Université Paris 8 — Année 2022–23 — J. Lavauzelle

Master 1 Mathématiques et applications

Cryptographie à clé publique – Solutions feuille de TD 2

03/02/2023

Retrouvez le sujet du TD et d'autres exercices à l'adresse :

`www.math.univ-paris13.fr/~lavauzelle/teaching/2022-23/clef-publique.html`

($\star$) exercice fondamental ($\star\star$) pour s'entraîner ($\star\star\star$) pour aller plus loin sur machine

Exercice 1. ($\star$) Application de RSA brut.

Dans cet exercice, on s'intéresse à une version « jouet » (c'est-à-dire, avec de petites valeurs) du chiffrement RSA brut.

Les nombres premiers $p = 17$ et $q = 23$ ont été engendrés par Alice, et l'entier $n = pq = 9191$ a été publié.

Question 1.– Alice peut-elle utiliser $e = 50$ comme seconde partie de sa clé publique ?

Question 2.– On suppose maintenant que $e = 3$. Calculer la valeur de $\phi(n)$, puis de l'exposant privé d.

Question 3.– Chiffrer le message $m = 10$ avec la clef publique (n, e).

Question 4.– Calculer $d_p := d \mod (p-1)$ et $d_q := d \mod (q-1)$.

Question 5.– Calculer deux entiers u et v tels que $up + vq = 1$.

Question 6.– Étant donné le chiffré $c = 2$, calculer $c^{d_p} \mod p$ et $c^{d_q} \mod q$. Puis en déduire la valeur du message associé au chiffré c.

Solutions de l'Exercice 1.

Solution Q1. Non, car e est pair et $\phi(n)$ également (car p et q sont impairs), donc e et $\phi(n)$ ne sont pas premiers entre eux.

Solution Q2. On a $\phi(n) = 16 \times 22 = 352$. Puis, pour calculer d, il faut calculer l'inverse de 3 modulo 352. Pour cela, on exécute l'algorithme d'Euclide étendu. Ici, il peut se résumer à :

$$352 = 3 \times 117 + 1$$

puisqu'on en déduit que

$$3 \times (-117) + 352 \times 1 = 1$$

puis que $d = 3^{-1} \mod 352$ vaut $352 - 117 = 235$.

Solution Q3. On calcule $m^e = 10^3 \mod n$, autrement dit le reste de la division euclidienne de 1000 par 352. On obtient 296.

Solution Q4. On a :

$$d_p = 11 \quad \text{et} \quad d_q = 15$$

Solution Q5. Grâce à l'algorithme d'Euclide étendu, on obtient :

$$\begin{aligned} 23 &= 17 + 6 \\ 17 &= 2 \times 6 + 5 \\ 6 &= 5 + 1 \end{aligned}$$

Puis :

$$\begin{aligned} 1 = 6 - 5 = 3 \times 6 - 17 = 3 \times 23 + (-4) \times 1723 &= 17 + 6 \\ 17 &= 2 \times 6 + 5 \\ 6 &= 5 + 1 \end{aligned}$$

Autrement dit, $u = 3$ et $v = -4$.

Solution Q6. Voici un exemple de calcul « à la main ». On a :

$$c^{d_p} = 2^{11} = 2^3 \times (2^4)^2 = 8 \times 16^2 \equiv 8 \times (-1)^2 \equiv 8 \mod 17$$

et comme $2^6 = 64 \equiv -5 \mod 23$, on obtient

$$c^{d_q} = 2^{15} = (2^6)^2 \times 2^3 \equiv (-5)^2 \times 8 = 25 \times 8 \equiv 2 \times 8 = 16 \mod 23$$

Puis, on reconstitue c par un petit calcul :

$$c = (c^{d_p} \mod p)vq + (c^{d_q} \mod q)up = 8 \times 3 \times 23 + 16 \times (-4) \times 17 = -536 \equiv 246 \mod 391$$

Exercice 2. ($\star$) Factorisation de n grâce à $\phi(n)$.

Soit $n = pq$ où p et q sont deux nombres premiers distincts.

Question 1.– Rappeler comment calculer l'indicatrice d'Euler $\phi(n)$ à partir de p et q, les entiers qui composent la factorisation de n.

Question 2.– Supposons maintenant que l'on connaisse n et $\phi(n)$. Donner une méthode pour factoriser n. On précisera un ordre de grandeur pour sa complexité.

Solutions de l'Exercice 2.

Solution Q1. Lorsque n est le produit de deux nombres premiers distincts p et q, on a $\phi(n) = (p-1)(q-1)$.

Solution Q2. Développons l'égalité $\phi(n) = (p-1)(q-1)$. On a :

$$\phi(n) = pq - p - q + 1 = n + 1 - (p+q)$$

Une idée est alors que chercher p et q comme les solutions d'une équation du second degré (c'est un type d'équation que l'on sait résoudre). Ici, on a alors :

$$(X-p)(X-q) = X^2 - (p+q)X + pq = X^2 - (n+1-\phi(n))X + n$$

Les entiers p et q sont donc les deux solutions de $X^2 - (n+1-\phi(n))X + n = 0$ dans $\mathbb{Q}$ (par exemple).

Dans notre cas, la résolution de cette équation quadratrique nécessite quelques multplications, additions et l'extraction d'une racine carrée que l'on sait être un entier. Cela a un coût polynomial en la tailles des entrées ($\log(n)$).

Exercice 3. ($\star$) Attaque sur RSA à module identique.

Deux amis qui se font mutuellement confiance utilisent le même module RSA $n = pq$, mais avec des exposants (e_1, d_1) et (e_2, d_2) différents.

On se place dans un scénario où une troisième personne souhaite envoyer les chiffrés d'un même message m aux deux amis. On suppose qu'il utilise le mode d'utilisation « brut » du chiffrement RSA.

Question 1.– On suppose que les exposants e_1 et e_2 choisis par les deux amis sont premiers entre eux. Expliquer pourquoi, dans ce cas, un attaquant passif peut retrouver le message m.

Solutions de l'Exercice 3.

Solution Q1. On note $c_1 = m^{e_1} \mod n$ et $c_2 = m^{e_2} \mod n$ les deux chiffrés envoyés aux deux amis. L'attaquant cherche la valeur $m = m^1 \mod n$. Pour cela, une idée pour être de calculer un produit de la forme :

$$c_1^u c_2^v \mod n$$

qui soit égal à m.

Rappelons maintenant que les exposants e_1 et e_2 sont publics. On peut donc calculer les coefficients de Bezout associés à e_1 et e_2. C'est-à-dire, on calcule grâce à l'algorithme d'Euclide étendu deux entiers u et v tels que

$$ue_1 + ve_2 = \text{pgcd}(e_1,e_2) = 1\,.$$

Alors, dans ce cas on obtient m en calculant simplement

$$c_1^u c_2^v = m^{ue_1+ve_2} = m\,.$$

Cela nécessite deux exponentiations et une multiplication modulaires.

Exercice 4. ($\star\star$) Attaque de Håstad avec $e = 3$.

Trois utilisateurs ont engendré des clés RSA de modules n_1, n_2 et n_3. On fait l'hypothèse que ces modules sont deux-à-deux premiers entre eux, mais observons que c'est extrêmement probable si leur génération est aléatoire (commes les n_i sont produit de deux grand nombres premiers).

Les trois utilisateurs choisissent le même exposant de chiffrement $e = 3$, et on suppose qu'un même message m est envoyé aux trois utilisateurs.

Question 1.– Comment peut-on calculer $m^e \mod n_1n_2n_3$ à partir des chiffrés de m par les 3 clés publiques ?

Question 2.– En déduire une attaque passive permettant de retrouver le message m.

Question 3.– Cette attaque se généralise-t-elle, en pratique, pour n'importe quel exposant $e \geq 3$? Si oui, avec quelle contrainte ?

Solutions de l'Exercice 4.

Solution Q1. Notons $c_i := m^3 \mod n_i$ le chiffré de m par le i-ème utilisateur. La valeur de c_i est connue de l'attaquant.

Si a et b sont premiers entre eux, le théorème des restes chinois donne un isomorphisme d'anneaux :

$$\begin{array}{ccc} \mathbb{Z}/a\mathbb{Z} \times \mathbb{Z}/b\mathbb{Z} & \to & \mathbb{Z}/ab\mathbb{Z} \\ (x = c \mod a, y = c \mod b) & \mapsto & z = c \mod ab\,. \end{array}$$

On rappelle que cet isomophisme est explicite et calculable efficacement. Il suffit de retrouver les coefficients de Bezout u et v tels que $au + bv = 1$, puis de calculer

$$xbv + yau \mod ab\,.$$

En utilisant cet isomorphisme avec les $c_i = m^3 \mod n_i$ pour les paramètres $(a, b) = (n_1, n_2)$ puis $(a, b) = (n_1 n_2, n_3)$, on obtient alors la valeur de $m^3 \mod n_1 n_2 n_2$.

Solution Q2. Le message m a été choisi de sorte que $m < n_i$ pour tout i. On note donc que $m^3 < n_1 n_2 n_3$, donc le reste $m^3 \mod n_1 n_2 n_3$ est toujours égal à l'entier m^3.

Il suffit donc d'extraire une racine cubique de la valeur obtenue après application de l'isomorphisme des restes chinois. L'extraction d'une racine cubique est une opération très efficace dans les entiers naturels (cela requiert moins de $\log(m^3)$ opérations élémentaires sur les entiers).

Solution Q3. En théorie, l'attaque peut être généralisée à $e > 3$, mais il faut qu'il y ait e participants. Il y a alors deux contraintes importantes :

— il faut que e participants envoient le même message, ce qui est peu probable si e est grand,
— la complexité de l'attaque est polynomiale en e, donc une nouvelle fois, il ne faut pas que e soit trop grand.

Exercice 5. 💻 (⋆⋆) Implantation de l'algorithme de factorisation de Fermat.

Question 1.– Implanter l'algorithme de Fermat pour factoriser un entier n de la forme $n = pq$ avec p et q deux nombres premiers distincts assez proches.

Question 2.– Tester votre algorithme en factorisant les nombres suivants :

$n = 25199$
$n = 156671083$
$n = 11111148395556329947$
$n = 2313990783732947538207933535547834103442505060527789035227083$
$n = 16533082011949263925258910251486533787569244299818428169066201791667939411$
$09301928793123047363367149725481739142913896195578837035688735806993232952$
$17135965632281658378042394131207818120190147545168152283832265103113463740$
$04398334270694865159367733936978134600897908705674806643139952985945646368$
$11947176565058919290144182763100104940038556737097132199675547974821171076$
$02370708075051839060591806419939722030332762246457642592180431405229608091$
$96557899930545084338329506150798920037641690302506887447751207884557752337$
$50866404439635435041766331267212617692243110705703766732763604380629660308$
98045246243

Enfin, l'entier n (de taille $\simeq 10\,000$ bits) que vous pouvez trouver à la page suivante :

`www.math.univ-paris13.fr/~lavauzelle/teaching/2022-23/docs/CP/td/aux/factorisation-fermat.txt`

Pour ce dernier entier, la factorisation pourrait éventuellement prendre quelques dizaines de secondes, selon votre implantation et votre machine.

Solutions de l'Exercice 5.

Voir script en ligne :

`https://www.math.univ-paris13.fr/~lavauzelle/teaching/2022-23/docs/CP/td/scripts/fermat.py`