Niveau : 1 **Jaune**

La primitive RSA

1 Définition de la primitive RSA

La primitive RSA est construite à l'aide de deux grands nombres premiers p et q distincts et de tailles similaires. Le module n est le produit de ces deux grands nombres premiers :

$$n = pq.$$

Pour des applications cryptographiques, on conseille actuellement une taille d'au moins 2048 bits pour n et en tous les cas, supérieure à 1024 bits.

On introduit aussi la valeur de la fonction d'Euler de n :

$$\Phi(n) = (p-1)(q-1).$$

Soit e un entier $1 < e < \Phi(n)$ (en général tiré au sort, mais pas toujours) premier avec $\Phi(n)$:

$$\text{pgcd}(e, \Phi(n)) = 1.$$

On détermine alors l'entier d tel que $1 < d < \Phi(n)$ inverse de e modulo $\Phi(n)$:

$$ed \equiv 1 \quad \Phi(n).$$

On définit alors la primitive $\text{RSA}_{e,n}$ comme fonction de l'intervalle d'entier $[0, n-1]$ dans lui même par :

$$\text{RSA}_{(e,n)}(x) = x^e \mod n.$$

2 Propriétés de la primitive RSA

Théorème 2.1 *La fonction $\text{RSA}_{e,n}$ est une bijection de l'intervalle d'entiers $[0, n-1]$ sur lui-même, et la bijection réciproque est $\text{RSA}_{d,n}$.*

Preuve.C'est une conséquence immédiate du Théorème 3.2 énoncé dans la fiche "fichecrypto_ 106" sur le "Petit Théorème de Fermat et ses généralisations".

La primitive RSA est considérée comme une "**fonction à sens unique avec trappe**". C'est une fonction "facile à calculer", "difficile à inverser" en pratique pour les tailles de modules considérées (fonction à sens unique). Elle devient "facile à inverser" si on connaît un secret : d (la trappe), ou de manière équivalente en pratique, la factorisation de n. De ce fait, un système de chiffrement RSA simplifié fonctionne de la manière suivante. Tout utilisateur A possède une clé publique (n, e) et une

clé privée d. L'utilisateur A est le seul à connaitre d. En revanche tout le monde connaît la clé publique (n, e) de A. Si un utilisateur B veut envoyer un message m à A (m est assimilé pour simplifier à un entier vérifiant $0 \leq m < n$) il calcule le chiffré

$$c = m^e \mod n$$

et l'envoie à A. Le destinataire A utilise sa clé privée d pour déchiffré le message reçu :

$$m = c^d \mod n.$$

On utilise aussi la primitive RSA pour la signature.

3 Construire une primitive RSA

La construction des nombres premiers p et q se fait par tirage au sort d'un nombre impair puis par test probabiliste (Miller Rabin par exemple) pour savoir si le nombre tiré est premier ou pas. S'il n'est pas premier on retire un nombre impair au hasard et on teste à nouveau. On réitère ces opérations jusqu'à tomber sur un nombre premier. Quand le test de Miller Rabin décrète qu'un nombre est premier, il peut se tromper avec une probabilité très faible. On ne peut pas utiliser en boucle un test qui donne une réponse sûre. En revanche, une fois qu'un nombre a été décrété premier par le test de Miller-Rabin, il est possible si l'importance des sommes mises en jeu le commande et si on a du temps, de vérifier avec un test déterministe que le nombre construit est bien premier (ceci est réalisable en pratique, mais est long).

Pour construire e on peut penser à trois manières, qui sont effectivement utilisées dans les applications :
- **la plus courante :** On tire e au sort jusqu'à ce qu'il soit premier avec $\Phi(n)$;
- **pour des applications bien particulières :** on fixe e à priori (par exemple $e = 3$ et on construit p et q avec la contrainte $(p-1)(q-1)$ premier avec e ;
- **construction astucieuse :** On construit un nombre premier e plus grand que p et q et plus petit que $(p-1)(q-1)$.

4 Sécurité de la primitive RSA

Il existe des attaques de la primitive RSA dans des cas de mauvaise utilisation. On renvoie aux livres classiques de cryptographie ou à l'annexe 6 du cours de formation générale en cryptographie donné sur ce site pour avoir des détails sur diverses attaques de la primitive RSA.

5 Remarques

On peut utiliser $\lambda(n) = \text{ppcm}(p-1, q-1)$ au lieu de $\Phi(n)$. Les systèmes concrets enrobent la primitive RSA dans un mécanisme plus complexe de manière à rendre le chiffrement probabiliste (si on chiffre deux fois le même texte on obtient des résultats différents), de manière à traiter le cas de textes courts (chiffrements de clés) ou de textes longs qu'on doit découper. On a ainsi par exemple les systèmes RSA-OAEP ou RSA-KEM pour le chiffrement, RSA-PSS pour la signature.

Auteur : Ainigmatias Cruptos
Diffusé par l'Association ACrypTA