Licence 3 – 4TTI603U Université de Bordeaux
2024/2025

Feuille 3 : RSA

Exercice 1. Chiffrement RSA

1. Soit $n = pq$, où p et q sont des nombres premiers distincts. Le système RSA chiffre $m \in \mathbb{Z}/n\mathbb{Z}$ en $m^e \in \mathbb{Z}/n\mathbb{Z}$, où e est inversible modulo $\varphi(n)$. Puis on déchiffre $c \in \mathbb{Z}/n\mathbb{Z}$ en calculant $c^d \in \mathbb{Z}/n\mathbb{Z}$, où d est l'inverse de e modulo $\varphi(n)$.

   (a) Quelle est la clé publique ? La clé privée ?
   La fonction de chiffrement est

   $$f : \mathbb{Z}/n\mathbb{Z} \to \mathbb{Z}/n\mathbb{Z} m \mapsto m^e$$

   Donc le couple d'entiers (e, n) est la clé publique La fonction de déchiffrement est

   $$f^{-1} : \mathbb{Z}/n\mathbb{Z} \to \mathbb{Z}/n\mathbb{Z} m \mapsto m^d$$

   Donc le couple d'entiers $(d, \varphi(n))$ est la clé privée. On a utilisé la proposition suivante pour voir que f^{-1} est bien la fonction de déchiffrement de f.

   Proposition : *Soient p et q des nombres premiers distincts. Posons $n = pq$. Soit t un entier naturel congru à 1 modulo $\varphi(n)$. Alors, quel que soit $a \in \mathbb{Z}$, on a*

   $$a^t \equiv a \mod n.$$

   (b) Pourquoi vaut-il mieux prendre m dans $(\mathbb{Z}/n\mathbb{Z})^\times$? Soit x pris au hasard avec probabilité uniforme dans $\mathbb{Z}/n\mathbb{Z}$. Quelle est la probabilité pour que $x \in (\mathbb{Z}/n\mathbb{Z})^\times$? Si m n'est pas inversible modulo n alors cela signifie que m possède un facteur commun avec n, ce qui impact la sécurité du système RSA. La proportion des éléments de $\mathbb{Z}/n\mathbb{Z}$ qui sont inversibles est donnée par la fonction indicatrice d'Euler $\varphi(n)$, qui compte le nombre d'entiers inférieurs à n et premiers avec n. La probabilité pour qu'un élément x pris au hasard dans $\mathbb{Z}/n\mathbb{Z}$ appartienne à $(\mathbb{Z}/n\mathbb{Z})^\times$ est donc :

   $$\frac{\varphi(n)}{n}$$

   où $\varphi(n) = (p-1)(q-1)$ pour $n = pq$. Cette fraction est généralement proche de :

   $$1 - \frac{1}{p} - \frac{1}{q} + \frac{1}{pq}$$

   ce qui est souvent proche de 1 pour de grands nombres premiers p et q. Ainsi, la probabilité est relativement élevée, mais il reste préférable de s'assurer que $m \in (\mathbb{Z}/n\mathbb{Z})^\times$ lors du chiffrement.

(c) Montrer que le système est correct, c'est-à-dire que si c est un chiffré de m alors le déchiffrement de c redonne bien m (même si $m \notin (\mathbb{Z}/n\mathbb{Z})^\times$).
$f^{-1}(f(m)) = f^{-1}(m^e) = (m^e)^d = m^{ed} = m$

(d) La composée de deux chiffrements RSA de même module n est-elle un chiffrement RSA ? Oui, il suffit juste de revoir la définition de cryptosystème RSA.

(e) Dans cette question on fixe p et q deux nombres premiers distincts. Combien a-t-on de choix pour la clé publique ?
Le nombre de possibilités de clé publique est juste $\varphi(pq) = (p-1)(q-1)$.

2. Dans cette question on souhaite implémenter un système RSA avec $n = 221$.

(a) Calculer $\varphi(n)$. $n = 221 = 13 * 17$, donc $\varphi(221) = \varphi(13) \cdot \varphi(17) = 12 * 16 = 192$

(b) Vérifier que l'on peut choisir 7 comme exposant de chiffrement. 7 est premier avec 192, donc on a une chiffrement RSA.

(c) Chiffrer le message $m = 3$ pour cet exposant. $3^7 = 198 \bmod 221$

(d) Calculer la clé privée. Utiliser l'algo d'euclide étendu. L'inverse de $e = 7$ est 55.

(e) Déchiffrer le message $c = 198$. Utiliser l'exponentiation rapide pour trouver 3.

Exercice 2. Déchiffrement de RSA

Dans cette exercice, on montre comment on peut accélérer le déchiffrement du système RSA en utilisant le théorème des restes chinois. Soit $n = pq$ produit de deux nombres premiers distincts et $d \in \mathbb{N}$ premier avec $\varphi(n)$. On s'intéresse au calcul du déchiffrement $c^d \pmod n$.

1. On pose $m_p \equiv c^d \pmod p$, $m_q \equiv c^d \pmod q$, $d_p = d \mod (p-1)$ et $d_q = d \mod (q-1)$. Montrer que $m_p \equiv c^{d_p} \pmod p$ et $m_q \equiv c^{d_q} \pmod q$.
On écrit $d_p = (p-1)k + d$, donc $d = d_p - (p-1)k$. Par définition, $m_p = c^d \pmod p$. En substituant d par son expression précédente, on obtient :

$$m_p = c^{d_p-(p-1)k} = \frac{c^{d_p}}{c^{(p-1)k}}.$$

Or, $\forall x \in \mathbb{Z}/p\mathbb{Z}, x^{p-1} \equiv 1 \pmod p$, donc :

$$m_p \equiv c^{d_p} \pmod p.$$

Il en est de même pour m_q.

2. Soit le système dans $\mathbb{Z}/n\mathbb{Z}$:

$$\begin{cases} m \equiv m_p \pmod p \\ m \equiv m_q \pmod q \end{cases}$$

Justifier que si m est solution du système ci dessus alors $m \equiv c^d \pmod n$.
Il suffit d'appliquer le théorème des restes chinois

3. Comparer la complexité de cet algorithme de déchiffrement avec celle de l'algorithme usuel.

Supposons que n soit de N bits. Alors, l'exponentiation modulaire nous donne un algorithme de complexité $\mathcal{O}(N^3)$. Mais si nous utilisons le théorème des restes chinois pour accélérer le décodage, nous obtenons un algorithme de complexité

$$\mathcal{O}\left(\left(\frac{N}{2}\right)^3 + \left(\frac{N}{2}\right)^3\right) = \mathcal{O}\left(\frac{N^3}{4}\right).$$

Ainsi, cet algorithme est normalement au moins 4 fois plus rapide que l'exponentiation modulaire classique.

4. En utilisant cette méthode déchiffrer le message $c = 198$ pour $n = 221$ et $d = 67$.
   Appliquer ce qu'on a fait avant pour trouve $m = 107$.

Exercice 3. Dans RSA, connaître $\varphi(n)$ est équivalent à connaître p et q

Soit $n = pq$ produit de deux nombres premiers distincts.

1. Exprimer pq et $p + q$ en fonction de n et $\varphi(n)$. En déduire une méthode pour obtenir p et q lorsque l'on connait n et $\varphi(n)$.
   $p + q = n - \varphi(n) + 1$ et $pq = n$. Si nous posons $P(X) = X^2 - (n - \varphi(n) + 1)X + n$, alors p et q sont exactement les racines de ce polynôme.
2. Si $n = 17063$ et $\varphi(n) = 16800$, calculer p et q.
   Poser le polynôme à coefficients entiers et trouver ses deux racines entiers qui sont $p = 151$ et $q = 113$.

Exercice 4. Une attaque sur RSA : petit exposant public commun

On suppose que k personnes $B_1, \ldots, B_k$ ont pour exposant public RSA $e = 3$ avec des modules respectifs n_i, $1 \leq i \leq k$.

1. Pourquoi est-il raisonnable de supposer que les n_i, $1 \leq i \leq k$ sont deux à deux premiers entre eux ?
   Supposons qu'il existe deux entiers $1 \leq i, j \leq k$, tels que n_i et n_j ne sont pas premiers entre eux. Ainsi il existe un entier k qui divise n_i et n_j qui ne soit pas dans $\{\pm 1\}$. Ainsi on connaît alors une factorisation de $n_i = p_i \cdot q_i$, qui va nous dévoiler la clé secrète de B_i. Donc on suppose que les n_i sont deux à deux premiers entre eux.
2. Alice envoie les chiffrés d'un même message m à tous les B_i. Montrer qu'un attaquant peut déterminer m^3 modulo $P := \prod_{i=1}^{k} n_i$; en déduire qu'il peut calculer m si $P > m^3$.
   Alice envoie à tous les B_i le même message m mais crypté avec l'exposant $e = 3$. Donc tous B_i reçoit m^3 comme message. Si on suppose qu'un attaquant peut intercepter les messages, alors par une simple utilisation du théorème des restes chinois, on obtient un unique $m^3 \bmod P := \prod_{i=1}^{k} n_i$. Comme $m < n_i, \forall i$ alors $m^3 < P$. Ainsi en calculant juste la racine cubique de m^3 dans les entiers (i.e. dans $\mathbb{Z}$), on retrouve m.
3. Quelle est la valeur minimale de k qui permet de toujours faire cette attaque ?
   Pour assurer que cette attaque marche toujours. Il faut au moins trois Bob. Sinon on peut avoir m^3 pas forcément inférieur à P.

Exercice 5. Une attaque sur RSA : module commun

Bob et Catherine ont choisi le même module RSA n. Leurs exposants publics e_B et e_C sont distincts.

1. Expliquer pourquoi Bob peut déchiffrer les messages reçus par Catherine et réciproquement.

   Bob connait la factorisation de n donc connait une partie de la clé secrète $\varphi(n)$, et ça sera alors facile de calculer l'inverse de e_C de Catherine et vice versa.

2. On suppose que e_B et e_C sont premiers entre eux et qu'Alice envoie les chiffrés d'un même message m à Bob et à Catherine. Expliquer comment l'attaquant Oscar peut obtenir m.

   Comme e_B et e_C sont premiers entre eux. Alors il existe a et b des entiers tels que $e \cdot e_B + f \cdot e_C = 1$. Donc Oscar peut utiliser l'algorithme d'Euclide étendu pour retrouver $e \cdot e_B + f \cdot e_C = 1$. Ainsi Oscar peut calculer $(m^{e_B})^e (m^{e_C})^f = m^1 = m \bmod n$

3. Application : Bob a la clé publique $(221, 11)$ et Catherine la clé $(221, 7)$. Oscar intercepte les chiffrés 210 et 58 à destinations respectives de Bob et Catherine. Retrouver le message m.

   Il suffit de trouver $e, f \in \mathbb{Z}$ tels que $11a + 7b = 1$. On retrouve $11 * 2 - 3 * 7 = 1$, avec $e = 2$ et $f = -3$. Calculons alors $(m^{e_B})^e (m^{e_C})^f = (210)^2 \cdot (58)^{-3} = 121 \cdot (190)^{-1} = 190 * 57 = 46 \bmod 221$. Donc le message initial d'Alice est 46.

Exercice 6. Module RSA avec deux facteurs proches

Supposons que n soit un entier produit de deux nombres premiers p et q, $p > q$. On suppose que p et q sont proches, c'est à dire que $\epsilon := p - q$ est petit. On pose $t = \frac{p+q}{2}$ et $s = \frac{p-q}{2}$.

1. Montrer que $n = t^2 - s^2$.

$$\begin{aligned} t^2 - s^2 &= (\frac{p+q}{2})^2 - (\frac{p-q}{2})^2 \\ &= (\frac{p+q}{2} + \frac{p-q}{2})(\frac{p+q}{2} - \frac{p-q}{2}) \\ &= pq = n. \end{aligned}$$

2. Quelle est la taille de s ? Comparer t et $\sqrt{n}$.

   D'après la question 1, nous avons que $s^2 = t^2 - n$ alors $s = \sqrt{t^2 - n}$. Lorsque s est petit, alors s^2 est aussi petit, alors t^2 s'approche de n, tout en lui étant plus grand.

3. Montrer comment utiliser cela pour écrire un algorithme (de Fermat) factorisant n.

   En utilisant la question 2, on peut déduire qu'il existe un petit entier naturel u tel que

$$(\lfloor \sqrt{n} \rfloor + u)^2 - n \text{ soit un carré.}$$

   Afin de déterminer un tel entier u, on examine successivement les entiers

$$\lfloor \sqrt{n} \rfloor + 1, \lfloor \sqrt{n} \rfloor + 2, \ldots$$

   On teste pour chacun d'eux si son carré moins n est un carré. Si l'on y parvient, on obtient n comme différence de deux carrés, ce qui donne une factorisation de n.

4. Application : factoriser 11598781.
   Calculons la racine carrée de 11598781 et l'approcher par un entier $\lfloor\sqrt{11598781}\rfloor = 3405$. Et testons l'algorithme précédent.

$$\begin{aligned}(3405+1)^2 - 11598781 &\text{ n'est pas un carré}\\ (3405+2)^2 - 11598781 &\text{ n'est pas un carré}\\ (3405+3)^2 - 11598781 &\text{ n'est pas un carré}\\ (3405+4)^2 - 11598781 = 22500 = 150^2 &\text{ est un carré}\end{aligned}$$

   Donc en posant $t^2 = (3405+4)^2$ d'où $t = \frac{p+q}{2} = 3406+3$ et $s = \frac{p-q}{2} = 150$. Il suffit alors de résoudre un petit système et on retrouve que $p = 3559$ et $q = 3259$.

5. Déterminer le nombre d'itérations de l'algorithme en fonction de p et de n. Que se passe-t-il si $p - \sqrt{n} < \sqrt[4]{4n}$?
   Il suffit isoler le terme u qui est le nombre d'itérations.

$$\begin{aligned} & t = \lfloor\sqrt{n}\rfloor + u\\ \iff & u = (\frac{p+q}{2}) - \lfloor\sqrt{n}\rfloor\\ \iff & u = (\frac{p+\frac{n}{p}}{2}) - \lfloor\sqrt{n}\rfloor\\ \iff & u = \frac{p^2+n}{2p} - \lfloor\sqrt{n}\rfloor\end{aligned}$$

   Si $p - \sqrt{n} < \sqrt[4]{4n}$, alors on retrouve $u < 2$. Ainsi si $p - \sqrt{n} < \sqrt[4]{4n}$ alors le nombre d'itérations à faire est juste 1.