Série 12 : Solutions

1 Chiffrer et déchiffrer

a) Le fait d'utiliser un alphabet de 28 lettres plutôt qu'un alphabet de 26 lettres augmente le nombre de chiffrements possibles de 26! à 28!, donc d'un facteur multiplicatif $28!/26! = 28 \times 27 = 756$. Le temps de déchiffrement sera donc de **756 heures**.

Note : Il y a des moyens plus efficaces que l'attaque par force brute pour décrypter un tel chiffrement (notamment en analysant les fréquences d'apparition des lettres dans le message chiffré et en essayant de faire correspondre celles-ci aux fréquences communes des lettres en français).

b) Doubler la taille de la clé K de 20 à 40 bits multiplie le nombre de possibilités par un facteur $2^{20} \simeq 10^6$. Avec un ordinateur 100 fois plus puissant que le premier, le temps de déchiffrement sera donc environ 10 000 fois plus long, soit :

$$5 \text{ min} \times 10\,000 = 50\,000 \text{ min} \simeq 833 \text{ heures}.$$

2 One-Time Pad

a) On calcule $C = M \oplus K$ bit par bit :

M	1	1	0	1	0	0	1	1
K	1	0	1	1	1	0	0	1
$C = M \oplus K$	0	1	1	0	1	0	1	0

Alice envoie donc $C = 0\,1\,1\,0\,1\,0\,1\,0$. Pour vérifier, Bob calcule :

C	0	1	1	0	1	0	1	0
K	1	0	1	1	1	0	0	1
$C \oplus K$	1	1	0	1	0	0	1	1

On retrouve bien M. Ceci découle de $C \oplus K = (M \oplus K) \oplus K = M \oplus (K \oplus K) = M \oplus 0 = M$.

b) On convertit chaque lettre selon A= 0, B= 1, . . ., Z= 25, on additionne $C = (M + K) \bmod 26$, puis on reconvertit en lettre :

M	C (2)	R (17)	Y (24)	P (15)	T (19)	O (14)
K	S (18)	E (4)	C (2)	R (17)	E (4)	T (19)
$M + K \bmod 26$	20	21	0	6	23	7
C	U	V	A	G	X	H

Alice envoie donc $C =$ `UVAGXH`. Pour déchiffrer, Bob calcule $D = (C - K) \bmod 26$ lettre par lettre, ce qui lui redonne bien $M =$ `CRYPTO`.

c) Si Alice utilise la même clé K pour chiffrer deux messages différents, alors :

$$C_1 \oplus C_2 = (M_1 \oplus K) \oplus (M_2 \oplus K) = M_1 \oplus M_2.$$

Autrement dit, $C_1 \oplus C_2$ ne dépend plus du tout de la clé K. Eve obtient donc directement le XOR des deux messages clairs. À partir de là, en exploitant la redondance de la langue (lettres fréquentes, espaces, mots probables, etc.), elle peut souvent reconstituer M_1 et M_2. La sécurité du *one-time pad* repose donc crucialement sur le fait que la clé ne doit *jamais* être réutilisée (d'où son nom de clé à usage unique).

3 Protocole de Diffie-Hellman

a) Avec $P = 23$, $Q = 5$, $N_A = 6$ et $N_B = 15$:

• Alice calcule $N_{QA} = Q^{N_A} \bmod P = 5^6 \bmod 23$. On a $5^2 = 25 \equiv 2 \pmod{23}$, donc $5^4 \equiv 4$ et $5^6 \equiv 5^4 \cdot 5^2 \equiv 4 \cdot 2 = 8 \pmod{23}$. Donc $\mathbf{N_{QA} = 8}$.

Bob calcule $N_{QB} = Q^{N_B} \bmod P = 5^{15} \bmod 23$. On a $5^8 \equiv 4^2 = 16$, et :

$$5^{15} = 5^8 \cdot 5^4 \cdot 5^2 \cdot 5 \equiv 16 \cdot 4 \cdot 2 \cdot 5 = 640 \equiv 19 \pmod{23}$$

(car $640 = 27 \cdot 23 + 19$). Donc $\mathbf{N_{QB} = 19}$. Alice envoie 8 à Bob et Bob envoie 19 à Alice.

• Alice calcule $K = N_{QB}^{N_A} \bmod P = 19^6 \bmod 23$. On a $19 \equiv -4 \pmod{23}$, donc $19^6 \equiv 4^6 \pmod{23}$. Or $4^2 = 16$ et $4^4 = 256 \equiv 3 \pmod{23}$ (car $256 = 11 \cdot 23 + 3$), d'où $4^6 = 4^4 \cdot 4^2 \equiv 3 \cdot 16 = 48 \equiv 2 \pmod{23}$. Donc $\mathbf{K = 2}$.

Bob calcule $K = N_{QA}^{N_B} \bmod P = 8^{15} \bmod 23$. En écrivant $8 = 2^3$, on a $8^{15} = 2^{45}$. Par le petit théorème de Fermat, $2^{22} \equiv 1 \pmod{23}$, donc $2^{45} = (2^{22})^2 \cdot 2 \equiv 2 \pmod{23}$. On retrouve bien $\mathbf{K = 2}$.

• Pour retrouver K, Eve doit, par exemple, retrouver N_A à partir de P, Q et N_{QA}, c'est-à-dire résoudre $5^{N_A} \equiv 8 \pmod{23}$ pour N_A. C'est ce qu'on appelle le *problème du logarithme discret.* Avec un grand nombre premier P (typiquement 2048 bits ou plus), il n'y a pas d'algorithme connu pour le résoudre en un temps raisonnable : c'est l'opération à sens unique sur laquelle repose la sécurité du protocole.

b) Le protocole pour 3 personnes (Alice, Bob et Charlie) est le suivant :

• Les 3 personnes se mettent d'accord publiquement sur un grand nombre premier P et une base Q entre 1 et $P-1$. Puis Alice, Bob et Charlie choisissent respectivement N_1, N_2, N_3, des nombres compris entre 1 et $P-1$. Ces 3 nombres restent privés.

• Premier tour : Alice, Bob, Charlie calculent respectivement $N_4 = Q^{N_1} \bmod P$, $N_5 = Q^{N_2} \bmod P$, $N_6 = Q^{N_3} \bmod P$ et publient ces nombres.

• Deuxième tour :
- Alice calcule $N_7 = N_5^{N_1} \bmod P = Q^{N_2 N_1} \bmod P$
- Bob calcule $N_8 = N_6^{N_2} \bmod P = Q^{N_3 N_2} \bmod P$
- Charlie calcule $N_9 = N_4^{N_3} \bmod P = Q^{N_1 N_3} \bmod P$

Ces trois nombres N_7, N_8 et N_9 sont également publiés.

• Dernier tour, pour arriver à la clé commune K :
- Alice calcule $N_8^{N_1} \bmod P = Q^{N_1 N_2 N_3} \bmod P = K$
- Bob calcule $N_9^{N_2} \bmod P = Q^{N_1 N_2 N_3} \bmod P = K$
- Charlie calcule $N_7^{N_3} \bmod P = Q^{N_1 N_2 N_3} \bmod P = K$

On peut vérifier qu'Eve, en possession des nombres publiés $P, Q, N_4, N_5, N_6, N_7, N_8, N_9$ mais pas de N_1, N_2 ou N_3, ne peut retrouver K à moins d'être capable de résoudre le problème du logarithme discret.

4 Chiffrement RSA

a) Avec $p = 3$ et $q = 11$:

$$n = p \cdot q = 33, \qquad \varphi(n) = (p-1)(q-1) = 2 \cdot 10 = 20.$$

b) On a $1 < 3 < 20$ et $\gcd(3, 20) = 1$, donc $e = 3$ est un choix valide. On cherche d tel que $3d \equiv 1 \pmod{20}$. En essayant $d = 7 : 3 \cdot 7 = 21 \equiv 1 \pmod{20}$. Donc $\mathbf{d = 7}$.

• Clé publique d'Alice : $(e, n) = (\mathbf{3}, \mathbf{33})$.

• Clé privée d'Alice : $(d, n) = (\mathbf{7}, \mathbf{33})$.

c) Bob chiffre $m = 4$ avec la clé publique d'Alice :

$$c = m^e \bmod n = 4^3 \bmod 33 = 64 \bmod 33 = \mathbf{31}.$$

d) Alice déchiffre en appliquant sa clé privée :

$$m = c^d \bmod n = 31^7 \bmod 33.$$

Comme $31 \equiv -2 \pmod{33}$, on a $31^7 \equiv (-2)^7 = -128 \pmod{33}$. Or $-128 + 4 \cdot 33 = -128 + 132 = 4$, donc $31^7 \equiv 4 \pmod{33}$. Alice retrouve bien $\mathbf{m = 4}$.

5 Questions d'examens passés

a)

Concernant le protocole d'échange de clés de Diffie-Hellman, quelles affirmations sont **vraies** ?

✓ La sécurité du protocole repose sur la difficulté calculatoire de résoudre le problème du logarithme discret.

- ☐ Pour une sécurité maximale, les clés privées secrètes N_A et N_B doivent être plus grandes que le module premier P.
- ✓ Le protocole est vulnérable à une attaque de l'homme du milieu si un attaquant peut intercepter et modifier les clés publiques échangées.
- ☐ La sécurité du protocole exige que la base Q soit gardée secrète entre les deux parties.

Solution

1. Vrai. C'est l'opération à sens unique sur laquelle repose le protocole : $a^b \bmod P$ est facile à calculer, mais retrouver b à partir de P, a et $a^b \bmod P$ est très difficile.
2. Faux. Les exposants N_A et N_B sont choisis entre 1 et $P-1$. Choisir $N_A > P$ ne change rien car par le petit théorème de Fermat $Q^{N_A} \bmod P = Q^{N_A \bmod (P-1)} \bmod P$.
3. Vrai. Si un attaquant peut intercepter et remplacer N_{QA} et N_{QB} par ses propres valeurs, il établit deux clés indépendantes (une avec Alice, une avec Bob) et peut relayer/lire les messages : c'est l'attaque dite *man-in-the-middle*. Pour s'en prémunir, on doit authentifier les clés publiques (par exemple via une signature).
4. Faux. P et Q sont des paramètres publics du protocole. Seuls N_A et N_B doivent rester secrets.

b)

La sécurité du protocole d'échange de clé Diffie-Hellman repose sur la difficulté de factoriser de grands nombres entiers.

- ☐ VRAI
- ✓ FAUX

Solution

FAUX. La sécurité de Diffie-Hellman repose sur la difficulté du *problème du logarithme discret* (retrouver b à partir de $a^b \bmod P$), et non sur la factorisation. C'est en revanche la factorisation de grands entiers qui sert de fondement à la sécurité de RSA.

c)

Parmi les propositions suivantes concernant le hachage (stockage de mots de passe), les signatures numériques et la cryptographie asymétrique, cochez *toutes* celles qui sont correctes.

- ✓ Ajouter un sel aléatoire propre à chaque utilisateur (`hash(salt + mot_de_passe)`) empêche qu'on puisse déduire qu'un même mot de passe est utilisé par plusieurs comptes.
- ✓ Une signature numérique se crée avec la clé privée du signataire et se vérifie à l'aide de sa clé publique.
- ☐ Pour assurer la confidentialité d'un message, on le chiffre avec la clé *privée* du destinataire.
- ☐ Le sel utilisé pour le hachage des mots de passe doit rester secret et ne jamais être stocké en clair.

Solution

1. Vrai. Sans sel, deux utilisateurs ayant le même mot de passe auraient le même hash ; le sel (aléatoire et propre à chaque utilisateur) garantit que les hash diffèrent même pour des mots de passe identiques.
2. Vrai. C'est la définition même de la signature numérique : signer avec la clé privée (que seul le signataire possède), vérifier avec la clé publique (que tout le monde peut consulter).
3. Faux. Pour la confidentialité, on chiffre avec la clé *publique* du destinataire ; lui seul, grâce à sa clé privée, pourra déchiffrer.
4. Faux. Le sel est stocké *en clair* à côté du hash (on en a besoin pour vérifier le mot de passe à la connexion). Sa sécurité ne vient pas du secret, mais du fait qu'il rend les attaques par dictionnaire/rainbow tables inutiles.

d)

En utilisant RSA, vous souhaitez protéger une information que vous envoyez à un ami dont la clé publique est $(17,\ 77)$ et la clé privée est $d = 41$. Votre clé publique est $(11,\ 65)$ et votre clé privée est $d = 23$.

A. En utilisant les lettres m et c pour le message en clair et chiffré respectivement, expliquez précisément chaque étape de la communication.

B. Vous envoyez $m = 19$ à votre ami. Il déchiffre et obtient 12 au lieu de 19. Donnez une possible erreur.

Solution

A. Les étapes de la communication sont :

1. Vous récupérez la clé publique de votre ami : $(e, n) = (17, 77)$.
2. Vous chiffrez le message m avec cette clé publique :

$$c = m^e \bmod n = m^{17} \bmod 77.$$

3. Vous envoyez c sur le canal public.
4. Votre ami reçoit c et le déchiffre avec sa clé privée $(d, n) = (41, 77)$:

$$m = c^d \bmod n = c^{41} \bmod 77.$$

La correction du procédé est garantie par le choix de d tel que $e \cdot d \equiv 1 \pmod{\varphi(n)}$, ce qui donne $m^{e \cdot d} \equiv m \pmod{n}$.

B. Avec un chiffrement correct : $c = 19^{17} \bmod 77 = 34$. Avec un déchiffrement correct : $34^{41} \bmod 77 = 19$ (valeur attendue).
L'ami obtient 12. En cherchant dans la table de valeurs, on voit que $34^{17} \bmod 77 = 12$. Autrement dit, l'ami a calculé $c^{17} \bmod 77$ au lieu de $c^{41} \bmod 77$: **il a utilisé sa clé publique (exposant $e = 17$) au lieu de sa clé privée (exposant $d = 41$) pour déchiffrer le message**.
Remarque : d'autres erreurs sont parfois invoquées (chiffrement avec la mauvaise clé, etc.) mais ne donnent pas le résultat 12 ici. La seule explication cohérente avec la table fournie est bien celle ci-dessus.