IUT Littoral **2019 - 2020**
Info 2A **Math – C++**

TP 7 - Chiffrement RSA

Définition 1. Un *nombre premier* est un entier p supérieur ou égale à 2 dont les seuls diviseurs positifs sont 1 et p

1 L'algorithme d'Euclide

Définition 2. Soient a et b des entiers non tous deux nuls. Le plus grand entier qui divise a et b s'appelle le plus grand commun diviseur de a et de b et se note $\mathrm{pgcd}(a, b)$.

Proposition 3. *Soient a et b des entiers positifs non tous deux nuls. Si r est le reste de la division euclidienne de a par b, alors $\mathrm{pgcd}(a, b) = \mathrm{pgcd}(b, r)$.*

Exercice 4. Calculer $\mathrm{pgcd}(585, 247)$.

Exercice 5. Écrire une fonction récurssive `Entier pgcd_rec(Entier a,Entier b)` calculant les pgcd de deux nombres positifs `a` et `b`.

Soient a et b des entiers positifs non tous deux nuls. On construit une suite r_i en posant $r_0 = a$, $r_1 = b$ et pour $k \geqslant 2$.

$$\begin{aligned} r_k &= \text{reste de la division euclidienne de } r_{k-2} \text{ par } r_{k-1} \\ &= r_{k-2} - q_{k-1} r_{k-1} \quad \text{où} \quad q_{k-1} = \left\lfloor \frac{r_{k-2}}{r_{k-1}} \right\rfloor \end{aligned}$$

Il existe alors un unique entier N tels que

$$b = r_1 > r_2 > \ldots > r_{N-1} > r_N = 0,$$

et nous avons alors $\mathrm{pgcd}(a, b) = r_{N-1}$.

Exercice 6. Ecrire une procédure non récurssive `Entier pgcd(Entier a,Entier b)`calculant les pgcd de deux nombres positifs `a` et `b`.

Définition 7. Soient a et b des entiers non tous deux nuls. On dit que a et b sont *premiers entre eux* si $\mathrm{pgcd}(a, b) = 1$.

2 Théorème de Bézout

Théorème 8. *Si a et b sont des entiers positifs, alors il existe des entiers u et v tels que* $\mathrm{pgcd}(a, b) = au + bv$.

En s'inspirant de l'algorithme d'Euclide, on construit deux suites u_k et v_k de $\mathbb{Z}$ vérifiant $au_k + bv_k = r_k$ pour tout $k \leqslant N$. On pose alors $u_0 = 1, v_0 = 0$ et $u_1 = 0, v_1 = 1$ ainsi que

$$u_k = u_{k-2} - q_{k-1} u_{k-1} \quad \text{et} \quad v_k = v_{k-2} - q_{k-1} v_{k-1}.$$

On a alors $\mathrm{pgcd}(a, b) = r_{N-1} au_{N-1} + bv_{N-1} = r_{N-1} = \mathrm{pgcd}(a, b)$ et il suffit de prendre $u = u_{N-1}$ et $v = v_{N-1}$.

Exercice 9. Déterminer des entiers u et v tels que $585u + 247v = \mathrm{pgcd}(585, 247)$.

3 Congruences

Définition 10. On dit que deux entiers a et b sont congrus modulo n, lorsqu'il donne le même reste par la division euclidienne par n. On note alors $a \equiv b \mod n$.

Ou de manière équivalente, deux entiers a et b sont congrus modulo n lorsque $a - b$ est un multiple de n, c'est-à-dire, lorsqu'il existe $k \in \mathbb{Z}$ tel que $a - b = nk$.

Exemple. Quels que soient $a \in \mathbb{Z}$ et $n \in \mathbb{N}$, on a
– $a \equiv a \mod n$
– $a \equiv r \mod n$ si r est le reste de la division euclidienne de a par n.

Proposition 11. *Soeint $a, b, c, d \in \mathbb{Z}$ et $n \in \mathbb{N}$, on a*
– *si $a \equiv b \mod n$ et $b \equiv c \mod n$ alors $a \equiv c \mod n$.*
– *si $a \equiv b \mod n$ et $c \equiv d \mod n$ alors $a + c \equiv b + d \mod n$, $a - c \equiv b - d \mod n$ et $ac \equiv cd \mod n$.*

4 Indicatrice d'Euler

Définition 12. Un entier $a \in \mathbb{Z}$ est dit inversible modulo n s'il existe $b \in \mathbb{Z}$ tel qu'on ait $ab \equiv 1 \mod n$.

Exercice 13. Quels sont les entiers de $\{0, \dots, 12\}$ qui sont inversibles modulo 12 ?

Définition 14. Pour $n \geqslant 2$, on note $\phi(n)$ le nombre d'élément de $\{0, \dots, n-1\}$ qui sont inversibles modulo n.

Exercice 15.
a. Déterminer $\phi(4)$, $\phi(5)$, $\phi(6)$ et $\phi(7)$.
b. Que vaut $\phi(n)$ lorsque n est un nombre premier.

Proposition 16. *Soient p et q deux nombres premiers distincts. On a $\phi(pq) = (p-1)(q-1)$.*

Exercice 17. Démontrer la proposition précédente.

Exercice 18. Calculer $\phi(35)$.

5 Chiffrement RSA

Alice veut envoyer un message à Bob de manière sécurisée : à la réception du message seul Bob doit pouvoir le déchiffrer. Pour ce faire Bob génére de manière aléatoire deux grands nombres premiers p et q. Il calcule ensuite leur produit $n = p \times q$ ainsi que $\phi(n) = (p-1) \times (q-1)$. Il choisit alors un troisième nombre premier e qui ne divise pas $\phi(n)$ (par convention e vaut souvent 3 ou 65537). A l'aide de l'algorithme d'Euclide étendue, il calcule $0 \leqslant d < \phi(n)$ tel que $ed \equiv 1 \mod \phi(n)$. Il compose alors deux clés : une publique qu'il transmet à Alice (en clair) et une secrète qu'il conserve pour lui. La clé publique est composée de n et de e. La clé secrète est composée de n et de d. Les autres entiers ne sont plus nécessaires. À la réception de la clé publique (n, e), Alice chiffre sont message $M \in [0, n-1[$ par $C = M^e \mod n$ (exponentiation modulaire) qu'elle transmet à Bob. Bob déchiffre alors le message C en posont $M' = C^d \mod n$. Nous avons alors $M' = M$.

Pour pouvoir utiliser le chiffrement RSA il nous faut détailler quelques points :
— comment calculer d à partir de e et $\phi(n)$;

— l'exponentiation modulaire rapide (pour effectuer des exponentiations modulaires efficacement) ;
— la génération de nombre aléatoire (ou presque) ;
— comment tester la primalité (ou presque) d'un nombre de manière efficace.

Pour calculer d il suffit d'appliquer l'algorithme d'Euclide étendu à e et $\phi(n)$. Comme e est premier et ne divise pas $\phi(n)$ les entiers e et $\phi(n)$ sont premiers entre eux. Il existe alors u et v dans $\mathbb{Z}$ vérifiant $u \times e + v \times \phi(n) = 1$. Les entiers u et v sont obtenus à l'aide de l'algorithme d'Euclide étendu. Pour d on prend alors le reste de la diviosion euclidienne de u par $\phi(n)$.

6 Exponentiation modulaire

Exercice 19. Caculer de manière naive $4^{10} \mod 13$. Quel est le nombre d'étapes ?

Une autre stratégie permet de réduire le nombre d'étapes. On part de $4^{10} = 4^5 \times 4^5$. Pour calculer 4^{10} il suffit alors de connaitre 4^5. De même on a $4^5 = 4^2 \times 4^2 \times 4$ et $4^2 = 4 \times 4$. On obtient alors :

$$
\begin{aligned}
4^2 &= 4 \times 4 = 16 \equiv 3 \mod 13 \\
4^5 &= 4^2 \times 4^2 \times 4 \equiv 3 \times 3 \times 4 \equiv 36 \equiv 10 \mod 13 \\
4^{10} &= 4^5 \times 4^5 \equiv 10 \times 10 \equiv 100 \equiv 9 \mod 13
\end{aligned}
$$

Avec cette méthode seule 4 multiplicatins sont nécessaires.

Exercice 20. A l'aide de la méthode précédente, imaginer un algorithme `exp_mod_rapide` prenant en entrée tois entiers a, p et n et retournant l'unique entier b vérifiant $b \equiv a^p \mod n$.

7 Générateurs pseudo-aléatoires

Une suite de bits est dite *aléatoire* si elle est imprévisible, c'est-à-dire qu'aucune stratégie effective ne peut mener à un gain infini si l'on parie sur les termes de la suites. Un ordinateur ne peut pas créer de telles suites. Une suite de bits $(a_n)_{n\in\mathbb{N}}$ est dite *pseudo-aléatoire* si elle est produite par un algorithme et s'il est algorithmiquement "difficile" de prévoir avec une probabilité $\geqslant \frac{1}{2}$ le bit a_{n+1} à partir des premiers bits $a_1, ..., a_n$.

D'un point de vu concret les suites pseudo-aléatoires ont un gros défaut, liés au fait que les ressources d'un ordinateur sont limités. Le générateur pseudo-aléatoire retrouvera alors le même état interne au moins deux fois et la suite sera donc périodique.

En 1946, John Von Neumann propose le générateur "middle-square". Son principe est le suivant :

1. on choisit en entier n à quatre chiffres
2. on note m le nombre formé des quatre chiffres au milieu de n^2
3. on retourne m
4. on pose $n = m$ et on retourne à l'étape 2

Exercice 21.
a. Essayer cette algorithme avec $n = 1111$.
b. Donner un majorant de la périodicité de la suite retournée.
c. Quel est la périodicité minimale ? Pour quel n est elle atteinte ?

En 1948, Derrick Henry Lehmer à introduit les générateurs congruentiels linéaires. Le principe est le suivant, on choisit des entiers a, c et n que l'on garde secret. Puis on choisi x_0 et on calcul x_{i+1} à partir de x_i, en posant $x_{i+1} = (ax_i + b) \mod n$.

Exercice 22.
a. Tester ce générateur pour $a = 3$, $b = 4$, $n = 8$ et $x_0 = 5$.
b. Quelle est la périodicité maximale d'un générateur congruentiel linéaire ?

Supposons que n soit connu, alors pour retrouver a et b il suffit ce connaître les trois premiers termes x_0, x_1 et x_2. On pose $y_1 = x_1 - x_0$, $y_2 = x_2 - x_1$.

Exercice 23.
a. Établir la relation $y_2 \equiv ay_1 \mod n$.
b. Comment retrouver a ? puis b ?
b. Retrouver les paramètres a, b qui on permis de créer la suite $97, 188, 235, 293, 604, 596, 412$ avec $n = 1023$.

Le choix des paramètres a, b et n d'un générateur congruentielle linéaire influence l'efficacité du générateur est doit donc être fait avec précaution. Un bon choix est celui utilisé par `standard minimal` : $a = 16807$ et $n = 2^{31} - 1$.

8 Test de primalité

Donnons sans démonstration le petit théorème de Fermat :

Théorème 24. *Si p est un nombre premier et a un nombre non divisible par p on a*

$$a^{p-1} \equiv 1 \mod p$$

La réciproque de ce théorème est fausse : il existe des entiers n non premier tels que pour tout entier a premier avec n on ait $a^{n-1} \equiv 1 \mod n$ ($n = 561$ par exemple). Un tel entier n est appelé nombre de Carmichael.
Supposons que l'on veuille tester si un nombre n est premier. On choisit alors des entiers $t_0, ..., t_{k-1}$ appelés témoins. Si pour une valeur de i dans $\{0, ..., k-1\}$ on a $t_i^{n-1} \not\equiv 1 \mod n$ alors n n'est pas premier. Si on ne trouve pas de tel i alors n est probablement premier. Ce test est appelé test de primalité de Fermat.
Les nombres de Carmickael sont détectés comme probablement premier par ce test or ils ne sont par premiers. Le test n'est donc pas sûr à 100%. Néanmoins les nombres ce Carmickael sont assez rares et si le nombre de témoins est important il est peut probable de tomber sur un nombre non premier et détecter probablement premier.
Nous allons utiliser ce test sur des nombres codés sur 32 bits avec les témoins $2, 3, 5, 7, 11, 13$ et 17. Parmi les entiers entre 2 et $2^{32} - 1$, exactement 203280702 sont détectés probablement premier. Parmis eux seulement 481 ne sont pas premier. La liste complète des exceptions est diponible sur ma page web.