Introduction à la cryptographie

Vanessa VITSE

Université Grenoble Alpes

M1 Maths 2021

Section 8

Factorisation et RSA

Contexte

- RSA du nom de ses inventeurs : Rivest, Shamir et Adleman (1978)
- repose sur la difficulté de factoriser des grands entiers :
  la clef publique est un nombre composé, la clef privée est la décomposition en facteurs premiers de ce nombre
- l'entier composé doit être suffisament grand et sans petit facteurs (environ 3072 bits)
  record actuel : factorisation d'un nombre semi-premier de 830 bits en 2700 années CPU sur un simple coeur
- de plus en plus remplacé par des schémas basés sur le DLP

Contexte

- RSA du nom de ses inventeurs : Rivest, Shamir et Adleman (1978)
- repose sur la difficulté de factoriser des grands entiers :
  la clef publique est un nombre composé, la clef privée est la décomposition en facteurs premiers de ce nombre
- l'entier composé doit être suffisament grand et sans petit facteurs (environ 3072 bits)
  record actuel : factorisation d'un nombre semi-premier de 830 bits en 2700 années CPU sur un simple coeur
- de plus en plus remplacé par des schémas basés sur le DLP

On ne présente ici qu'une version basique, loin de l'implémentation réelle qu'il faudrait faire.

Description du RSA basique en chiffrement

1. **Génération de clefs :** Alice choisit deux grands entiers p, q de taille similaire et calcule $n = pq$.
   Alice choisit $e \in (\mathbb{Z}/\varphi(n)\mathbb{Z})^\times$ et calcule $d = e^{-1} \bmod \varphi(n)$.
   La clef publique d'Alice est (n, e) et sa clef privée d.

2. **Chiffrement :** Bob récupère la clef publique d'Alice dans un annuaire certifié. Il encode son message comme un entier m avec $1 \leqslant m < n$ et calcule $c = m^e \bmod n$ et envoie c à Alice.

3. **Déchiffrement :** Alice calcule $m = c^d \bmod n$ et le décode pour obtenir le message original.

Preuve

On distingue deux cas :

- **Si $m \wedge n = 1$:** alors $m \wedge p = 1$ et $m \wedge q = 1$
  Comme $ed = 1 + k(p-1)(q-1)$, Fermat donne
  $\begin{cases} m^{ed-1} = 1 \bmod p \\ m^{ed-1} = 1 \bmod q \end{cases}$ et on conclut avec les restes chinois

- **Si $m \wedge n \neq 1$:** alors ops $m \wedge p = p$
  Alors $m = m^{ed} = 0 \bmod p$. Ensuite soit $m \wedge q = 1$ et Fermat donne $m^{ed} = m \bmod q$, soit $m \wedge q = q$ et on toujours $m = m^{ed} = 0 \bmod q$.
  On conclut encore avec les restes chinois.

Complexité

1. **Chiffrement :** Bob récupère la clef publique d'Alice dans un annuaire certifié. Il encode son message comme un entier m avec $1 \leqslant m < n$ et calcule $c = m^e \bmod n$ et envoie c à Alice.
2. **Déchiffrement :** Alice calcule $m = c^d \bmod n$ et le décode pour obtenir le message original.

- essentiellement des exponentielles modulaires en $O(\log n^3)$
- e n'a pas besoin d'être choisi aléatoirement : on prend souvent $e = 2^{16} + 1$ pour avoir un petit poids de Hamming ($e = 3$ déconseillé)
- RSA en signature est simplement un RSA en chiffrement en "renversé" : la signature d'un message m s'obtient en calculant $s = H(m)^d$, et se vérifie en testant $s^e = H(m) \bmod n$.

Sécurité théorique de RSA

Problème RSA

Pour e premier à $\varphi(n)$ avec n composé, savoir calculer des racines e-ièmes modulo n.

Possible si :

- on sait factoriser n
- on connait $\varphi(n)$: mais alors on sait factoriser n...
- on connait clef de déchiffrement d : mais alors on sait factoriser n...

Sécurité théorique de RSA

Problème RSA

Pour e premier à $\varphi(n)$ avec n composé, savoir calculer des racines e-ièmes modulo n.

Possible si :

- on sait factoriser n
- on connait $\varphi(n)$: mais alors on sait factoriser n...
- on connait clef de déchiffrement d : mais alors on sait factoriser n...

Conjecture :
calculer des racines e-ièmes mod n $\iff$ factoriser n

Vrai pour $e = 2$ (schéma de Rabin), mais $2 \wedge \varphi(n) \neq 1$

Sécurité pratique de RSA

Nombreuses vulnérabilités :

- RSA en chiffrement est *malléable* $\rightarrow$ on rajoute un bourrage (OAEP)
- bourrage également nécessaire pour éviter que les petits messages ne soient pas vraiment chiffrés
- attaque "broadcast" si $e = 3$
- attaque de Wiener (cf TD4)
- etc...

...des milliers d'autres attaques, ce qui rend l'implémentation très difficile (actuellement, seul l'usage en signature de RSA n'est pas déconseillé)

Comparaison entre cryptosystèmes basés sur le DLP ou la factorisation

Nombreux algorithmes communs pour attaquer la factorisation et le DLP dans $(\mathbb{Z}/p\mathbb{Z})^*$:

- Pollard rho (cf TP5)
- Méthodes de calcul d'indices, crible quadratique (cf TP5)
- méthodes de crible de corps de nombres (NFS)
  $\hookrightarrow$ meilleurs algorithmes actuels,
  complexité sous-exponentielle en $2^{O(1)(\log n)^{1/3}(\log\log n)^{2/3}}$

Même sécurité dans les deux cas

Comparaison entre cryptosystèmes basés sur le DLP ou la factorisation

Nombreux algorithmes communs pour attaquer la factorisation et le DLP dans $(\mathbb{Z}/p\mathbb{Z})^*$:

- Pollard rho (cf TP5)
- Méthodes de calcul d'indices, crible quadratique (cf TP5)
- méthodes de crible de corps de nombres (NFS)
  ↳ meilleurs algorithmes actuels,
  complexité sous-exponentielle en $2^{O(1)(\log n)^{1/3}(\log\log n)^{2/3}}$

Même sécurité dans les deux cas

Mais : DLP généralisable à d'autres groupes, pas RSA

Crypto actuelle (ex : protocole `https`) surtout basée sur DLP dans groupes formés des points d'une courbe elliptique (de la géométrie algébrique appliquée !)