

# CHIFFREMENT DE RABIN

- Inventé par Michael Rabin en 1979



Michael Rabin

- Amélioration du processus de chiffrement
- Problème de factorisation

# CHIFFREMENT DE RABIN

- Choisir deux nombres premiers  $p$  et  $q$  :
  - $p \bmod 4 = 3$
  - $q \bmod 4 = 3$
- Clé publique  $\Rightarrow n = p \times q$
- Clé privée  $\Rightarrow (p, q)$
- Chiffrement  $\Rightarrow C = M^2 \bmod n$
- Déchiffrement  $\Rightarrow M = C^{1/2} \bmod n$ 
  - $m_p = C^{(p+1)/4} \bmod p$
  - $m_q = C^{(q+1)/4} \bmod q$
  - $p \times y_p + q \times y_q = 1$
  - $R = (p \times m_q \times y_p + q \times m_p \times y_q) \bmod n$
  - $S = (p \times m_q \times y_p - q \times m_p \times y_q) \bmod n$
  - Solutions possibles :  $\{R, S, n-R, n-S\}$

## CHIFFREMENT DE RABIN

- $p = 11, q = 23, M=158$
- $n = 11 \times 23 = 253 \Rightarrow$  Clé publique
- $(11, 23) \Rightarrow$  Clé privée
- $C = 158^2 \bmod 253 = 170$
- $m_p = 170^{(11+1)/4} \bmod 11 = 4$
- $m_q = 170^{(23+1)/4} \bmod 23 = 3$
- $11y_p + 23y_q = 1 \Rightarrow \{y_p = -2, y_q = 1\}$
- $R = (11 \times 3 \times (-2) + 23 \times 4 \times 1) \bmod 253 = 26$
- $S = (11 \times 3 \times (-2) - 23 \times 4 \times 1) \bmod 253 = 95$
- Les solutions possibles :  $\{26, 95, (253 - 26) = 227, (253 - 95) = 158\}$

# CHIFFREMENT DE MERKLE~HELLMAN

- Inventé par Ralph Merkle et Martin Hellman en 1978



Ralph Merkle



Martin Hellman

- Nouvelle direction de chiffrement à clés asymétriques
- Problème du Sac-à-dos (*Knapsack problem*)

## PROBLÈME DE SAC-À-DOS (SUITE NORMALE)

1x

**A** 48 kg

0x

**B** 01 kg

0x

**C** 12 kg

1x

**D** 10 kg

1x

**E** 11 kg

1x

**F** 08 kg

0x

**G** 15 kg

1x

**H** 02 kg

SAC	
Objet	Taille
Poids total	79 kg

# PROBLÈME DE SAC-À-DOS (SUITE SUPER-CROISSANTE)

0×

A 96 kg

1×

B 39 kg

0×

C 20 kg

1×

D 11 kg

1×

E 05 kg

0×

F 02 kg

1×

G 01 kg

SAC

Objet	Taille
Poids total	56 kg

Reste 17 kg

Reste 06 kg

Reste 01 kg

Reste 00 kg

## PROBLÈME DE SAC-À-DOS

- Etant donnée une suite  $A = \{P_1, P_2, \dots, P_k\}$  et un poids  $T$ . Est-il possible de trouver une suite de bits  $b_1 \dots b_k$  tel que

$$T = b_1 \times P_1 + b_2 \times P_2 + \dots + b_k \times P_k$$

- Suite normale  $\Rightarrow$  le problème est **NP-Difficile**
- Suite super-croissante  $\Rightarrow$  il existe un algorithme polynomial permettant de calculer les  $b_i$

```
Pour  $i=k$  à  $1$  Faire  
    Si  $T \geq P_i$  Alors  $T := T - P_i$ ;  
                            $b_i := 1$ ;  
    Sinon  $b_i := 0$ ;  
Fin pour  
Si  $T = 0$  Alors  $\{b_1, b_2, \dots, b_k\}$  est la solution ;  
Sinon Il n'existe pas de solutions ;
```

## RÉSoudre UNE SUITE SUPER-CROISSANTE

○  $A = \{2, 3, 6, 13, 27, 52\}$  pour  $T = 70$

○  $70 \geq 52 \Rightarrow T = 70 - 52 = 18$

○  $18 < 27 \Rightarrow$


○  $18 \geq 13 \Rightarrow T = 18 - 13 = 5$

○  $5 < 6 \Rightarrow$

○  $5 \geq 3 \Rightarrow T = 5 - 3 = 2$

○  $2 \geq 2 \Rightarrow T = 2 - 2 = 0$

$b_6 = 1$   
 $b_5 = 0$   
 $b_4 = 1$   
 $b_3 = 0$   
 $b_2 = 1$   
 $b_1 = 1$



○  $S = 110101_{(2)} = 53$

$$1 \times 2 + 1 \times 3 + 0 \times 6 + 1 \times 13 + 0 \times 27 + 1 \times 52 = 70$$



# CHIFFREMENT DE MERKLE~HELLMAN

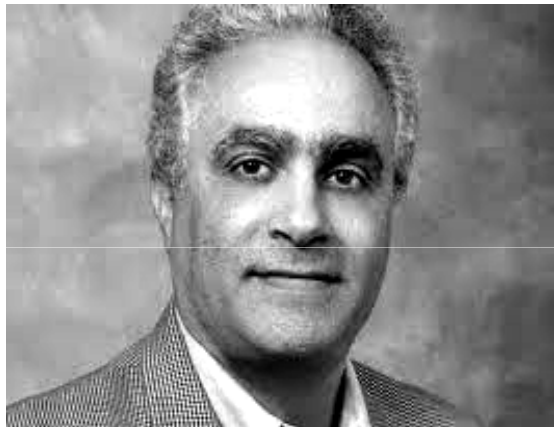
- Choisir une suite super-croissante  $A = \{P_1, P_2, \dots, P_k\}$
- Choisir deux nombres  $n$  et  $m$  :
  - $m > \sum P_i$
  - $\text{pgcd}(n, P_i) = 1$
- Calculer la suite  $B = \{P'_1, P'_2, \dots, P'_k\}$  tel que  $P'_i = n \times P_i \bmod m$
- Clé publique  $\Rightarrow B$
- Clé privée  $\Rightarrow (A, n, m)$
- Chiffrement
  - $M = b_1 b_2 \dots b_k$
  - $C = b_1 \times P'_1 + b_2 \times P'_2 + \dots + b_k \times P'_k$
- Déchiffrement
  - Calculer  $n^{-1}$  tel que  $n \times n^{-1} \bmod m = 1$
  - Calculer  $T = n^{-1} \times C \bmod m$
  - Calculer les  $b_i$  avec l'algorithme de résolution de suite super-croissante

## CHIFFREMENT DE MERKLE~HELLMAN

- $A = \{2, 3, 6, 13, 27, 52\}, n = 31, m = 105, M=53$
- $B = \{62, 93, 81, 88, 102, 37\}$
- $M = 53 = 110101_{(2)}$   
 $\Rightarrow C = 1 \times 62 + 1 \times 93 + 0 \times 81 + 1 \times 88 + 0 \times 102 + 1 \times 37 = 280$
- $31 \times n^{-1} \bmod 105 = 1 \Rightarrow n^{-1} = 61$
- $T = (61 \times 280) \bmod 105 = 70$   
 $= 1 \times 2 + 1 \times 3 + 0 \times 6 + 1 \times 13 + 0 \times 27 + 1 \times 52$   
 $\Rightarrow M = 110101_{(2)} = 53$

# CHIFFREMENT D'ELGAMAL

- Inventé par Taher Elgamal en 1985



Taher Elgamal

- Concept hérité du protocole de Diffie-Hellman
- Permet le chiffrement et déchiffrement des messages
- Problème du Logarithme Discret

# CHIFFREMENT D'ELGAMAL

- Choisir un grand nombre premier  $p$  et deux nombres  $a$  et  $g$  :
  - $a < p$
  - $g < p$
- Calculer  $A = g^a \bmod p$
- Clé publique  $\Rightarrow (A, g, p)$
- Clé privée  $\Rightarrow a$
- Chiffrement
  - Choisir un nombre aléatoire  $b$  ( $b < a$  et  $\text{pgcd}(b, p-1) = 1$ )
  - Calculer  $B = g^b \bmod p$
  - Calculer  $C = M \times A^b \bmod p$
  - Le chiffré du message  $\Rightarrow (B, C)$
- Déchiffrement  $\Rightarrow M = C \times B^{p-a-1} \bmod p$

## CHIFFREMENT D'ELGAMAL

- $p = 23, g = 7, a = 6, M = 7, b = 3$
- $A = 7^6 \bmod 23 = 4$
- $B = 7^3 \bmod 23 = 21$
- $C = 7 \times 4^3 \bmod 23 = 11$
- $M = 11 \times 21^{23-6-1} \bmod 23 = 7$

# SIGNATURE NUMÉRIQUE AVEC RSA

- Paire de clés de l'expéditeur  $\Rightarrow \{(e, n), (d, n)\}$
- Expéditeur  $\Rightarrow$  Génération de la signature de M
  - Calculer  $h=H(M)$
  - Calculer  $S=h^d \bmod n$
  - La signature numérique  $\Rightarrow (M, S)$
- Destinataire  $\Rightarrow$  Vérification de la signature
  - Calculer  $h1=H(M)$
  - Calculer  $h2=S^e \bmod n$
  - Si  $h1=h2 \Rightarrow$  la signature est valide

# SIGNATURE NUMÉRIQUE AVEC RABIN

- Paire de clés de l'expéditeur  $\Rightarrow \{n, (p, q)\}$
- Expéditeur  $\Rightarrow$  Génération de la signature de M
  - Calculer  $h=H(M)$
  - Calculer  $\alpha=h^{\frac{1}{2}} \bmod n$
  - Choisir S parmi les quatre solutions possibles
  - La signature numérique  $\Rightarrow (M, S)$
- Destinataire  $\Rightarrow$  Vérification de la signature
  - Calculer  $h1=H(M)$
  - Calculer  $h2=S^2 \bmod n$
  - Si  $h1=h2 \Rightarrow$  la signature est valide

# SIGNATURE NUMÉRIQUE AVEC MERKLE~HELLMAN

- Paire de clés de l'expéditeur  $\Rightarrow \{B, (A, n, m)\}$
- Expéditeur  $\Rightarrow$  Génération de la signature de M
  - Calculer  $h=H(M)$
  - Calculer  $n^{-1}$  et  $T=n^{-1} \times h \bmod m$
  - Calculer les  $b_i$
  - La signature numérique  $\Rightarrow (M, S)$  tel que  $S=b_1b_2 \dots b_k$
- Destinataire  $\Rightarrow$  Vérification de la signature
  - Calculer  $h1=H(M)$
  - Calculer  $h2 = b_1 \times P'_1 + \dots + b_k \times P'_k$
  - Si  $h1=h2 \Rightarrow$  la signature est valide



# SIGNATURE NUMÉRIQUE AVEC ELGAMAL

- Paire de clés de l'expéditeur  $\Rightarrow \{(A, g, p), a\}$
- Expéditeur  $\Rightarrow$  Génération de la signature de M
  - Calculer  $h=H(M)$
  - Choisir un nombre aléatoire  $b$  ( $b < p$  et  $\text{pgcd}(b, p-1)=1$ )
  - Calculer  $B=g^b \bmod p$
  - Calculer  $C$  tel que  $h=(a \times B + b \times C) \bmod (p-1)$
  - La signature numérique  $\Rightarrow (M, S)$  tel que  $S=(B, C)$
- Destinataire  $\Rightarrow$  Vérification de la signature
  - Calculer  $h=H(M)$
  - Si  $A^B \times B^C \bmod p = g^h \bmod p \Rightarrow$  la signature est valide

## SIGNATURE NUMÉRIQUE AVEC ELGAMAL

- $p = 23, g = 7, a = 6, A = 4, h(M) = 7, b = 3$
- $B = 7^3 \bmod 23 = 21$
- C tel que  $(6 \times 21 + 3C) \bmod 22 = 7 \Rightarrow C = 19$
- Vérification :  $4^{21} \times 21^{19} \bmod 23 = 7^7 \bmod 23 = 5$