Cryptographie

M. Belkasmi

ENSIAS 2012-2013

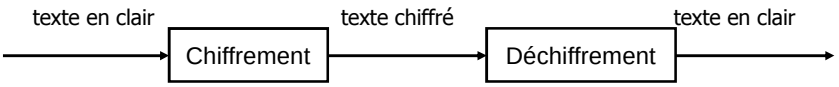
Plan

- Introduction
- Cryptographie symétrique
- Cryptographie asymétrique

Introduction

Terminologie

- Expéditeur
- Destinataire
- Message
  - texte en clair
  - texte chiffré (cryptogramme)



Terminologie

- **Cryptographie**: art et science du chiffrement
- **Cryptanalyse**: Science permettant d'étudier les systèmes cryptographiques en vue de les tester ou de les casser
- **Cryptologie**: Branche de l'informatique théorique qui traite de la cryptographie et de la cryptanalyse

Formalisation

- M : suite de caractères (message)
- Fonction de chiffrement (cryptage)
  - E(M)=*C*
- Fonction de déchiffrement (décryptage)
  - D(*C*)=M
- On doit avoir D(E(M))=M

Algorithme cryptographique

- Paire de fonctions utilisées pour le chiffrement et le déchiffrement.

- **Algorithme restreint:** l'algorithme est secret
  - N'est plus utilisé
  - La conception d'un tel algorithme nécessite un expert
  - On ne peut utiliser d'algorithmes commerciaux
  - Différents algorithmes pour différents utilisateurs.

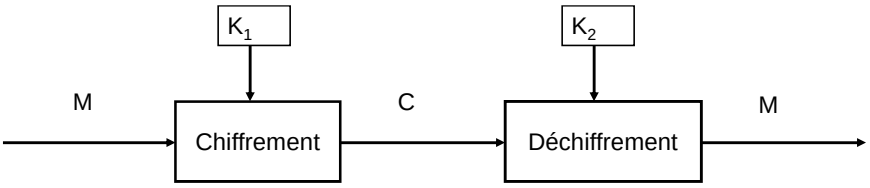
Clef cryptographique

- Valeur K provenant d'un grand ensemble de valeurs possibles **(espace des clefs)**.

- En général on a besoin de deux clefs
  - K_1 : pour le chiffrement
  - K_2 : pour le déchiffrement

- L'algorithme cryptographique est connu de tous mais dépend des deux clefs:
  - $E_{K1}(M)=C$
  - $D_{K2}(C)=M$

Clef cryptographique



Le secret réside dans la (ou les) clef(s) et non dans les détails de l'algorithme (principes de Kerchoffs).

Les objectifs de la cryptographie

- **Confidentialité :** Ceux qui ne sont pas les destinataires d'une information ne doivent pas avoir accès à cette information.
- **Authentification :** il doit être possible pour le récepteur du message de garantir son origine. Une tierce personne ne doit pas pouvoir se faire passer pour quelqu'un d'autre (usurpation d'identité).

Les objectifs :

- **Intégrité** : le récepteur doit pouvoir s'assurer que le message n'a pas été modifié durant sa transmission. Une tierce personne ne doit pas pouvoir substituer un message légitime (ayant pour origine l'émetteur) par un message frauduleux.
- **Non répudiation** : un émetteur ne doit pas pouvoir nier l'envoi d'un message.

Les Applications :

- Armée (et plus généralement sécurité au niveau des états),
- Système bancaire,
- Internet (achats, identification, déclaration d'impôts),
- Téléphones portables, clés électroniques (e.g., voitures),
- TV payante,
- Cartes d'identités électroniques, cartes de santé,
- Vote électronique,
- DVD, audio numérique (certains formats, e.g.,WMA, AAC),
- …

Deux types d'algorithmes

- **Algorithmes à clef secrète (crypto-systèmes symétriques)**
  - K1 peut être calculé à partir K2 et vice versa.
  - On a souvent K1=K2
  - K1 et K2 doivent être secrètes

- **Algorithmes à clef publique (crypto-systèmes asymétriques)**
  - K1≠K2
  - K2 ne peut pas être calculé à partir de K1
  - K1 peut être publique
  - K2 doit être secrète (clef privée)

Algorithme à clef secrète

Chiffre à substitution: Chaque caractère du message est remplacé par un autre caractère.

- Exemple: Chaque caractère du message est remplacé par celui qui se trouve K places plus loin (modulo 26)
  - Code de Jules César: K=3
  - A → D B → E etc
  - SEMINAIRE → VHPLQDLUH

  → Substitution mono-alphabétique.

Cryptanalyse du chiffre de César

- **Recherche exhaustive** : Essayer les 26 clés possibles jusqu'à obtenir un message intelligible.
- **Analyse de fréquence** : Déterminer la lettre du message la plus fréquente. La connaissance de la lettre la plus fréquente permet de retrouver la clé.

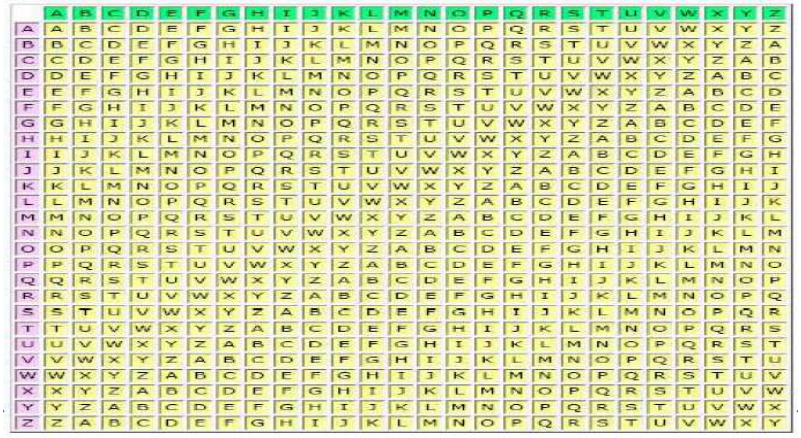
→ La robustesse était en fait basé sur le secret de l'algorithme

- Les fréquences d'apparitions des lettres dans la langue française :

A → 0	8.11 %	J → 9	0.18 %	S → 18	8.87 %
B → 1	0.81 %	K → 10	0.02 %	T → 19	7.44 %
C → 2	3.38 %	L → 11	5.99 %	U → 20	5.23 %
D → 3	4.28 %	M → 12	2.29 %	V → 21	1.28 %
E → 4	17.69 %	N → 13	7.68 %	W → 22	0.06 %
F → 5	1.13 %	O → 14	5.20 %	X → 23	0.53 %
G → 6	1.19 %	P → 15	2.92 %	Y → 24	0.26 %
H → 7	0.74 %	Q → 16	0.83 %	Z → 25	0.12 %
I → 8	7.24 %	R → 17	6.43 %		

Substitution poly-alphabétique

Plusieurs substitutions sont utilisées au cours du chiffrement selon la clef



Algorithme à clef secrète

- **Chiffre à transposition**: Les caractères du message sont inchangés mais leur position est modifiée.
  - Exemple (transposition simple en colonne):

"L'assassin est le docteur Matrix, regardez derrière l'horloge"

```
LASSASSIN
ESTLEDOCT
EURMATRIX
REGARDEZD
ERRIERELH
ORLOGE
```

"LEERE OASUE RRSTR GRLSL MAIOA EAREG SDTDR ESORE EICIZ LNTXD H"

Cryptographie symétrique

- Introduction
- Algorithme DES
- Algorithme AES
- Echange de clé

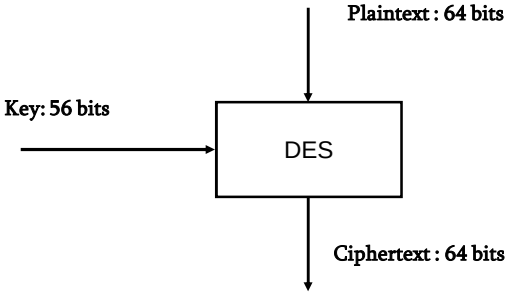
Crypto-systèmes symétriques

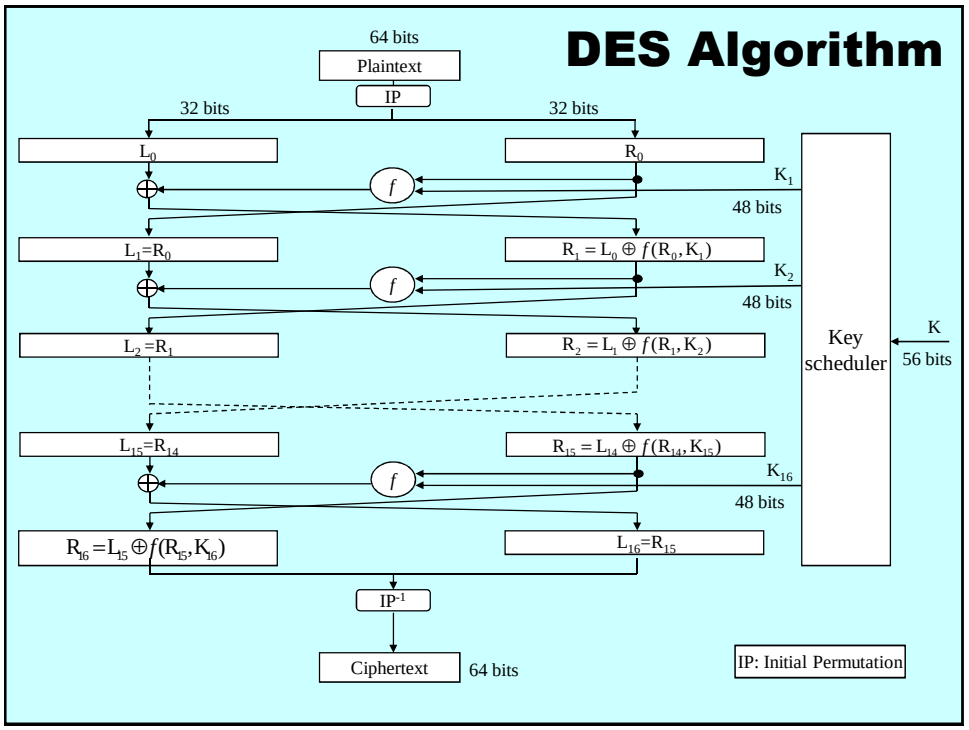
- **Algorithmes à clef secrète** :
  K1 peut être calculé à partir K2 et vice versa.
  - On a souvent K1=K2
  - K1 et K2 doivent être secrètes

- **Exemples DES, AES..**

Algorithme DES

- DES : Data Encryption Standard
- Développé par IBM, standard depuis 1977
- Utilise des clés de taille 56 bits
- 16 rondes.
- 16 clés de ronde de 48 bits



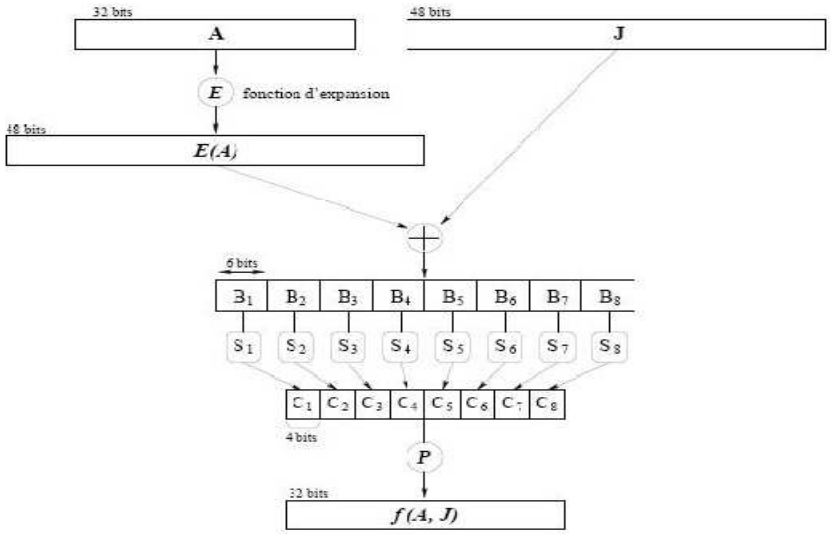


Permutation initiale

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

- Si p =p1 p2 p3… alors IP(p)=p58 p50 p42…
- IP^{-1} est également donnée par une table. Opération très rapide (lecture d'une table).

La fonction de ronde f



Les substitutions S-Box

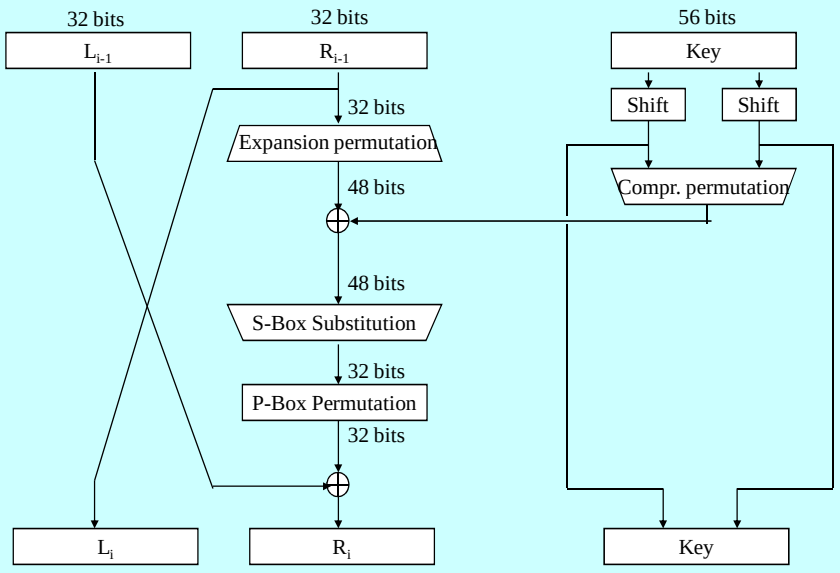
Substitution de 6 bits sur 4 bits

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

Utilisation des S-box :
Si l'entrée d'une S-box est b1b2b3b4b5b6, sa sortie est le développement binaire de la valeur se situant sur à l'intersection de la ligne b1b6 et de la colonne b2b3b4b5 dans la S-box (on numérote les lignes et les colonnes à partir de 0).

→ Les S-box sont le coeur du DES.

One Round of DES



Historique et cryptanalyse du DES

- 1977 : DES devient un standard FIPS
- 1998 : une machine dédiée casse une clé DES en 56h
- 1999 : clé DES cassée en 22h via un calcul distribué
- 2005 : DES est retiré des standards FIPS

Etape 1 : exploitation du DES;
Etape 2 : DES n'offre plus un niveau de sécurité acceptable
Etape 3 : le procédé 3DES a remplacé DES mais il est extrêmement lourd

→ AES : Advanced Encryption Standard (Rijndael)

Algorithme AES

- En 1997 le NIST [National Institute of Standards and Technology] fait à nouveau un appel d'offre pour l'élaboration d'un nouveau système cryptographique

- Le cahier des charges comporte les points suivants :
  - Grande sécurité (!)
  - Large portabilité.
  - Rapidité.
  - Lecture aisée de l'algorithme
  - Le chiffrement se fait par blocs de 128 bits. Les clés comportent 128, 192 ou 256 bits.

Le Choix du NIST

- Le 2 octobre 2000, le NIST donne sa réponse

- L'algorithme retenu est Rijndael (Vincent **Rij**men et Joan **Dae**men)
- Bon compromis entre sécurité, performance, efficacité, facilité d'implémentation et flexibilité
- Rijndael travaille par blocs de 128 bits et il est symétrique
- La taille de la clé est généralement de 128 bits avec les variantes 192 et 256 bits

Description rapide de l'AES

- Chiffrement par blocs itérés
- taille des blocs : 128 bits
- taille des clés : 128, 192 ou 256 bits
- 10,12 ou 14 rondes
- Travaille sur $\{0;1\}^8$ (i.e. sur les octets)
- Utilise des tableaux 4x4 d'octets (états)

Les rondes de l'AES

- **Constituées de 4 étapes :**
  - SubBytes : substitution non linéaire où chaque octet est remplacé par un autre via une table

    → (S-box sur des octets).
  - ShiftRows : permutation où chaque ligne de l'état est soumise à une permutation circulaire de longueur variable.

Les rondes de l'AES

- **Les dernières étapes :**
  - MixColumns :mélange sur les colonnes via une transformation linéaire.
  - AddRoundKey : Chaque octet de l'état est combiné avec la clé de ronde.

Échange de clefs

- Le premier algorithme d'échange de clefs a été inventé par Diffie et Hellman en 1976.
- Alice et Bernard se mettent d'accord sur deux grands entiers n et g de telle manière que g soit *primitif* par rapport à n.

Nombres primitifs

- g est primitif par rapport à n si tous les nombres entre 1 et n-1 peuvent être exprimés sous la forme g^i mod n.
- **Exemple: g=2 et n=11**

  $2^{10} = 1024 = 1 \pmod{11}$
  $2^1 = 2 = 2 \pmod{11}$
  $2^8 = 256 = 3 \pmod{11}$
  $2^2 = 4 = 4 \pmod{11}$
  $2^4 = 16 = 5 \pmod{11}$
  $2^9 = 512 = 6 \pmod{11}$
  $2^7 = 128 = 7 \pmod{11}$
  $2^3 = 8 = 8 \pmod{11}$
  $2^6 = 64 = 9 \pmod{11}$
  $2^5 = 32 = 10 \pmod{11}$

Protocole de Diffie et Hellman

- Alice choisit un grand nombre entier aléatoire x et envoie à Bernard le résultat du calcul:

$$A=g^x \bmod n$$

- Bernard choisit un grand nombre entier aléatoire y et envoie à Alice le résultat du calcul:

$$B=g^y \bmod n$$

- Alice calcule $k = B^x \bmod n$
- Bernard calcule $k' = A^y \bmod n$
- Les valeurs k et k' sont toutes deux égales à $g^{xy} \bmod n$

Cryptographie asymétrique

- Introduction
- Algorithme RSA
- Algorithme Merkle-Hellman

Crypto-systèmes asymétriques

- **Algorithmes à clef publique (crypto-systèmes asymétriques)**
  - K1≠K2
  - K2 ne peut pas être calculé à partir de K1
  - K1 peut être publique
  - K2 doit être secrète (clef privée)

- **Exemples RSA, Merkle-Hellman**

Cryptologie à clef publique

- Clef publique K1 (ou privée)
- Clef privée K2 (ou publique)
- Fonction de chiffrement: $E_{K1}(M)$ utilise une clé K1
- Fonction de déchiffrement: $D_{K2}(C)$ utilise une clé K2
- $DK_2(EK_1(M)) = M$

Comment cela fonctionne?

- Chaque utilisateur doit posséder une clef publique et une clef privée.
- Si Alice veut envoyer un message M à Bernard elle utilise la clef publique K1 de Bernard et envoie:

$$C = E_{K1}(M)$$

- Bernard reçoit C et utilise sa clef privée K2 pour calculer:

$$M = D_{K2}(C)$$

Quel est l'avantage?

- Usage 1 : N'importe qui peut envoyer un message à Bernard en utilisant sa clef publique (→ Confidentialité).
- Usage 2 : Bernard peut utiliser sa clef privée pour « signer » un message M (→ authentification et intégrité du message).
- Autres techniques : Signature numérique

Algorithme RSA

- Crypto-Système à clé publique introduit en 1978 par Rivest, Shamir et Adleman après le papier de Diffie et Hellman en 1976.
- Il est basé sur la difficulté de factoriser des nombres.

Clefs du RSA

- La Clé privée :
  - p et q : deux nombres premiers très grands.
  - Un entier e premier avec le nombre (p – 1)(q – 1).
  - Ces données sont très confidentielles
- La clé publique :
  - n = p×q
  - d l'inverse de e modulo $\phi(n)$ avec $\phi(n)=(p-1)(q-1)$ (ie un entier tel que $e \times d = 1 \bmod \phi(n)$)
  - ces données sont publiques et peuvent être divulgués.
- Aller : clé privée → clé publique : facile mais l'inverse est très difficile.

Chiffrement/déchiffrement RSA

- On suppose que Alice veut transmettre a Bernard un message Chiffré M. Le message M est représenté par un nombre $< n$.
- Fonction de chiffrement: $E(M) = M^e \bmod n$
- Fonction de déchiffrement: $D(C) = C^d \bmod n$

Utilisation de RSA

- Quand le message est représenté par $M < n$ le chiffrement se fait en une seule étape.
- Pour chiffrer de plus long messages il suffit de le diviser en blocs $M=M1\ M2\ \ldots\ Mt$ de sorte que $Mi < n$ pour tout $1 \leq i \leq t$
- On peut aussi utiliser RSA pour transmettre une clef et ensuite utiliser un algorithme à clef secrète.

Exemple 1

- Les clés :

$$p = 5, q = 7, n = 35, \Phi(n) = (p-1)(q-1) = 24$$

$$e = 5, p\,\gcd(5,24) = 1, d = e^{-1} = 5, 5*5 = 25 \equiv 1 (\mathrm{mod}\, 24)$$

- Chiffrement/déchiffrement :

$$E(3) \equiv 3^5 \equiv 243 \equiv 33 (\mathrm{mod}\, 35)$$

$$D(33) \equiv 33^5 \equiv 39135393 \equiv 3 (\mathrm{mod}\, 35)$$

$$E(5) \equiv 5^5 \equiv 10 (\mathrm{mod}\, 35)$$

$$D(10) \equiv 10^5 \equiv 5 (\mathrm{mod}\, 35)$$

Exemple 2

- Prenons p=47 et q=71
- n = 3337 et $\phi(n)=(p-1)(q-1) = 46 \cdot 70 = 3220$
- Choisissons e=79 et d=1019
  (on peut vérifier que ed = 80501 = 1+ 25 ·3220)

Exemple 2 suite

- Pour chiffrer le message 6882326879666683 on commence par le diviser en petit blocs de trois chiffres:

  m1 m2 m3 m4 m5 m6 = 688 232 687 966 668 3

- En utilisant la clef (e,n)=(79, 3337) on calcule le premier bloc:

  $688^{79} \bmod 3337 = 1570 = c1$

Exemple 2 fin

- Les autres blocs sont chiffrés de la même manière et on obtient:

  c1 c2 c3 c4 c5 c6 = 1570 2756 2091 2276 2423 158

- Pour déchiffrer le message on utilise la clef (d,n)=(1019, 3337) et on obtient pour le premier bloc:

  $1570^{1019} \bmod 3337 = 688 = m_1$

- Le reste du message en clair est obtenu de la même manière

Rappel : Arithmétique modulaire

$x \equiv y (\mathrm{mod}\, n)$ ssi $x = kn + y$ avec $y < n$

$27 = 3*7 + 6$ alors $27 \equiv 6 (\mathrm{mod}\, 7)$

$x (\mathrm{mod}\, n) + y (\mathrm{mod}\, n) \equiv x + y (\mathrm{mod}\, n)$

- Exemple

$9 + 11 = 20 \equiv 6 (\mathrm{mod}\, 7)$ et

$9 (\mathrm{mod}\, 7) + 11 (\mathrm{mod}\, 7) \equiv 2 + 4 \equiv 6 (\mathrm{mod}\, 7)$

Rappel : Arithmétique modulaire

$x (\mathrm{mod}\ n) * y (\mathrm{mod}\ n) \equiv x * y (\mathrm{mod}\ n)$

- Exemple

$9*11 = 99 = 14*7 + 1 \equiv 1 (\mathrm{mod}\, 7)$ et

$9 (\mathrm{mod}\, 7) + 11 (\mathrm{mod}\, 7) \equiv 2*4 \equiv 8 \equiv 1 (\mathrm{mod}\, 7)$

Exponentiation modulaire

Comment calculez $165789^{23456781} (\mathrm{mod}\, 456712)$

$x^8 = \left(\left(x^2\right)^2\right)^2$ donc pour calculer $x^{(2^k)} (\mathrm{mod}\ n)$

on calcule $x \leftarrow x^2 (\mathrm{mod}\, n)$ k fois

- Exemple

$21 = 10101 = 2^4 + 2^2 + 2^0$ et

$5^2 = 8 (\mathrm{mod}\ 17)$ $5^4 = 13 (\mathrm{mod}\ 17)$

$5^8 = 16 (\mathrm{mod}\ 17)$ $5^{16} = 1 (\mathrm{mod}\ 17)$

$5^{21} (\mathrm{mod} 17) = 5^{16} 5^4 5^1 (\mathrm{mod} 17) = 1 * 13 * 5 = 14 (\mathrm{mod} 17)$

Calcul du PGCD

- Algorithme Euclide (récursif) :

$euclide(a,b)$ $a > b$

$(a,b) \leftarrow (b, a \bmod b)$

si $b = 0$ alors répondre a

- Exemples

$euclide(42,30) = 6$

$(42,30)$

$(30,12)$

$(12,6)$

$(6,0)$

$euclide(105,45) = 15$

$(105,45)$

$(45,15)$

$(15,0)$

Inverse multiplicatif

- Algorithme Euclide étendu (récursif)

$On \quad cherche\ a^{-1} \bmod m \quad$ avec $\quad \text{pgcd}(a,m) = 1$

$(m, a, 1, 0)$

$(a, b, c, d) \leftarrow (b, a \bmod b, d - c(a \text{ div } b) \bmod m, c)$

si $b = 1$ alors répondre c

- Exemples

$5^{-1} \equiv 8 \bmod 13$

$(13, 5, 1, 0)$

$(5, 3, 0 - 1*(2) \bmod 13, 1) = (5, 3, 11, 1)$

$(3, 2, 1 - 11*(1) \bmod 13, 11) = (3, 2, 3, 11)$

$(2, 1, 11 - 3*(1) \bmod 13, 3) = (2, 1, 8, 3)$

$5*8 = 40 \equiv 1 \bmod 13$

$7^{-1} \equiv 19 \bmod 22$

$(22, 7, 1, 0)$

$(7, 1, 0 - 1*(3) \bmod 22, 1) = (7, 1, 19, 1)$

$7*19 = 133 = 6*22 + 1 \equiv 1 \bmod 22$

Avec un ordinateur, on calcule le PGCD et l'inverse multiplicatif de très grands nombres efficacement.

Trouver les clefs du RSA

- On veut un entier e relativement premier à $\phi(n)$.
- On choisit e < $\phi(n)$ au hazard puis on appelle Euclide(e, $\phi(n)$) (pour trouver leur pgcd):

```
Euclide(a,b)
    si b=0 alors retourner a
    sinon retourner Euclide(b,a mod b)
```

- Si e n'est pas relativement premier à $\phi(n)$ alors on incrémente e successivement jusqu'à ce que l'on obtienne le nombre désiré.

Pourquoi cela fonctionne-t-il?

Lemme: Soit $n=pq$ et $\phi(n)=(p-1)(q-1)$.
Alors pour tout $g \in \{1,2,\ldots,n-1\}$ on a $g^{\phi(n)} = 1 \pmod n$

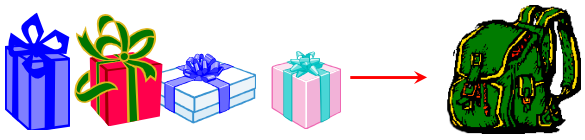
Remarque: Ce lemme est une généralisation du petit théorème de Fermat.

Donc $D(E(M)) = D(M^e \bmod n)$
$= M^{ed} \bmod n$
$= M^{1+c\phi(n)} \bmod n$ pour un certain c.
$= M \cdot ((M^{\phi(n)})^c) \bmod n$
$= M \cdot 1^c \bmod n$
$= M \bmod n$

Les Chiffres Knapsack

- Exemple chiffre de Merkle-Hellmann 1978
- Système asymétrique (le premier !)
- Difficulté du problème du sac à dos
- Le système a été cassé en 1982 par Adi Shamir et n'est plus utilisé de nos jours

Problèmes du sac à dos



- **Problème de décision :**
  Soient un ensemble de paquets de poids donnés et un sac de poids P. Existe-t-il un sous-ensemble de paquets de poids total P ?

Problèmes du sac à dos

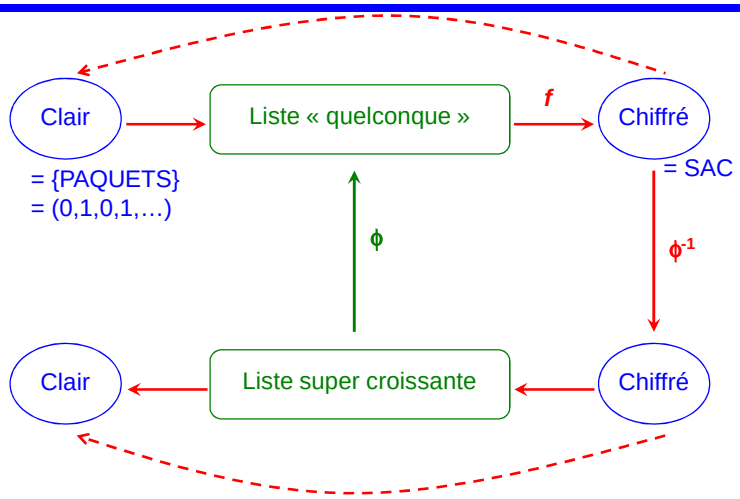
Problème de cryptanalyse :

- Comment déterminer les paquets qui ont été mis dans le sac, sans avoir le droit de l'ouvrir et en ne connaissant que les poids des paquets disponibles et le poids total du sac ?

Chiffre Knapsack

- Le problème Knapsack :
  - Difficile (NP-complet)
- Le problème knapsack avec une liste super-croissante (LSC) :
  - Simple
- L'idée de Merkle et Hellman
  - Choisir un problème knapsack avec une LSC
  - Le Transformer en un problème knapsack
  - Le problème knapsack est la clef publique, alors que le problème knapsack avec une LSC est la clef privée.

Algorithme NP-complet



Algorithme linéaire

Le problème du Knapsack

- Considérons A= $(a_1,a_2,..,a_i,\ldots,a_n)$ une liste de n entiers naturels.
- Etant donné un nombre s qui est la somme d'un sous ensemble de A, trouver un vecteur binaire $X=(x_1,x_2,..,x_i,\ldots,x_n)$ tel que $s= \sum_{i=1}^{n} x_i a_i$

Le problème du Knapsack avec une LSC

- Considérons A= $(a_1,a_2,..,a_i,\ldots,a_n)$ une liste super-croissante d'entiers naturels i.e $a_i > \sum_{j=1}^{i-1} a_j$ (cad chaque élément est supérieur à la somme des éléments qui le précèdent).
- Etant donné un nombre s qui est la somme d'un sous ensemble de A, trouver un vecteur binaire $X=(x_1,x_2,..,x_i,\ldots,x_n)$ tel que $s= \sum_{i=1}^{n} x_i a_i$

Algorithme pour un problème Knapsack avec une LSC

- For $i = n$ downto 1
  - If $s \geq a_i$ then $\{x_i=1; s = s - a_i\}$ else $x_i=0$;
- Return $X=(x_1,x_2,..,x_n)$

- Exemple : $A=(3,7,11,25,51)$ une LSC

- Pour $s=69$ ➔ $X=(0,1,1,0,1)$

Chiffre de Merkle-Hellman

- Choisir une liste super-croissante d'entiers naturels $A= (a_1,a_2,..,a_i,...,a_n)$
- Choisir un entier $u > 2.a_n$ et un nombre p relativement premier avec le nombre u.

- Pour chaque élément a_i de A calculer $b_i=p.a_i \pmod u$ de telle sorte à deduire une nouvelle liste $B=(b_1,b_2,..,b_i,...,b_n)$ qui n'est plus super-croissante.

- Clef publique B;
- Clef privée A, u, p, $p^{-1} \pmod u$.

Exemple de Chiffrement / Déchiffrement

- M = (m_i) : Message clair binaire à n bits.
- B = (b_i) : Liste quelconque à n entiers.
- ***f***: M $\rightarrow$ C= Σ $m_i \times b_i$ (c'est le chiffrement)
- ϕ : x $\rightarrow$ p.x (mod u)
- ϕ^{-1} : x $\rightarrow$ p^{-1}.x (mod u)
- B=p.A= ϕ(A) → A=p^{-1}.B= ϕ^{-1}(B)
- C=M.B → C'=M.A problème knapsack avec LSC

Exemple 1

- Liste super - croissante :
  A=(3,7,11,25,51)

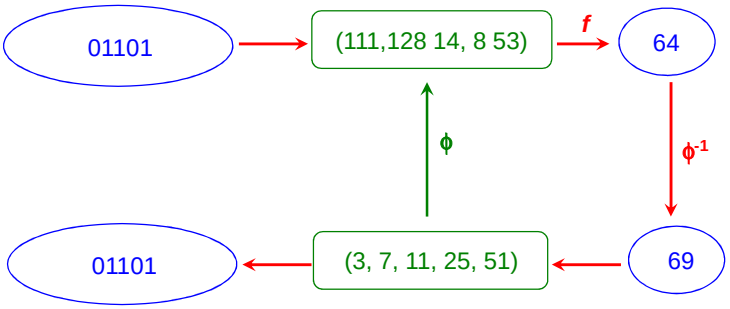
  Clé privée :
  (p,u)=(37,131) $\rightarrow$ ϕ(x) = 37x (mod 131)
  $\rightarrow$ ϕ^{-1}(x) = 85x (mod 131)

- Liste quelconque (Clé publique) :
  B=(111,128,14,8,53)= p.A

  Chiffrement de M=(0,1,1,0,1) :

  C =128+14+53 = 195=64 mod 131

Exemple 1 suite



Déchiffrement de C'=69 :

M =(0,1,1,0,1) par algo knapsack avec LSC

Exemple 2

- Liste super - croissante :
  A=(1,2,4,9,20,40,77,160,321,640,1280,2557,5112,10225,20449,40899)

Liste quelconque (Clé publique) :
B=(447,894,1788,4023,8940,17880,34419,71520,61348,39663,79326,
75172,67311,52930,23274,46995)

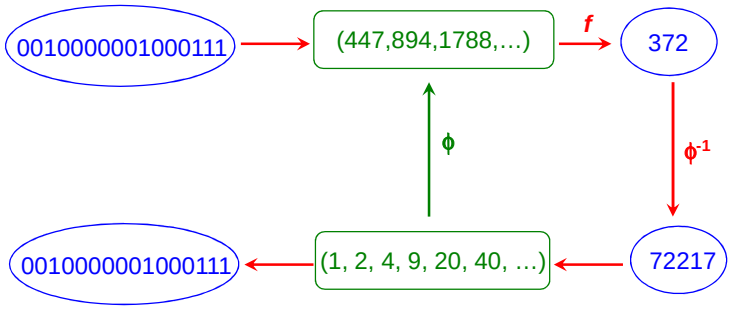
Clé privée :

(447,82139) $\rightarrow \phi(x) = 447x \pmod{82139}$
$\rightarrow \phi^{-1}(x) = 74605x \pmod{82139}$

Chiffrement :

$0\times447+0\times894+1\times1788+\ldots+1\times46995 = 164650$
$= 372 \bmod 82139$

Exemple 2 suite



Déchiffrement de C'=72217 :

M =(0010000001000111) par algo knapsack avec LSC