : DES, RSA, IDEA, RC2

■ Chiffrement en continu

- Principe:
 - le message est un flot de texte ou de données binaires découpé en blocs (de 1,8,32 ou 64 bits)
 - chaque bloc est chiffré en fonction de la valeur de la clé **mais aussi** de la valeur du bloc précédent (et/ou suivant)
- Algo : RC4, SEAL, WAKE

La CryptoAnalyse

■ Attaque d'un chiffrement

- l'attaquant cherche à connaître
 - le texte en clair
 - la clé « secrète ou privée » utilisée
- à partir d'un texte encodé : très difficile
 - Paul Leyland et 1600 machines relèvent le Défi [1994]
RSA : 129-digits (430 bits) -> 5000 Mips - Year
« THE MAGIC WORDS ARE SQUEAMISH OSSIFRAGE »
- à partir du texte clair et du texte encodé : faisable
Attention aux en-têtes de formulaires !!!!

Validité d'un Chiffrage

- dépendant de la nature de la donnée à protéger
 - transaction bancaire
 - » quelques minutes
 - secret d'état, signature de contrat à long terme
 - » 50 ans
- dimension de la clé
 - plus la clé est grande, elle est difficile à casser

■ Réponse : augmentation de la dimension de la clé

- Même algorithme mais agrandi (*TripleDES*)
- Surchiffrement (avec 2 ou 3 clés) (*xDES*)
 - pas de changement du programme ou du hardware
- Nouveaux algorithmes

Génération des Clés

■ DES

- Clé Secrète : un nombre quelconque
 - hormis certains 0, ...
- tirage aléatoire sécurisé
 - surtout non reproductible

(car il permettrait aux crypto-analyses le limite de domaine de recherche des clés)

■ RSA

- 2 nombres premiers p et q très grands

Systeme Cryptographie

■ Fonction

- Authentification
- Confidentialité
- Non Répudiation

■ Outils à assembler

- Algorithme de Chiffage à Clé Symétrique
- Algorithme de Chiffage à Clés Asymétriques
- Fonction de Hachage
- Générateur de Nombres Aléatoires
- Générateur de Nombres Premiers
- ...

Théorie de la complexité

- La complexité est mesurée par 2 paramètres
 - T: complexité en temps
 - S: complexité en espace (mémoire)
- T et S sont exprimés en fonction de n (taille du pb.)
- Classes d'équivalence
 - Polynomiaux : (linéaire, quadratique, cubique, etc)
 - Exponentiels : $O(t^{f(n)})$ où t est const et f(n) est fonction polynomiale de n
 - Super-polynomiaux : $O(t^{f(n)})$ où f(n) est plus qu'une const. mais moins que linéaire

Complexité des problèmes

- Pbs. Solubles - qui peuvent être résolus avec des algs. polynomiaux
- Pbs. Non-solubles - qui ne peuvent pas être résolus en temps polynomial (pbs. difficiles)
- Pbs. indécidables - impossible de concevoir un algorithme pour la résolution du problème

Classes de complexité

- **P** : pbs. qui peuvent être résolus en temps polynomial
- **NP**:pbs. qui peuvent être résolus en temps polynomial sur une machine de Turing non déterministe
- **NP-complets**:pbs aussi difficile que tout autre pb dans NP
- **PSPACE**: pbs. qui peuvent être résolus en espace polynomial mais pas nécessairement en temps polynomial
- **PSPACE-complets**:
 - si n 'importe lequel d 'entre eux est dans NP => PSPACE=NP;
 - si l 'un d 'entre eux est dans P =>PSPACE=P
- **EXPTIME**: pbs. solubles en temps exponentiel
- **EXPTIME-complets**: pbs. qui ne peuvent pas être résolus en temps déterministe polynomial (P not = EXPTIME)

Fonction de hachage à sens unique

- $h=H(M)$ où
 - M est de longueur arbitraire
 - la valeur de hachage h est de longueur fixe
- + les caractéristiques suivantes:
 - Etant donné M , il est facile de calculer h ;
 - Etant donné h , il est difficile de calculer M ;
 - Etant donné M , il est difficile de trouver un autre message M' , tq $H(M')=H(M)$;
- + résistance à la collision
 - Il est difficile de trouver 2 messages aléatoires M et M' tq $H(M')=H(M)$;
- Exemples: il est facile de multiplier deux grands nombres premiers, mais il est difficile de factoriser leur produit
 - multiplication de un l -bits par un k -bits se fait en $O(kl)$;
 - factorisation de nombre n se fait en $e^{(1+o(1))(\ln n)^{1/2} (\ln(\ln n))^{2/3}}$;

Force et Faiblesse d'un Système Cryptographie

■ Force d'un système cryptographique

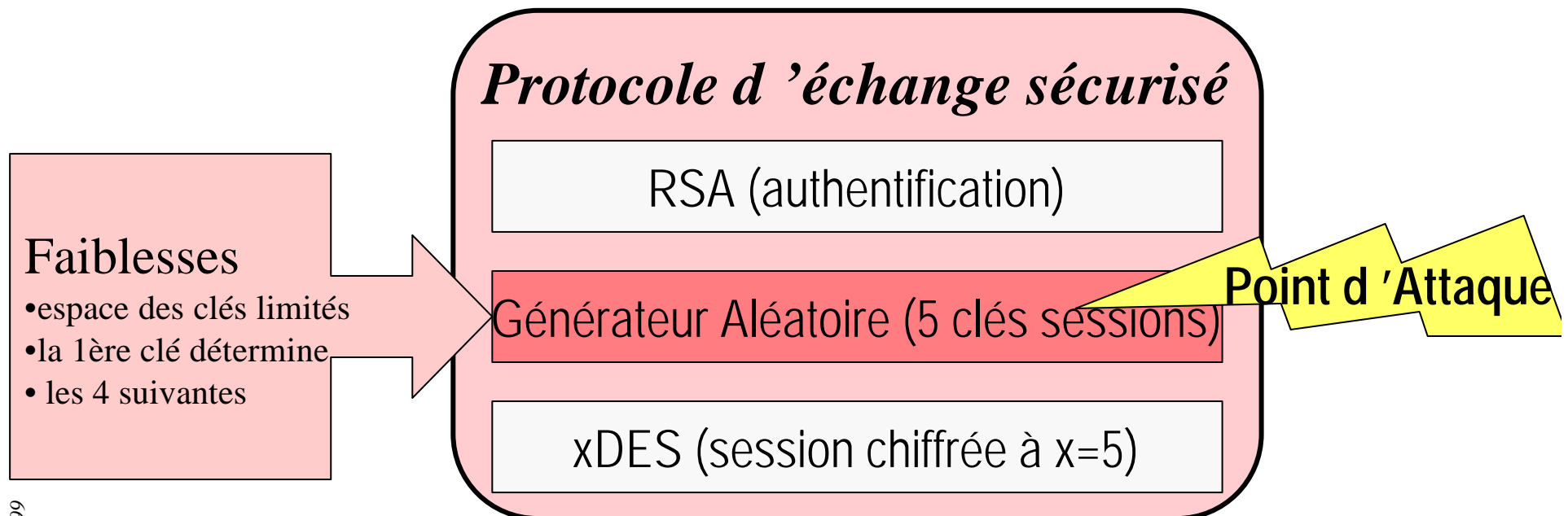
- = Force du composant le plus faible

B. Schneier, « Cryptographic Design Vulnerabilities », Computer, 09/98

www.counterpane.com

■ Exemple

- Syst. = Protocole d'échange sécurisé



Longueur des clés secrètes

- *La sécurité d'un cryptosystème à clé secrète doit résider dans la clé et non pas dans les détails de l'algorithme !*
- Hypothèse: la solidité de l'alg. est parfaite (l'attaque exhaustive est le seul moyen possible de casser le cryptosystème). On connaît un bout de texte chiffré et du texte en clair correspondant.
- Estimation du temps et du coût d'une attaque exhaustive de DES
 - machines dédiées
 - 1977 la machine de Diffie et Hellman
10⁶ proc. chaque teste 10⁶ clés/sec. => 2⁶⁴ clés en 214 jours
 - 1993 la machine de Wiener (puces et cartes spécialisées)
coût 10⁶ \$, 2⁵⁶ clés en 3h30 en moyen
 - généralisation de ces résultats sur tab. 7.1
 - Rappel de loi de Moore: la puissance de calcul double tous le 18 mois (le coût est divisé par 10 tous les 5 ans)
 - méthodes logicielles : 1000 fois plus lentes mais « gratuites »
 - Le Peer-to-Peer
 - 10000 PC sur le Web se partage l'espace de recherche
 - Une annlet dans le page de garde d'un site hacké et visité comme Google

Longueur des clés publiques

- *La cryptographie à clé publique est basée sur utilisation des fonctions à sens unique*
 - factorisation des grands nombres qui sont le produit de deux grands nombres premiers.
 - Problème logarithmique discret
- Les records de factorisation sur tab 7.3 .
 - **Attention:** Factoriser 512 bits est dans le domaine de possible!!!
- Mesure utilisée : mips-an $3 \cdot 10^{13}$ instr. (10^6 instr./sec. pendant 1 an)
 - ex. Pentium 100 MHz est de 50 mips
- Algs. utilisés: le crible quadratique, le crible général sur corps numérique, le crible spécial sur corps numérique
- Récommandations:
 - un module de 1024 bits devrait être suffisant jusqu'au 2005
 - pour les 20 prochaines années 1024 bits seront insuffisant
 - La longueur doit être adaptée au niveau de sécurité de votre clé. Tab 7.6

Bibliographie

Cryptographie Appliquée, Bruce Schneier (Wiley), 1996,
ISBN 0-471-59756-2 (ISBN 2-84180-036-9 en VF)