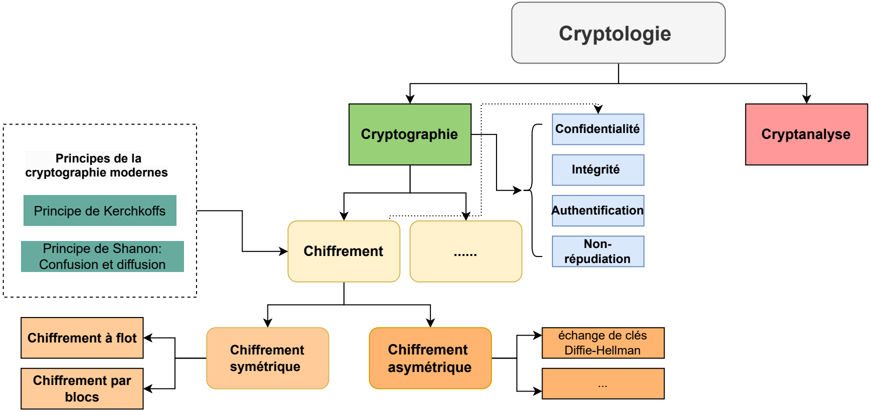
Initiation à la Cryptographie

Chiffrement asymétrique RSA

April 20, 2025

Awaleh HOUSSEIN

Recap de trois posts précédents [1] [2] [3]



[1] Premier post: `https://www.linkedin.com/feed/update/urn:li:activity:7284685244455157761/`

[2] Deuxième post: `https://www.linkedin.com/feed/update/urn:li:activity:7292307076343578624/`

[3] Troisième post: `https://www.linkedin.com/feed/update/urn:li:activity:7299899870608343040/`

RSA : Contexte historique

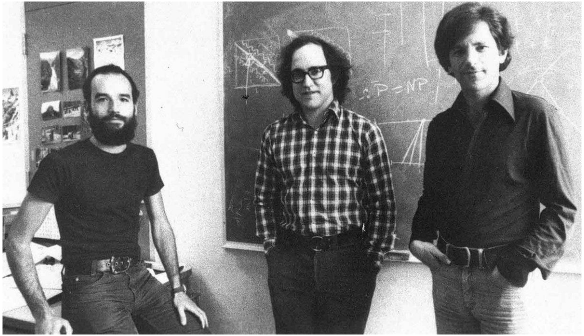


Figure: Les inventeurs de l'algo RSA. De gauche à droite: Adi Shamir, Ron Rivest, Len Adleman

- RSA inventé en 1977 par Rivest, Shamir et Adleman.

Quelques rappels mathématiques pour RSA [4]

- **Nombre premier** : Un entier $p \geq 2$ est premier si ses seuls diviseurs sont 1 et lui-même.
- **Congruence modulaire** : $a \equiv b \mod c \Leftrightarrow c$ divise $(a - b)$
  - Exemple : $17 \equiv 2 \mod 5$ car $17 - 2 = 15$ divisible par 5
- **Indicatrice d'Euler** $\phi(n)$:
  - Si p premier alors $\phi(p) = p - 1$
  - Si $n = p \cdot q$ (p et q premiers) alors $\phi(n) = (p - 1)(q - 1)$
- **Théorème d'Euler** : Si $a \wedge n = 1$ (a et n sont premier entre eux), alors :
  - $pgcd(a, n) = 1$ (a et n n'ont pas un diviseur commun)
  - $a^{\phi(n)} \equiv 1 \mod n$
- **Identité de Bézout** : Si $a \wedge b = 1$:
  - $\exists (x, y) \in \mathbb{Z}^2,\ ax + by = 1$

[4] Pour plus de détails: `https://www.di.ens.fr/~nitulesc/files/crypto3.pdf`

Fonction à sens unique

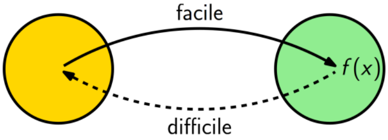


Figure: Illustration de la fonction à sens unique [5]

Principe

Trouver une fonction f telle que:

1. Calculer $y = f(x)$ est facile.
2. Calculer x à partir de y et f est difficile.

Question

Quelles fonctions peuvent satisfaire cette propriété ? Comment les construire ?

[5] Image source: `https://www.di.ens.fr/~nitulesc/files/crypto3.pdf`

Factorisation des entiers

Factorisation

- $(p, q) \longrightarrow p \cdot q$ facile.
- $n = p \cdot q \longrightarrow (p, q)$ difficile.

Multiplier deux nombres premiers est simple, mais l'opération inverse — factoriser leur produit — est difficile pour les entiers très grands.

- Plusieurs algorithmes permettent de factoriser les entiers.
- Leur complexité explose pour les très grands nombres.
- Exemples : divisions successives, méthode ρ de Pollard, etc.
- Pour aller plus loin : consulter le lien ci-dessous. [6]

[6] Cours détaillé sur la factorisation : `https://math.univ-lyon1.fr/~roblot/resources/factorisation.pdf`

Fonctionnement RSA: génération des clés

Génération des clés

- Choisir deux grands nombres premiers p et q
- Calculer $n = p \cdot q$ et $\phi(n) = (p-1)(q-1)$
- Choisir e tel que $1 < e < \phi(n)$ et $e \wedge \phi(n)$
- Soit d un entier qui satisfait $d \cdot e = 1 \mod \phi(n)$

$$e \cdot d + u \cdot \phi(n) = 1 \ \ (\textit{Bézout})$$

Clé publique

- $n = pq$: le module public
- e : exposant public

Clé privée

- $d = e^{-1}(\mod \phi(n))$
- Les nombres premiers p et q

Protocole RSA: chiffrement et déchiffrement

- Pour chiffrer un message en RSA:

$$C = Enc(pk = (e, n), m) = m^e \mod n$$

- Pour déchiffrer un message chiffré en RSA :

$$m' = Dec(sk = d, C) = C^d \mod n$$

Objectif

Démontrer que $m' = m$ lors du déchiffrement RSA, afin de prouver la validité de l'algorithme [a].

[a] La preuve détaillé: https://crypto.stackexchange.com/questions/2884/rsa-proof-of-correctness

Vérification de la déchiffrement RSA : $m = m'$

$$m' = C^d = (m^e)^d \mod n = m^{ed} \mod n$$

Puisque $ed \equiv 1 \mod \phi(n)$, il existe un entier k tel que :

$$ed = 1 + k\phi(n)$$

Donc,

$$m^{ed} = m^{1+k\phi(n)} = m \cdot (m^{\phi(n)})^k$$

D'après le théorème d'Euler, si $\gcd(m, n) = 1$, alors $m^{\phi(n)} \equiv 1 \mod n$, ce qui implique :

$$m' \equiv m \cdot 1^k \equiv m \mod n$$

Si $\gcd(m, n) \neq 1$ **:** on utilise le théorème des restes chinois pour prouver $m' = m$:

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

Sécurité de RSA

Hypothèse fondamentale

Factoriser $n = p \cdot q$ est **calculatoirement difficile**.

- Basé sur une fonction trappe : la factorisation de $n = p \cdot q$
- Casser RSA revient à factoriser n
- RSA est de moins en moins utilisé : nécessite des clés de grande taille
- Taille minimale recommandée : 2048 bits (env. 617 chiffres)

Menace quantique sur RSA

Risque majeur

L'algorithme de Shor[a] permet de factoriser n en temps polynomial avec un ordinateur quantique.

[a] Peter W Shor (1998). "Quantum computing". In: Documenta Mathematica.

- RSA devient obsolète avec un ordinateur quantique opérationnel.
- **Contre-mesures** :
  - Clés RSA plus longues (solution temporaire)
  - Cryptographie post-quantique (standardisation en cours par le NIST[7].

[7] Un aperçu détaillé sera présenté dans un prochain post

À retenir

- **RSA** est un algorithme de chiffrement à clé publique (asymétrique).
- **Clés RSA** :
  - Clé publique : $(n = p \cdot q,\ e)$
  - Clé privée d: $d \equiv e^{-1} \mod \phi(n)$
- **Chiffrement** : $C = m^e \mod n$
- **Déchiffrement** : $m' = C^d \mod n$
- Sécurité fondée sur la difficulté de factoriser $n = p \cdot q$
- Menace quantique : l'algorithme de Shor casse RSA avec un ordinateur quantique
- Recommandation : utiliser des algorithmes post-quantiques standardisés par le **NIST**[8]

[8] https://csrc.nist.gov/projects/post-quantum-cryptography