RSA : security basis and practical aspects

Sandra Marcello

6 mars 2022

Plan

Introduction

- RSA's security relies on the factorization Problem.
- El gammal's security relies on the DLP (Discrete Logarithm Problem.

Goal :Description of some algorithms against these problems.

Basic analysis

- Factorisation Problem : Giving $n = pq$ with p, q prime numbers .Find p and q.
  1. Knowing p and q. Compute $\phi(n)$.
  2. Extended Euclidean algorithm : find the private key thanks to the public one.
- Necessary to hide $\phi(n)$ $\phi(n) = (p-1)(q-1)$ Writing $\frac{q=n}{p}$, we obtain the following equation :

$$p^2 + p(\phi(n) - 1 - n) + n,$$

whose roots are p and q.

Pollard's $p-1$: idea

- Let p be a prime number such that $n = 0[p]$.
- Let q be a prime number, with $p-1 = o[q]$
- Using a "random "list of elements L ($\mathbb{F}_n$)
- As n is not a prime : existence of collision modulo p.
- In case of collision $GCD(x_j - x_i, n) = p$

Pollard's ρ-method

Main ideas :$n = p.q$, with

- Using a "random "list of elements L ($\mathbb{F}_n$)
- As n is not a prime : existence of collision modulo p.
- In case of collision $GCD(x_j - x_i, n) = p$

Pollard's ρ-method : function f

- Consider $f(x) = x^2 + a[n]$, usually $a = 1$.
- Build the List : L
- Compute $GCD(x_i, x_{2i})$, if $GCD(x_i, x_{2i}) > 1$ then $GCD(x_i, x_{2i}) = p$ or $GCD(x_i, x_{2i}) = q$

Pollard's ρ-method : description

- Consider $f(x) = x^2 + a[n]$, usually $a = 1$.
- Build the List : L
- Compute $GCD(x_i, x_{2i})$, if $GCD(x_i, x_{2i}) > 1$ then $GCD(x_i, x_{2i}) = p$ or $GCD(x_i, x_{2i}) = q$

Example : $n = 7171 = 71 * 101$, $f(x) = x^2 + 1$ and $x_1 = 1$.

Pollard's ρ-method : pseudo-code

Pollard's Rho : Input (n, x_1)

- $x \leftarrow x_1$.
- $x' \leftarrow f(x)[n]$
- $p \leftarrow GCD(x - x', n)$
  While $p = 1$ Do
  - $x \leftarrow f(x)[n]$
  - $x' \leftarrow f(x')[n]$
  - $p = GCD(x - x', n)$

  If $p = n$ return "Fail" Else return "p".

Shanks' algorithm

Compromis Espace-Temps

Shanks's algorithm : (G,n,α, β)

1. $m \leftarrow \lceil \sqrt{n} \rceil$
2. For $j \leftarrow o$ until $m-1$ Do : α^{mj}
3. Build a list L_1 (j, α^{mj})
4. For $i \leftarrow o$ until $m-1$ Do : $\beta\alpha^{-i}$
5. build a List L_2 $(i, \beta\alpha^{-i})$
6. Find a collision (on the second coordinate) $(j, y$ et (i, y).
7. $\log_\alpha(\beta) \leftarrow (mj+i)[n]$.

Pollard Rho algorithm

Input : Let G be a cyclic group. (mandatory)
Same ideas (cf. Pollard Rho algorithm factorization)

- Split your group G in 3 subsets (partition). ($S_0 = \{x \in G$ such that $x = 0[3]\}$,.... if $G = (\mathbb{Z}/p\mathbb{Z})^*$)
- Define a "random " function on G thanks to these 3 subsets.
- Buid a List L
- Find a collision

Pollard Rho algorithm

Pollard Rho algorithm for DLP : (G,n,α, β)
function f : $f(x, a, b)$ If $x \in S_0$ then $f \leftarrow (\beta.x, a, (b+1)[n])$
else if $x \in S_1$ then $f \leftarrow (x^2, 2a[n], 2b[n])$
else $f \leftarrow (\alpha.x, (a+1)[n], b)$
return (f)

Pollard Rho algorithm

Main :

1. Define $G = S_0 \cup S_1 \cup S_2$
2. $(x, a, b) \leftarrow f(1, 0, 0)$
   $(x', a', b') \leftarrow f(x, a, b)$
3. While $x \neq x'$ Do :
   - $(x, a, b) \leftarrow f(x, a, b)$
   - $(x', a', b') \leftarrow f(x', a', b')$

   If $(b' - b) \wedge n \neq 1$
   then Return "fail" else return $((a - a')(b' - b)^{-1}[n])$

Exponentiation

- Algorithme naíf : $x^n = x.x \cdots x.x$ $n-1$ multiplications successives .
- Optimisation : Pour $n = 2^k$

$$x^n = \left(\cdots ((x^2)^2)^2 \cdots \right)^2.$$

k multiplications élémentaires au lieu de $2^k - 1$.
Complexity :$\mathcal{O}((\log(x))^k)$.

Exemple : x^{15} ?

Fast exponentiation (square and multiply)

Goal : $z = x^c[n]$

Binary decomposition : Let $c \in \mathbb{N}$ with $c = \sum_{i=0}^{l-1} c_i 2^i$

Fast exponentiation(x, c, n) :

$z \leftarrow 1$

For $i \leftarrow l - 1$ to 0

Do

- $z \leftarrow z^2[n]$
- **If** $c_i = 1$ **then** $z \leftarrow (z \times x)[n]$

Return (z)