Cours du 14.12.2020

lundi, 14 décembre 2020 18:37

Cryptographie :

Nombre de César => décalage de l'alphabet.

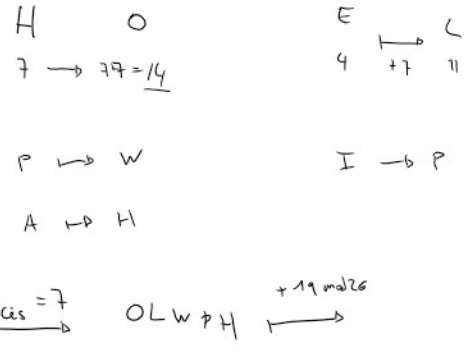
N = 3 => A devient D, B -> E.... W -> Z, X -> A ("cyclique")....

Considérons le nombre sous forme d'indices dans le tableau des lettres.
26 lettres => la lettre à la position i devient la lettre à la position
A est la lettre à la position 0
B 1
Z est la lettre à la position 25.

Lettre i devient lettre à la position (i + N) mod 26 !!!!!

Exemple : N = 7

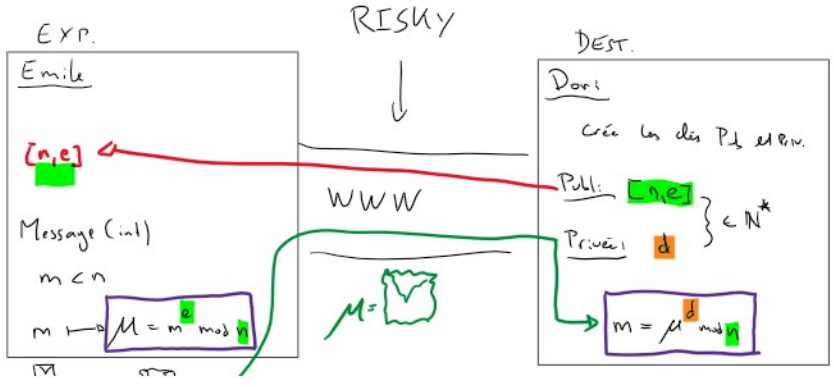
HEPIA



C'est une méthode dite SYMETRIQUE car la clé de chiffrement ET de déchiffrement sont les mêmes. Et il faut la communiquer d'une manière ou d'une autre !

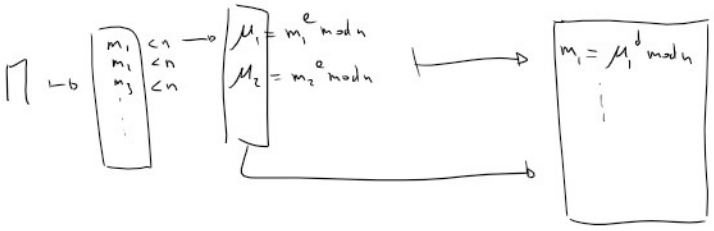
RSA : **R**ivest, **S**hamir et **A**dleman (brevet déposé au MIT en 1977)

RSA est une méthode de chiffrement ASYMETRIQUE.
Il y a 2 clés : une dite ***publique*** (envoyée en clair) utilisée pour CHIFFRER les données, et l'autre dit ***PRIVEE***, utilisée pour DECHIFFRER les données.
Seule 1 des clés (publique) est publiée, l'autre n'est jamais divulguée.



RSA ⇒ on utilise 2x Exponentiation RAPIDE !
↳ $O(\log(n))$

Note : quand on écrit du texte, il est stocké sous forme de bits.
On peut ré-interpréter les bits des charactères comme des nombres entiers.
Quand on a un message en texte, on le convertit en nombre via la représentation binaire (encodage, du ASCII ou du UTF-8).
On découpe le texte sous forme de blocs dont la conversion donne un nombre ne dépassant pas n (le n de la clé publique).



Texte ↳ INT

Pour le RSA-X, le n est composé de X bits.
En principe, nous utilisons le RSA-1024 => $n \approx 2^{1024} \approx 10^{310}$

Exponentiation rapide (chiffrer ET déchiffrer) nécessite $\log(10^{155}) = 1024$ itérations.
Sur une machine standard, cela prend moins de 1 ms !!!!

Comment génère les clés : (TRAVAIL DU DESTINATAIRE, DORI !!)

1. Choisir (au hasard) 2 nombres premiers ***p*** et ***q*** (d'ordre de grandeur de 512 bits pour le RSA-1024).
2. $n = p \times q$
3. Calcule $\varphi(n) = (p-1) \cdot (q-1)$
4. On choisit e qui est un nombre premier avec $\varphi(n)$
5. On calcule par Euclide étendu les coefficients de Bézout de

$\text{PGCD}(e, \varphi(n)) = 1 = e \cdot x + \varphi(n) \cdot y$ (point 4)
$\Rightarrow d = x \bmod \varphi(n)$
$\Rightarrow e \cdot d = 1 - \varphi(n) \cdot y$

Pourquoi cela fonctionne : (théorème de Fermat !!)

Dori : $\mu^d \bmod n$ ↳ 1

$\mu = m^e \bmod n$

$= (m^e \bmod n)^d \bmod n \underset{n}{\equiv} (m^{e \cdot d}) \bmod n$ (Dist. du mod.) ($(m^e)^d$)

$\underset{h}{\equiv} (m^{1-\varphi(n)\cdot y}) \bmod n = \left(m \cdot (m^{\varphi(n)})^{-y}\right) \bmod n$ (par déf. de d (Bachet-Bézout))

$= (m \bmod n) \cdot \left(m^{\varphi(n)} \bmod n\right)^{-y}$

d (Bachet-Bezout)

$\equiv_n (m \bmod n) \cdot \left(m^{\varphi(n)} \bmod n \right)^{-y}$

(distrib. mod)

$= 1$ (si m premier avec n !) FERMAT

$\equiv_n m \bmod n \cdot 1^{-y} = m \bmod n$

$= m$ (car $m < n$ et $m > 0$)

(égalité)

On vient de pouver que $m = m$!

$m^e \bmod n$ ⟶ $M^d \bmod = m$!

M

CQFD!

Conditions: $m < n$ et m premier avec n !

d l'inverse modulaire de e mod $\varphi(n)$ compris entre 1 et n.

Exemple :

1. p = 23 et q = 19
2. n = 23 * 19 = 437
3. $\varphi(n) = (23 - 1)*(19-1) = 396$
4. Choisissons e premier avec 396 : e = 35
5. Calcul Euclide Etendu :
   PGCD(396, 35) = 1 = 396 * *16* + 35 * (*-181*)
   d = (-181) mod 396 = 215.

[n, e] = [437, 35]

d = 215

Envoyons un message codé : "Bonjour Lulu" -> ("UTF-8") : m = 72

(StrToInt)

Emile:

$M = 72^{35} \bmod 437 = 147$

[437, 35]

www

147

↳ intToStr = Ç"KDIP0"*?`!üEQö

Dori:

$m = 147^{215} \bmod n = 72$ ⟶ "Bonjour Lulu"

int2Str

9

8 calculs

Quelle est la sécurité du processus ?
Si on connait d, on a craqué le RSA ! (car on connaît n et e qui sont publics) => on peut calculer $m = \mu^d \bmod n$

Or, si on connaît $\varphi(n)$, on peut alors appliquer Euclide étendu de manière déterministe pour calculer d !
ATTENTION: l'indice d'Euler n'est PAS facile à calculer en général => il se base sur la FACTORISATION de n en facteurs premiers : ici c'est $(p-1)\cdot(q-1)$.

Craquer le RSA revient à trouver p et q, les deux diviseurs premiers de n.

Si on doit deviner d <= n = 437 on doit tester les facteur entre 2 et $\sqrt{437} \simeq 20$
On doit faire 20 test.
2 ne divise pas 437
3 non plus
4
5
6
...
19 OK => 19 = p et q = 437/19 = 23 !

↳ Si n a un facteur premier, au moins 1 diviseur est entre 2 et $\sqrt{n}$.

=> doit faire $\sqrt{n}$ test pour trouver p et q.

Algo du RSA

Expon. Rapide $O(\log(n))$

$n = 437$ 8 itérations

$\mu^d \bmod n$

Craquage du RSA

Factoriser n = p * q

$O(\sqrt{n})$

$n = 937$ 20 iter.

En vrai, RSA encode n sur 1024 bits donc $n \approx 2^{1024} \approx 10^{310}$

$O(\log(2^{1024})) = 1024$ itérations

$(\sim 1\,ms)$

Craquer

$O(\sqrt{n}) \approx 10^{155}$ calculs

Note : on estime qu'il y env 10^{80} atomes dans l'univers.

$O(\sqrt{n}) \approx (10^{80})^2$

Imaginons que la puissance de calcul de TOUS les ordinateurs de la terre 10^{30} opérations par seconde.
Il me faudrait pour craquer les RSA-1024 : $\frac{10^{155}}{10^{30}}\,s = 10^{125}\,s. \simeq 10^{117}$ années

$1\,an \approx 100 \cdot 10^{6}\,s = 10^{8}\,s$

$1 \text{ an} \approx 100 \cdot 10^6 \text{ s} = 10^8 \text{ s}.$

Moralité : c'est IMPENSABLE de craquer le RSA-1024 en force brute !!!

$n = p \cdot q$

https://eswys.ch/hepia_data/files/2.%20Archives/1.%20Maths%20%201/Exercices/enonce_tp.pdf

TP

La clé publique suivante :

$$n = 1190836873 = p \cdot q$$
$$e = 1051$$

1) Résoudre $n = p \cdot q$

2) Trouver $d = x \bmod \varphi(n)$ avec Euclide Etendu

$$\varphi(n) = (p-1) \cdot (q-1)$$

$$\text{PGCD}(e, \varphi(n)) = ex + \varphi(n)y = 1$$

$\downarrow$

$d = x \bmod \varphi(n)$

3) pour chaque bloc $M_i \longmapsto m_i = M_i^d \bmod n$

---Coded message-

$M_1 = 139625027$

4) Convertir $m_1 \xrightarrow[(\text{UTF-8})]{\text{IntToStr}}$ "texte"...

5) Préparez-vous à présenter votre TP lors d'un petit oral 18.01.2021 !

n=1190836873 e=1051 d=581220751
d=581220751 p=43223 q=27551

m_1 = 139625027 $m_1^{581220751} \bmod n$ = 2123082

UTF-8 "Je" (with edits)