Cryptographie(suite)

Cryptographie à clé publique

2020-2021

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?

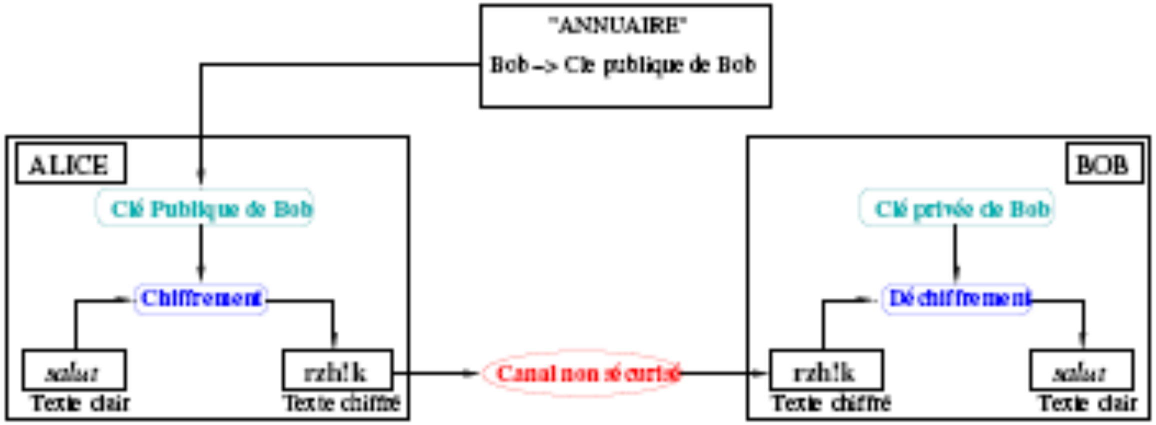
  - Comment établir une communication sécurisée entre deux entités sans échange préalable de clé?

Motivations (suite)

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

Pré-requis mathématiques

- Théorème d'Euclide
- Théorème de Bezout

  (Calcul pratique): Algorithme d'Euclide Etendu
- La fonction indicatrice d'Euler
- Propriétés de la fonction d'indicatrice d'Euler (voir complément de cours)
- Petit théorème de Fermat

Pré-requis mathématiques(2)

- Exponentiation rapide modulaire: calcul de a^e mod n
- Basé sur la remarque suivante :
  - Si e est pair, $a^e = (a^{e/2})^2$
  - **Si e est impair,** $a^e = (a^{(e-1)/2})^2 \cdot a$

- Algorithme d'exponentiation rapide modulaire
- Décomposer e en binaire $e = \sum_{i=0}^{k} e_i 2^i$
- Calcul de $\{a^{2^i} \mod n\}_{0 \leq i \leq k}$
  - Utiliser la relation : $a^{2^{i+1}} = \left(a^{2^i}\right)^2 \mod n$
- En déduire : $a^e = \prod_{i=0}^{k} \left(a^{2^i}\right)^{e_i}$

Exponentiation rapide: exemple

- Calcul de **51447 ^{21} mod 17 (E)**

51447 = 3026 x 17 + 5 donc (E) $\Leftrightarrow$ 5^{21} mod 17

- **Décomposition en binaire: $21=2^4+2^2+2^0$**
- **Calcul de** $\{5^{2^i} \mod 17\}_{0\leq i\leq 4}$
  - **i=0 :** $5^{2^0} \equiv 5 \mod 17$
  - **i=1 :** $5^{2^1} = 5^2 = 25 \equiv 8 \mod 17$
  - **i=2 :** $5^{2^2} = 8^2 = 64 \equiv 13 = -4 \mod 17$

Exponentiation rapide: exemple

- i=3: $5^{2^3} = (-4)^2 = 16 = -1 \mod 17$
- i=4 : $5^{2^4} = (-1)^2 \equiv 1 \mod 17$
- On en déduit :

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

Cryptosystème RSA (1977)

- Développé au *MIT* (Massachussets Institute of Technology) en 1977 par Ronald Rivest, Adi Shamir et Leonard Adleman;
- le système *RSA* convient parfaitement pour le chiffrement des fichiers stockés dans les mémoires de masse des ordinateurs personnels.
- RSA préserve la confidentialité des messages électroniques.

Cryptosystème RSA (1977)

Génération des clés

- Bob choisit au hasard 2 nombres premiers p et q
  - Bob calcule $n=pq$
  - Indicatrice d'Euler: $\varphi(n) = (p-1)(q-1)$

- Bob choisit au hasard un entier c (impair) tel que:

  $1 < c < \varphi(n)$

  $\mathrm{pgcd}(c, \varphi(n)) = 1$

- Bob calcule alors l'entier $1 < d < \varphi(n)$ tel que

  $cd = 1 \bmod \varphi(n)$

Cryptosystème RSA(2)

- Clé publique : (n,c) (c : exposant RSA, n module RSA
- Clé secrète : d

Chiffrement RSA

- Alice récupère la clé publique (n, c) de Bob
- Pour chiffrer le message M entier tel que 0<=M<n:

$$C = M^c \bmod n$$

- Alice envoie le message chiffré à Bob.

Cryptosystème RSA (3)

Déchiffrement RSA

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

$$C^d = M \bmod n$$

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

$$\begin{aligned} \mathbf{C^d} &\equiv \mathbf{M^{cd} \bmod n} \\ &\equiv \mathbf{M^{1+k.\,\varphi(n)} \bmod n} \\ &\equiv \mathbf{M.\,(M^{\varphi(n)})^k \equiv M \bmod n = M} \end{aligned}$$

Cryptosystème RSA: Exemple

- Prenons p = 47 et q= 59
- On calcule $n=p.q=47.59 = 2773$
- On choisit c, premier par rapport à $\varphi(n)$.Ex: c=17.
- On calcule alors, par l'algorithme d'Euclide étendu, d :

  $d.c \equiv 1 \mod (p-1)(q-1)$, soit d= 157

Clé publique : (c,n)= (17, 2773)

Clé privée : d=157

- Chiffrement du message M =01000010 =66:

  $$\mathbf{C \equiv M^c \bmod n \equiv 66^{17} \bmod 2773 = 872}$$

- Déchiffrement de C:

Cryptosystème RSA: Exemple

- Déchiffrement de C:

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

$\equiv$

Sécurité de RSA

- Le vrai but de l'attaquant : découvrir le texte calir
- Calculer d à partir de (n,e) $\Leftrightarrow$ factoriser n
- Limites actuelles de factorisation: $\simeq$ 200 chiffres
- Record actuel : RSA 200 (200 chiffres décimaux)
  - Bahr, Boehm, Franke and Kleinjung –9 mai 2005.
- Si la clé secrète d est petite (de l'ordre de $n^{1/4}$):
  - Attaque utilisant l'algorithme des fractions continues(algorithme LLL) (Voir TD)
  - Permet de calculer d à partir de n et e.

DLP & ElGamal

- Autre problème difficile : Discret Logarithme Problem
- Définition : (logarithme discret)

Soit (G, .) un groupe multiplicatif, g un élément d'ordre n de G et h dans $<g>= \{ g^i\}_{0<=i<n}$ (groupe monogène d'ordre n).

Alors le logarithme discret de h en base g, noté:

$\log_g(h)$, est l'unique entier x tel que $h=g^x$ $(0<= x < n)$.

- DLP: consiste alors à résoudre le problème suivant:

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

Cryptosystème El Gamal

- Données publiques pré-requises:
  - (G, .) le groupe multiplicatif où G=(Z//pZ)*et p premier
  - Et g un élément primitif de G mod p.
- Génération des clés:
  - Bob choisit **a** $\in$ [0, p-2] et calcule $A=g^a$ mod p.
  - Clé publique: (G, g, p, A)
  - Clé secrète : a

Cryptosystème El Gamal (2)

- Chiffrement:

  Alice souhaite envoyer à Bob le message $0 \leq M \leq p-1$

  - Alice récupère la clé publique (G, g, p, A) de Bob.
  - Alice choisit au hasard $k \in [0,p-2]$
  - Le message chiffré qu'Alice envoie à Bob est:

    $C=(y_1, y_2)$ avec:

$$\begin{cases} y1 = g^K & \mathrm{mod}(p) \\ y2 = M \, . \, A^k & \mathrm{mod}(p) \end{cases}$$

Cryptosystème El Gamal (3)

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

$$M = y_2 . y_1^{\,p-1-a}$$

- En effet: posons n=p-1

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

Sécurité du Cryptosystème Al Gamal

- Résoudre DLP dans G $\Rightarrow$ casser Al Gamal ds G
- l'attaquant peut alors calculer a à partir de A
- La réciproque n'est pas encore prouvée.

Problème: DH & Clés

- Problème DH:

  Etant donné un grand nombre premier p et une racine primitive de p, i.e un nombre g dont les puissances modulo p engendrent $\mathbb{Z}_p$ -{0}.

  – ***Le problème de DH*** est de trouver, étant donnés A, B dans Z non divisibles par p, l'entier C vérifiant:

    $C= g^{ab}$ avec $g^a=A$ et $g^b=B$ mod p

  avec a,b dans Z (inconnus)

  - Remarque: A et B sont dans:{1,2,…,p-1}

Clés de Diffie-Hellman

- Alice et Bob veulent partager une clé secrète K.

  Ils choisissent un p premier >>0, g racine primitive de (Z// pZ/)*avec 2<=g<=p-2 tel que l'ordre de g soit très élevé.

  On suppose que les données p et g sont publiques
  - **Alice choisit un entier $0 \leq a \leq p-2$**
  - **Alice calcule $A = g^a$ et l'envoie à Bob**
  - **Bob choisit un entier $0 \leq b \leq p-2$ 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 clé commune est donc**

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

Protocole d'échange de clés

Alice		Bob
génère a		génère b
$A = g^a \mod p$		$B = g^b \mod p$
	$\xrightarrow{A}$	
	$\xleftarrow{B}$	
(dispose de [a, A, B, p])		(dispose de [b, A, B,p])
Clé secrète: $K = B^a \mod p$		Clé secrète: $K = A^b \mod p$

Protocole d'échange de clés

- Le procédé d'échange des clés de Diffie et Hellman ne constitue pas à proprement parler un cryptosystème à clé publique.
- Il autorise « simplement » deux correspondants à convenir d'une clé de chiffrement (utilisable ultérieurement pour communiquer à l'aide d'un chiffre à clé secrète) sans avoir à se préoccuper de la confidentialité de cet échange.

Sécurité

- Problème de DH:
  - Connaissant p, 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 $\Longleftrightarrow$ résoudre DH