TD 3 chiffrement asymétrique

Remarque

Les questions marquées d'une étoile ($\star$) sont optionnelles.
Pour l'intégralité de ce travail dirigé, on suppose que l'on se place dans un système RSA, paramétré de la manière suivante :

- p et q deux premiers distincts ;
- $N = p \cdot q$ définissant l'anneau $\mathbb{Z}_N$;
- $\phi(N)$ l'indicatrice d'Euler évaluée en N ;
- e et d deux entiers tels que $e \cdot d \equiv 1[\phi(N)]$.

Pour rappel, *Textbook* RSA est définit comme suit :

- $KeyGen(1^\lambda)$: pour un niveau de sécurité λ, renvoie $P_k = (e, N)$ et $S_k = d$;
- $Encrypt(P_k, m) = m^e[N] = C$;
- $Decrypt(S_k, C) = C^d[N] = m$.

Quelques propriétés de l'indicatrice d'Euler

L'indicatrice d'Euler ϕ est une fonction qui évaluée en n, donne le nombre d'entiers compris entre 1 et $n-1$ premiers avec n (c'est-à-dire $PGCD(x, n) = 1, x \in \{1, \cdots, n-1\}$). On se propose d'étudier quelques propriétés de l'indicatrice d'Euler que l'on utilise pour le système RSA.

Question 1

Montrez que si n est un nombre premier, alors $\phi(n) = n - 1$.

Question 2

En considérant que $n = p \cdot q$, avec p et q premiers, montrer que $\phi(n) = (p-1) \cdot (q-1)$. (on supposera que $p > q$)

Question 3

On se place dans le système RSA paramétré dans l'introduction. On pose le polynôme $P[X] = X^2 + (\phi(N) - N - 1) \cdot X + N$. Calculez les racines du polynôme P.

Question 4

Conclure sur la complexité de la factorisation de N si l'on arrive à trouver un algorithme capable de calculer l'indicatrice d'Euler en temps polynomial.

Cryptanalyse d'un algorithme de remplissage pour RSA

Vous êtes expert sécurité dans votre entreprise et on vous demande de mettre en place un système RSA sécurisé. Vous essayez de vous souvenir de votre cours de cryptographie et vous vous rappelez vaguement que *Textbook* RSA n'est pas IND-CPA. Les mots-clés randomisation et algorithme de remplissage vous viennent à l'esprit. Vous décidez donc de proposer votre propre mécanisme de randomisation et de remplissage pour votre implémentation. Le remplissage que vous proposez est simple : Les 256 premiers bits sont constitués de bits aléatoires provenant d'une source d'aléa pure et uniforme, et les bits suivants sont consacrés à votre message. On considérera que le module N est sur 4096 bits et que notre message a au plus 3840 bits.

Question 5

Montrez que *Textbook* RSA n'est pas IND-CPA.

Question 6

Montrez que le schéma de remplissage proposé est compatible avec une utilisation pour RSA.

Question 7

Montrez que cette construction atteint un niveau de sécurité IND-CPA.

Question 8

Montrez que cette construction n'atteint pas un niveau de sécurité IND-CCA2. Pensez-vous que cette construction atteint un niveau de sécurité IND-CCA1 ?

Instances faibles de RSA - cas des messages bornés

On se propose d'étudier le cas d'un système RSA où le message est borné sur un nombre de bits donnés L. En d'autres termes, le message m vérifie la contrainte suivante :

$$m < 2^L$$

Pour étudier la sécurité de cette instance, nous allons raisonner à partir du constat qu'il y a une probabilité non négligeable que notre message m puisse s'écrire comme le produit de deux entiers r et s, avec r et s très inférieur à m. Plus formellement, on suppose qu'il existe un paramètre $\alpha \in]1/2, 1[$ tel que r et s soient plus petits que $2^{\alpha L} = T$.

Question 9

A partir de la définition du système RSA, donnez la forme probable du chiffré en fonction de r, s, e et N.

Question 10

Par quelle valeur peut-on multiplier un chiffré quelconque c pour supprimer la dépendance en s du résultat ?

Question 11

On suppose qu'un attaquant a calculé tous les couples $(r_i, r_i^e[N] = x_i)_{1 \leq r_i < T}$. Il peut le faire car e et N sont des paramètres publiques. A partir de la question précédente, en déduire un algorithme itératif qui permet de calculer m. (Indice : On cherche une collision avec la table)

Question 12

Quel est le nombre maximal d'itération de l'algorithme précédent ?

Question 13

Application numérique : Pour $\alpha = 1/2$, on suppose que la probabilité que $m = r \cdot s$, avec $s < 2^{L/2}$ et $r < 2^{L/2}$ est égale à 0.2. A partir de combien de messages chiffrés l'attaquant a une probabilité supérieure à 50% de pouvoir déchiffrer correctement un message ? 90% ?

Question 14

Cette attaque fonctionne-t-elle sur RSA-OAEP ?

Instances faibles de RSA - cas de deux nombres premiers proches

On se propose d'étudier une instance faible de RSA où la fonction KeyGen aurait choisi deux premiers p et q anormalement proches (nous verrons par la suite ce que l'on entend par anormalement proches). On note $N = p \cdot q$, $t = \frac{p+q}{2}$ et $s = \frac{p-q}{2}$. On fixe également $p > q$ par convention.

Question 15

Montrez que $N = t^2 - s^2$.

Question 16

Montrez que t est proche de la racine carré de N, et légèrement supérieur.

La méthode de factorisation de Fermat est un algorithme permettant la décomposition d'un entier en deux facteurs à partir du calcul de sa racine carrée :

Data: N
Result: p, q such as $N = p \cdot q$
$t \leftarrow \lceil \sqrt{N} \rceil$;
$z = 2$;
while z *not a square* **do**
  $t \leftarrow t + 1$;
  $z \leftarrow t^2 - N$;
end
$p \leftarrow t + \sqrt{z}$;

Algorithm 1: Algorithme de factorisation de Fermat

Question 17

Montrez que plus p et q sont proches, plus l'algorithme de factorisation de Fermat converge rapidement.

Question 18

A l'aide du langage de programmation de votre choix, décrivez l'algorithme de factorisation de Fermat et testez-le pour les entiers N suivants :

- 1517.
- 17855222095927.
- 30491161587179.

Question 19

Étudiez les valeurs que peuvent prendre les 4 premiers bits d'un nombre lorsque ce nombre est un carré. En déduire une légère modification de l'algorithme de Fermat permettant d'accélérer les calculs.

Instances faibles de RSA - cas du chiffrement d'un message par deux systèmes RSA partageant le même N

Alice, Bob et Corinne souhaitent communiquer entre eux en utilisant le système RSA. Trouvant la génération de grands premiers complexe à implémenter, ils décident d'un commun accord de ne générer qu'un couple (p, q) pour tout le monde, et de personnaliser les paramètres (e, d). Cette étape est relativement simple, il suffit de piocher de manière aléatoire un élément e dans $\mathbb{Z}_{\phi(N)}$, et tant que e est premier avec $\phi(N)$, on sait que d existe et on peut le calculer. Ils décident de faire bien les choses et se mettent d'accord que que les différents e_i générés soient premiers entre eux.
On note donc (e_A, d_A) les paires de clés (Publique,Privée) de Alice, (e_B, e_B) les paires de clés de Bob, et (e_C, d_C) les paires de clés de Corinne. Dans notre exemple, on suppose que Alice a trouvé un site internet exceptionnel et compte le partager avec Bob, elle chiffre donc un message m en utilisant la clé publique e_B de Bob et lui envoie $C_1 = m^{e_B}[N]$. Bob à son tour, trouve génial le site d'Alice et chiffre donc ce même message m pour Corinne en utilisant sa clé publique e_C et lui envoie $C_2 = m^{e_C}[N]$.

Question 20

Montrez qu'un attaquant ayant récupéré les paramètres publiques de Alice, Bob et Corinne et ayant intercepté les deux chiffrés peux récupérer le message. On pourra raisonner à partir de l'identité de Bézout :

$$\forall (x, y) \in \mathbb{Z}, \exists (u, v) \in \mathbb{Z}, x \cdot u + y \cdot v = PGCD(x, y)$$