Cryptographie : le système RSA

Fondements mathématiques et principe

1. Introduction au RSA

Le RSA (Rivest-Shamir-Adleman, 1977) est un système de cryptographie **asymétrique**. Cela signifie qu'il utilise deux clés différentes :

— Une **clé publique** pour chiffrer un message.
— Une **clé privée** pour le déchiffrer.

La sécurité du RSA repose sur la difficulté pratique de factoriser un très grand nombre entier en ses facteurs premiers.

2. Le théorème fondamental du RSA

Le fonctionnement du RSA découle directement du théorème suivant :

Théorème 1. *Soient p et q deux nombres premiers et soit $n = pq$. On considère $e \in \mathbb{N}$ tel que $1 < e < (p-1)(q-1)$ et $\mathrm{PGCD}(e, (p-1)(q-1)) = 1$. Alors :*

(i) Il existe un unique $d \in \mathbb{N}$ tel que $1 \leq d \leq (p-1)(q-1)$ et $ed \equiv 1 \pmod{(p-1)(q-1)}$.

(ii) Pour tout entier $m \in \mathbb{N}$, $m^{ed} \equiv m \pmod{n}$.

Explication des notations

— **p** et **q** : deux nombres premiers, grands et secrets.
— **n = pq** : le **module**, public.
— **e** : l'exposant de chiffrement (public), premier avec $\varphi(n) = (p-1)(q-1)$.
— **d** : l'exposant de déchiffrement (privé).

3. Démonstration du théorème

Démonstration de (i) : Existence et unicité de d

La condition $\mathrm{PGCD}(e, (p-1)(q-1)) = 1$ signifie que e est inversible modulo $(p-1)(q-1)$. D'après le théorème de Bachet-Bézout, il existe des entiers u et v tels que :

$$eu + (p-1)(q-1)v = 1$$

En réduisant cette équation modulo $(p-1)(q-1)$, on obtient :

$$eu \equiv 1 \pmod{(p-1)(q-1)}$$

L'entier u (que l'on prend dans l'intervalle $[1, (p-1)(q-1)]$) est l'inverse d recherché. L'unicité de d dans cet intervalle est une propriété de l'unicité de l'inverse modulo un nombre.

Démonstration de (ii) : La relation de déchiffrement

Puisque $ed \equiv 1 \pmod{(p-1)(q-1)}$, il existe un entier k tel que $ed = 1 + k(p-1)(q-1)$. Montrons la congruence séparément modulo p et modulo q.

— **Modulo p :**
  — Si $p \mid m$, alors $m \equiv 0 \pmod{p}$, donc $m^{ed} \equiv 0 \equiv m \pmod{p}$.
  — Si $p \nmid m$, d'après le petit théorème de Fermat, $m^{p-1} \equiv 1 \pmod{p}$. Alors :

$$m^{ed} = m^{1+k(p-1)(q-1)} = m \cdot (m^{p-1})^{k(q-1)} \equiv m \cdot 1^{k(q-1)} \equiv m \pmod{p}$$

— **Modulo q :** Par un raisonnement symétrique, on obtient $m^{ed} \equiv m \pmod{q}$.

Puisque $m^{ed} - m$ est divisible par p et par q, et que p et q sont premiers distincts, il est divisible par leur produit $n = pq$. Ainsi :

$$m^{ed} \equiv m \pmod{n}$$

Application pratique et mise en œuvre

1. Algorithme de génération des clés

Pour qu'Alice puisse recevoir des messages chiffrés, elle doit générer sa paire de clés :

1. **Choix des nombres premiers** : Choisir deux très grands nombres premiers distincts, p et q.
2. **Calcul du module** : Calculer $n = p \times q$.
3. **Calcul de l'indicatrice d'Euler** : Calculer $\varphi(n) = (p-1)(q-1)$. Cette valeur doit rester secrète.
4. **Choix de l'exposant public** e : Choisir un entier e tel que $1 < e < \varphi(n)$ et $\text{PGCD}(e, \varphi(n)) = 1$. Une valeur courante est $e = 65537$.
5. **Calcul de l'exposant privé** d : Calculer l'unique entier d tel que :

$$e \cdot d \equiv 1 \pmod{\varphi(n)}$$

   On utilise pour cela **l'algorithme d'Euclide étendu**.
6. **Destruction des valeurs sensibles** : Il est conseillé de ne plus conserver p, q et $\varphi(n)$.

Résultat :
— **Clé publique** : le couple (n, e), distribuée à tout le monde.
— **Clé privée** : le couple (n, d), gardée secrète par Alice.

2. Protocole de chiffrement et déchiffrement

Chiffrement (par Bob)

1. Bob obtient la clé publique d'Alice (n, e).
2. Il convertit son message en un nombre entier m tel que $0 \leq m < n$.
3. Il calcule le message chiffré c :

$$c \equiv m^e \pmod{n}$$

4. Bob envoie c à Alice.

Déchiffrement (par Alice)

1. Alice reçoit le message chiffré c.
2. Elle utilise sa clé privée (n, d) :

$$m \equiv c^d \pmod{n}$$

3. Elle reconvertit le nombre m en texte.

Pourquoi cela fonctionne-t-il ?

$$c^d \equiv (m^e)^d \equiv m^{ed} \equiv m \pmod{n}$$

d'après la partie (ii) du théorème.

3. Pourquoi est-ce sûr ?

La sécurité du RSA repose sur le problème de la **factorisation**. Pour retrouver m à partir de (n, e) et c, un attaquant a besoin de d. Pour trouver d, il a besoin de $\varphi(n) = (p-1)(q-1)$. Pour trouver $\varphi(n)$, il doit factoriser n pour retrouver p et q.

Si n est suffisamment grand (par exemple 2048 bits), factoriser ce nombre est impossible en pratique avec les ordinateurs actuels.

Résumé du processus

Étape	**Acteur**	**Action**	**Formule**	**Information utilisée**
1. Génération	Alice	Calcule les clés	$n = pq$, $ed \equiv 1$	p, q (secrets)
2. Distribution	Alice	Envoie	-	n, e (publics)
3. Chiffrement	Bob	Chiffre	$c = m^e \mod n$	n, e (publics)
4. Transmission	Bob	Envoie	-	c (public)
5. Déchiffrement	Alice	Déchiffre	$m = c^d \mod n$	n, d (privés)