# TD Cryptographie

ENSSAT

09/03/2021

---

## Exercice 6 – Chiffrement affine

Alice souhaite communiquer avec Bob en utilisant la méthode suivante : chacun des $n$ caractères possibles (par exemple $n = 26$ ou $n = 256$) est codé en un élément de $\mathbb{Z}_n = \{0, 1, \ldots, n-1\}$ ; Alice et Bob conviennent d'une clef commune $k = (a, b)$ où $a$ et $b$ sont dans $\mathbb{Z}_n$, et Alice chiffre chaque caractère codé $x$ à l'aide de la règle

$$e_k(x) = (a\,x + b) \bmod n$$

1. Donner la règle de déchiffrement $d_k$ que doit utiliser Bob. Alice et Bob peuvent-ils choisir n'importe quelle clef dans $\mathbb{Z}_n \times \mathbb{Z}_n$ ?
2. Illustrer la méthode avec un exemple numérique de votre choix lorsque $n = 26$.
3. Combien existe-t-il de clefs possibles ? Conclusion ?

---

**Corrigé**

1. Notons $c$ l'inverse de $a$ modulo $n$. Alors :

$$\begin{aligned} a\,x + b = y \mod n &\iff a\,x = y - b \mod n \\ &\iff x = c(y-b) \mod n \end{aligned}$$

   Ainsi, la règle de déchiffrement $d_k$ vérifie :

$$d_k(y) = c(y-b) \mod n$$

   On ne peut cependant pas choisir n'importe quelle clé. En effet, si $a$ n'est pas inversible modulo $n$, alors on ne peut pas inverser. Il est donc nécessaire, et suffisant, d'avoir $a$ inversible modulo $n$, c'est-à-dire $a$ premier avec $n$.
2. Prenons $a = 5$ et $b = 2$. PGCD$(5, 26) = 1$ et le couple $(a, b)$ est une bonne clé. Alors la clé de décryptage est

$$d_k(y) = 21(y-2) \mod 26 = 21\,y + 10 \mod 26$$

3. Par définition, il y a $\varphi(n)$ éléments premiers de $\mathbb{Z}_n$ premier avec $n$, et il y a donc $n \times \varphi(n)$ clés possibles : $n$ choix pour $b$, $\varphi(n)$ pour $a$. Dans tous les cas, il y a en moins de $n^2$ et est donc facilement déchiffrable en testant toutes les clés possibles.

---

## Exercice 7 – Simulation d'un chiffrement RSA

Alice et Bob veulent communiquer en utilisant RSA. Nous allons traiter un exemple numérique en utilisant des petits nombres afin que les calculs soient rapidement faisables avec une calculatrice de poche (il s'agit ici simplement d'illustrer le fonctionnement de ce système, mais en pratique les entiers $p$ et $q$ doivent bien sûr être très grands afin d'éviter la cryptanalyse du système).

Bob publie l'exposant de chiffrement $b = 17$ et le modulus $n = 35$.

1. Alice veut envoyer le message clair $x = 12$ à Bob (il s'agit par exemple du codage d'un caractère clair). Quel est le message chiffré $y$ qu'elle envoie à Bob (utiliser l'algorithme d'exponentiation modulaire rapide).

2. En pratique Bob a gardé secrets deux entiers premiers $p$ et $q$, mais ici le calcul de ces valeurs est immédiat, de sorte qu'Oscar, qui intercepte $y$, peut en profiter pour retrouver $x$. Faites le travail d'Oscar.

---

**Corrigé**

1. Alice doit envoyer $x^b \mod n$, c'est-à-dire $12^{17} \mod 35$, que l'on calcule par la méthode que l'on souhaite, par exemple l'exponentiation rapide :

$$12^2 \equiv 4 \mod 35 \text{ donc } 12^4 \equiv 4^2 = 16 \mod 35 \text{ et } 12^8 \equiv 16^2 \equiv 11 \mod 35$$

mais alors

$$12^{17} = 12^8 12^8 12 \equiv 11 \times 11 \times 12 \equiv 17 \mod 35$$

Ainsi, Alice envoie le message crypté 17.

2. Bob a choisi comme clé publique $b = 17$ et $n = 35$. Remarquons que

$$\varphi(35) = \varphi(5 \times 7) = \varphi(7)\varphi(5) = 6 \times 4 = 24$$

donc 17 est bien premier avec $\varphi(n)$. On doit chercher l'inverse de 17 modulo $\varphi(n) = 24$. Pour cela, on détermine par l'algorithme d'Euclide étendu le PGCD de 17 et 24 (qui vaut 1). En remontant, on obtiendra

$$5 \times 24 - 7 \times 17 = 1$$

et donc, modulo 24, $-7$ est l'inverse, c'est-à-dire aussi $-7 + 24 = 17$ (on prend l'entier positif entre 1 et 24. Donc $e = 17$ est l'inverse de 17.
Il suffit alors à Oscar de calculer message$^e \mod n$ pour déterminer le message originel. Ainsi il faut calculer $17^{17} \mod 35$. Or :

$17^2 \equiv 9 \mod 35$ puis $17^4 \equiv 9^2 \equiv 11 \mod 35$ ,$17^8 \equiv 16 \mod 35$ et $17^{16} \equiv 16^2 \equiv 11 \mod 35$

et alors

$$17^{17} = 17^{16} \times 17 \equiv 11 \times 17 \equiv 12 \mod 35$$

On retrouve bien le message décrypté originel.

---

## Exercice 8 – Échec de protocole de RSA : le problème du modulus commun

*Le but de l'exercice est de décrire l'un des cas où Oscar peut facilement casser RSA. En pratique, le protocole de la société RSA permet évidemment d'éviter cette attaque.*

On suppose que Bob et Bill ont publié respectivement les clefs RSA $(b_1, n)$ et $(b_2, n)$. Ils ont donc publié (sans le savoir) le même modulus $n = pq$.
Alice désire envoyer le même message $x \in [\![0, n-1]\!]$.

1. Quels sont les messages $y_1$ et $y_2$ qu'elle envoie respectivement à Bob et Bill ?
2. On suppose que $b_1$ et $b_2$ sont premiers entre eux, ainsi que $y_1$ et $n$ (ce qui a de grandes chances d'être vérifié en pratique). Oscar intercepte les deux messages cryptés $y_1$ et $y_2$. Expliquer comment il peut facilement retrouver $x$.
3. Que doit faire la société RSA ?

---

**Corrigé**

1. D'après le protocole RSA, Alice envoie $y_1 = x^{b_1} \mod n$ à Bob, et $y_2 = x^{b_2} \mod n$ à Bill.
2. $b_1$ et $b_2$ sont premiers entre eux. Il existe donc $u$ et $v$ deux entiers tels que $b_1 u + b_2 v = 1$, qu'Oscar peut calculer rapidement par l'algorithme d'Euclide étendu. Il suffit à Oscar de calculer $y_1^u y_2^v \mod n$. En effet :
$$\begin{aligned} y_1^u y_2^v &\equiv x^{b_1 u} x^{b_2 v} \mod n \\ &\equiv x^{b_1 u + b_2 v} \mod n \\ &\equiv x \mod n \end{aligned}$$
Lors d'une recherche de $u$ et $v$, l'un des deux (par exemple $u$) est négatif. Dans ce cas, le problème, puisque $u \leqslant 0$ est de calculer $y_1^u \mod n$. Or, par hypothèse, $y_1$ est premier avec $n$. Par l'algorithme d'Euclide étendu, on déterminer $z_1$ l'inverse de $y_1$ modulo $n$. Alors
$$y_1^u \equiv z_1^{-u} \mod n$$
et cette fois-ci, $-u \geqslant 0$ et se calcule habituellement.
3. Pour éviter cette attaque, il est important que deux personnes n'aient pas le même modulus commun (i.e. que le choix des nombres premiers $p$ et $q$ soient le plus aléatoire possible). Ainsi, sans modulus commun, cette attaque ne peut fonctionner.

---

## Exercice 9 – Théorème chinois et application en cryptanalyse

*Il s'agit ici d'étudier une application du théorème des restes chinois : la cryptanalyse de RSA dans le cas de petit exposants de chiffrement, erreur qu'on évite en suivant le protocole de la société RSA.*

Dans l'implémentation du système RSA, il est tentant de choisir un petit exposant public de chiffrement (car le chiffrement $y = x^b \mod n$ est d'autant plus rapide que $b$ est petit). Nous allons voir que c'est une erreur qu'il faut éviter, car Oscar retrouve alors $x$ facilement dans une situation assez fréquente.
Pour fixer les idées, supposons que Bob, Bill et Bart aient tous trois publié l'exposant public de chiffrement $b = 3$. Par ailleurs, les modulus publiés respectivement par Bob, Bill et Bart sont $n_1, n_2, n_3$, supposés premiers entre eux deux à deux. Alice souhaite envoyer le même message $x$ à Bob, Bill et Bart. Oscar intercepte les trois messages chiffrés $y_1, y_2, y_3$.

1. Expliquer comment Oscar peut retrouver $x$ en utilisant le théorème des restes chinois. On décrira précisément les opérations effectuées par Oscar en justifiant qu'elles sont faisables en pratique.
2. Comment cette méthode de cryptanalyse se généralise-t-elle lorsque l'exposant $b$ est quelconque ? Expliquer pourquoi un grand exposant $b$ protège contre cette attaque d'Oscar.

---

**Corrigé**

1. Concrètement, en utilisant les notations du cours sur le théorème des restes chinois, Oscar dispose de $x^3 \mod n_1, x^3 \mod n_2, x^3 \mod n_3$, avec $n_1, n_2, n_3$ deux à deux premiers entre eux. La théorème des restes chinois garantie qu'il existe une unique solution $y = x^3 \in \mathbb{Z}_m$ (où $m = n_1 n_2 n_3$) vérifiant $y = (y_1 \mod n_1, y_2 \mod n_2, y_3 \mod n_3)$. $x$ est alors le message envoyé par Alice.
Ainsi, $x^3$ sera calculé par la formule
$$\sum_{i=1}^{k} y_i n_i' n_i''$$
où $n_i' = n_1 \dots n_{i-1} n_{i+1} \dots n_k$ et $n_i''$ est l'inverse de $n_i'$ modulo $n_i$.

En pratique, Oscar calcule (par l'algorithme d'Euclide étendu) l'inverse de $n_1'$, $n_2'$ et $n_3'$ et calcule $y$ par la formule (3).
Or $y$ représente $x^3 \mod m$ ($x$ étant le message cherché). Mais par définition de RSA, $x < n_1, x < n_2$ et $x < n_3$ donc $x^3 < n_1 n_2 n_3 = m$. Donc $x^3 \mod m = x^3$ et il suffit de calculer $x = y^{1/3}$ (racine cubique classique).

2. Concrètement, pour pouvoir calculer $x$ aussi efficacement, il faut appliquer le théorème chinois (ce qui va relativement vite, avec Euclide étendu) mais aussi résoudre $y = x^b \mod m$ d'inconnue $y$. Le plus efficace est de connaitre $b$ valeurs (même message envoyé à $b$ personnes) : ainsi, il suffit de calculer la racine $b$-ième du nombre obtenu.
Si on ne dispose pas de $b$ valeurs, mais moins, on trouvera $y^b \mod m$ mais rien ne garantie que $y^b < m$. Ainsi, connaître $y^b \mod m$ ne permet de pas de revenir à $y^b$ puis à $y$. C'est là tout l'intérêt de choisir un exposant de chiffrement suffisamment grand.