TD noté 1

Date limite de remise du TD : vendredi 24 octobre 2014 à 23h59.

Le TD se fait par groupe de 2 étudiants (au maximum) et les fichiers à remettre doivent être envoyées au plus tard vendredi 24 octobre à minuit par courriel à l'adresse électronique "cristina.onete@gmail.com". Si vous devez programmer un algorithme, cela peut se faire en C, C++, Python ou Java. Merci de joindre le code de votre algorithme aux fichiers remis dans le cadre du TD. Dans le cas d'exercice « à la main » (ne requérant pas l'utilisation d'un ordinateur), merci d'expliquer en détails le raisonnement suivi.

Rappel : tout plagiat est formellement interdit et bien qu'il soit naturel que vous puissiez parfois discuter avec vos camarades oralement sur comment attaquer ou résoudre un problème, il est formellement interdit d'échanger des fichiers de code.

Le total des points du TD est de 40 plus 5 points en bonus pour l'exercice 4b qui est optionnel car « difficile ». Si jamais vous réussissez cet exercice, les 5 points de bonus peuvent compte en plus des 40 autres points pour la moyenne globale des TDs.

1. Ecrivez un programme qui permet de tester si un nombre donné en entrée est premier ou non. Si ce nombre est composite (non-premier), votre programme doit également retourner la liste complète des diviseurs de ce nombre. (5 points)

Utilisez votre programme pour déterminer lesquels parmi les nombres suivants sont premiers. Pour les nombres qui sont composites, donnez leur décomposition complète en diviseurs.

a) 19 (0.25 points)
b) 121 (0.25 points)
c) 1193 (0.5 points)
d) 2577 (0.5 points)
e) 55133 (0.5 points)
f) 1522363 (0.5 points)
g) 761183 (0.5 points)
h) 1415819 (0.5 points)
i) 26696683043 (0.5 points)
j) 202247599 (0.5 points)
k) 808990396 (0.5 points)

2. Factorisez à la main (sans utiliser l'ordinateur) $N = pq$ (où p et q sont des nombres premiers) pour chacun des cas suivants :

a) $N = 490687$ et $\varphi(N) = 489240$ (1 point)
b) $N = 7095577$ et $\varphi(N) = 7090020$ (1 point)
c) $N = 36605969$ et $\varphi(N) = 36593700$ (1 point)

3. Démontrez que l'algorithme SmallDiff décrit dans les transparents du TD 2 Octobre fonctionne correctement et qu'il retourne bien la factorisation de *N = pq*. Implémentez cet algorithme-ci pour une valeur de $max = 5000$.

Indice : si on veut vérifier si $(m^2 + N)$ est un carré parfait, on peut commencer par calculer le plus grand nombre entier qui est encore plus petit que $\sqrt{N}$ –noté $\lfloor\sqrt{N}\rfloor$ – et puis on teste, pour *i = 1* à *m,* si $m^2 + N = (\lfloor\sqrt{N}\rfloor + i)^2$. Une fois qu'une valeur *i* est trouvée pour laquelle cette relation est vraie, soit $s = \lfloor\sqrt{N}\rfloor + i$. Ainsi maintenant, on a un entier *s* tel que $m^2 + N = s^2$. (5 points)

En utilisant cet algorithme, factorisez les valeurs suivantes de *N* (vous pouvez considérer une valeur de *max* égale à 5000). Merci d'inclure les résultats de chaque factorisation dans votre rapport.

a) *N = 445243* (1 point)
b) *N = 1578893* (1 point)
c) *N = 3832951* (1 point)
d) *N = 27023201* (1 point)
e) *N = 63486623* (1 point)

4. A. Implémentez l'algorithme étendu d'Euclide qui permet de déterminer le plus grand diviseur commun de deux nombres *a* et *b*, ainsi que des nombres entiers *s* et *t* tels que: *sa + tb = GCD(a,b)*. Voir l'adresse suivante pour la description de cet algorithme : http://en.wikipedia.org/wiki/Extended_Euclidean_algorithm . (5 points)

B. (Difficile) Amélie chiffre un message *M* pour Baptiste avec un chiffrement RSA « standard ». La valeur publique de *N* est 42321743. Pour se rassurer que le message *m* arrive à Baptiste, Amélie chiffre le message deux fois en utilisant deux clés publiques (*e*) différentes. L'attaquant intercepte les valeurs suivantes. (5 points)

- $e_0 = 1634083, c_0 = 36020259$
- $e_1 = 25237, c_1 = 23841697$

En utilisant votre implémentation de l'algorithme d'Euclide étendu et sans factoriser *N*, retrouvez le message *m* chiffré par Amélie. Expliquez votre stratégie en détail.

Indice : Rappelez-vous que $c_0 = M^{e_0}(mod\ N)$ et $c_1 = M^{e_1}(mod\ N)$, et que si deux nombres *a* et *b* sont co-premiers, leur plus grand diviseur commun est égale à *1.*

5. Implémentez la méthode Pollard $(p - 1)$ et factorisez le module RSA $N = 5797667723$. (4 points)

6. Montrez comment on peut retrouver le message chiffré si on utilise la fonction de prétraitement SAEP. Comment est-ce qu'on peut vérifier que le padding a le format requis (*W(m, r)*)? (3 points)

7. Implémentez l'algorithme Square-and-Multiply pour faire l'exponentiation dans RSA. Réduisez *mod N* à chaque itération (après faire la mise au carré ou la mise au carré et la multiplication). (2 points)