Université Paris 8 Année 2021–2022

M1 Mathématiques et applications, parcours ACC et CSSD

Cryptographie à clef publique – Feuille de TD 2

04/02/2022

Le corrigé de certains exercices sera disponible à l'adresse suivante :

`www.math.univ-paris13.fr/~lavauzelle/teaching/2021-22/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.

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.

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.

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 $m^e \mod n_1 n_2 n_3$ 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 ?

Exercice 5. $(\star)$ RSA brut avec un petit message.

On considère le chiffrement RSA dans son mode de fonctionnement « brut ». On note $n = pq$ le module et e l'exposant public.

Question 1.– Soit c le chiffré d'un message m tel que $m < n^{1/e}$. Expliquer pourquoi, si un attaquant sait que le message envoyé m est plus petit que $n^{1/e}$, alors il peut retrouver m à partir de c.

Exercice 6. 💻 $(\star\star)$ 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 :

$n = 25199$

$n = 156671083$

$n = 11111148395556329947$

$n = 2313990783732947538207933535547834103442505060527789035227083$

$n = 165330820119492639252589102514865337875692442998184281690662017916679394110930192879312304736336714972548173914291389619557883703568873580699323295217135965632281658378042394131207818120190147545168152283832265103113463740043983342706948651593677339369781346008979087056748066431399529859456463681194717656505891929014418276310010494003855673709713219967554797482117107602370708075051839060591806419939722030332762246457642592180431405229608091965578999305450843383295061507989200376416903025068874477512078845577523375086640443963543504176633126721261769224311070570376673276360438062966030898045246243$

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/2021-22/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.