APRES-MIDI Cours du 14.12.2020

lundi, 14 décembre 2020 13:02

RSA

Encode : $m \longrightarrow \mu = m^e \bmod n$

Décode : $\mu^d \bmod n = \left(m^e \bmod n\right)^d \bmod n \overset{\text{dist. modulo}}{\underset{n}{\equiv}} (m^e)^d \bmod n = m^{e \cdot d} \bmod n$

$\overset{*}{\underset{n}{\equiv}} m^{1-\varphi(n)\cdot y} \bmod n$

$m \cdot (m^{\varphi(n)})^{-y}$

Bachet-Bezout : $e \cdot d + \varphi(n) \cdot y = 1 = PGCD(e, \varphi(n))$

$\hookrightarrow \boxed{e \cdot d = 1 - \varphi(n) \cdot y}$ *

$\overset{\text{dist. mod.}}{\underset{n}{\equiv}} [m \bmod n] \cdot \left[\left(m^{\varphi(n)} \bmod n\right)^{-y}\right]$

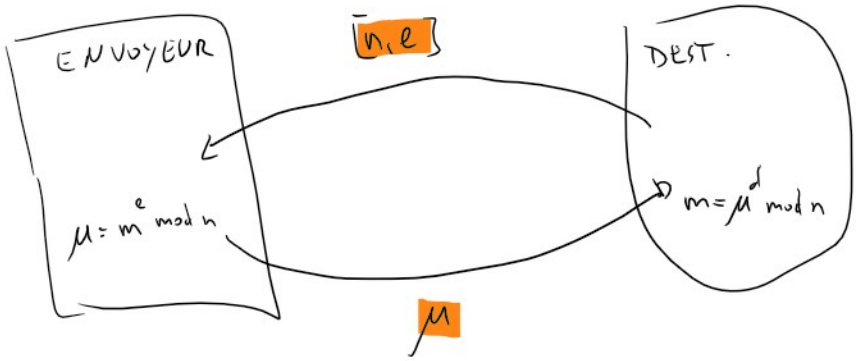
Fermat $\underset{n}{\equiv} 1$ si m et n premiers entre eux

$\underset{n}{\equiv} m \bmod n \cdot 1^{-y}$ ($1^{-y} = 1$)

$m \underset{n}{=} m \bmod n$ $\quad n > m$ (par choix de la taille bloc)

$\Rightarrow m = m \bmod n = m$

Pour être sûr que les fonctions, on s'assure que $m < \min(p, q)$



- génère p, q
- $n = p \cdot q$
- choisit e premier avec $\varphi(n)$
- calcule Bézout

$1 = e \cdot x + \varphi(n) \cdot y$

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

Là avec ces informations, comment peut-on retrouver m ? (message d'origine)

Il suffit de trouver $\varphi(n) = (p-1)(q-1)$

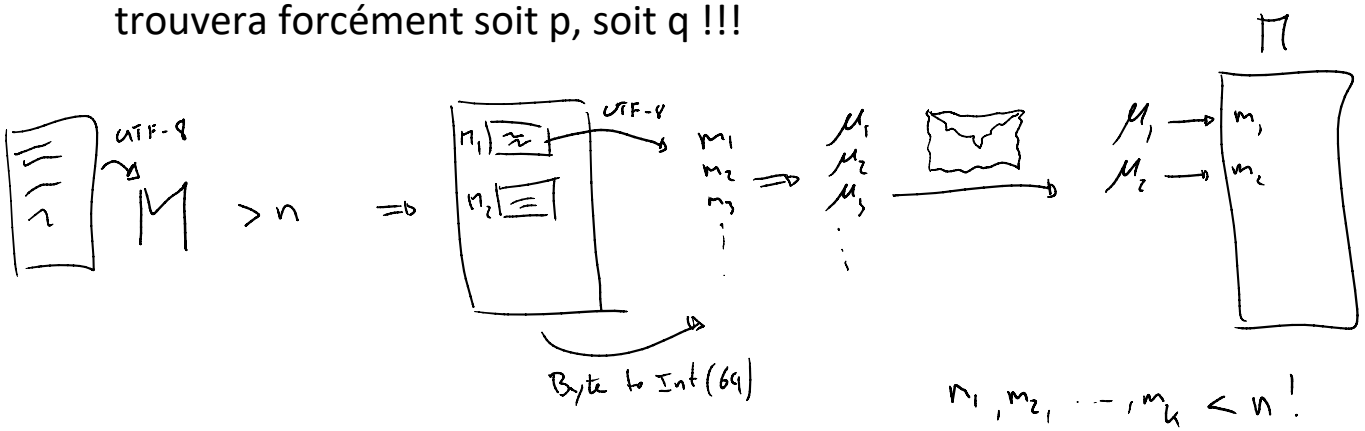
$\Rightarrow$ suffit de trouver p et q !

OR $n = p \cdot q$ $\Rightarrow$ FACILE ?!

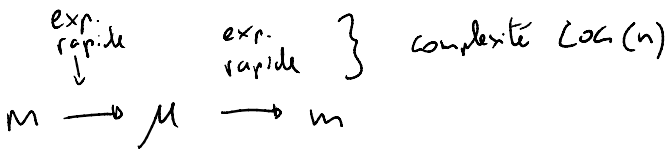
↳ il suffit de Factoriser n, et le RSA est craqué !!!

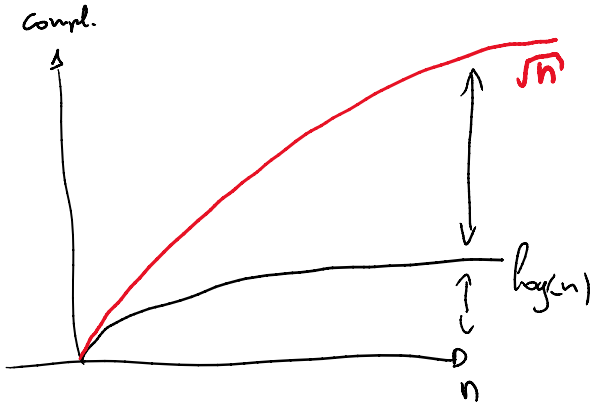
La complexité du RSA réside dans la "difficulté" de factoriser n.
RAPPEL : soit p soit que se trouvent dans 1,…,$\sqrt{n}$

Il suffit donc de tester TOUS les nombres de 1 à $\sqrt{n}$ et on trouvera forcément soit p, soit q !!!



Etude de complexité !





doubler la taille de n ajoute 1 itération !

Factoriser n : il faut faire $\sqrt{n}$ tests

$\Rightarrow$ complexité $O(\sqrt{n})$

RSA-512 n est codé sur 512 bits

$n \approx 10^{155}$ $\Rightarrow$ factoriser n il faut env. $\sqrt{10^{155}}$ calculs

$\approx 10^{77}$ calculs **pour craquer le RSA**

(Le nombre d'atomes dans l'univers $\approx 10^{80}$)

(1 Megaflop $\approx$ 8'000 ans!)

Or $\log_{10}(10^{155}) = 155$ $\quad \log_2(10^{155}) \approx 512$ calculs **chiffrer / déchiffrer**

$<$ 1 ms à faire !

$n = p \cdot q$ Difficile à résoudre si n très grand !

RSA-1024 (standard) RSA-2048 voir même RSA-8192.

$2^{1024} \approx 10^{310}$ $\Rightarrow$ Chiffrer-déchiffrer => 1024 itérations $\approx$ 1 ms

Craquer => factoriser => tester tous les impairs de 2 à $\sqrt{n}$

$0.5 \cdot \sqrt{10^{310}} = 0,5 \cdot 10^{155} \approx 10^{154}$

Estimons que nous ayons 10^{30} calculs par seconde
(c'est LARGEMENT plus que la puissance de calcul totale
de TOUS les ordinateurs de la terre !!!)

il nous faudrons $\frac{10^{154}}{10^{30}}$ s $\approx 10^{120}$ s. $= \underline{10^{112} \text{ ans!}}$

(1,5 an par chaque atome de l'univers !!)

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

Autrement dit : il est IMPENSABLE de craquer le RSA-1024 via la

force brut, et encore moins via un calcul aléatoire !

$n \approx 2^{1024}$ tient pas sur des 32 bits ▽

$n \approx 10^{32}$ exp: $\underbrace{m^2 \bmod n}_{\leq n-1} \approx n^2 < 10^{64}$ $\rightarrow n < 10^{32}$

TP:

Données : $n, e, \mu_1, \mu_2, \ldots$

A trouver :

1. p et q tels que $n = p \cdot q$
2. $\varphi(n) = (p-1) \cdot (q-1)$

   Bachet-Bézout $1 = e \cdot \underbrace{x}_{d = x \bmod \varphi(n)} + \varphi(n) \cdot y$ ⚠ $e < \varphi(n)$!!!
3. Exp. rapide $m_1 = \mu_1^d \bmod n$
4. int 64 (bytes) $\xrightarrow[\text{Little Endian}]{\text{UTF8}}$ string $M = m_1 m_2 m_3 \ldots$

① $= 43223 \times 27551$

② $d = 581220751$

③ $m_1 = (\mu_1)^d \bmod n = 212382$

④ int 64 $i = 212382$ $\xrightarrow[\text{UTF-8}]{\text{byte to string}}$ "Je␣" (100101...)

espace !

niklaus.eggenberg@hesge.ch