USTHB, Faculté d'Informatique Master IL

Epreuve de Sécurité Informatique

Question 1 (6 points). Montrer comment paramétrer le RSA (choix des clés secrète et publique). Si B choisit $p = 29, q = 31$. Peut-il choisir $b = 5 ? 11 ?$ Chiffrer et signer le message clair HACK dans le chiffrement par blocs avec m=2. On utilisera les algorithmes d'Euclide et d'exponentiation rapide ? Le chiffrement par blocs est intéressant du point de vue MEPS. Quel est son apport du point de vue sécurité ? Quels sont ses avantages et inconvénients par rapport aux systèmes symétriques de type DES ou AES.

Sol. $n = pq = 29 \times 31 = 899$, $\varphi(n) = (p-1)(q-1) = 840$ .On ne peut choisir $b = 5$ car $PGCD(840,5) = 5 \neq 1$. Par contre $b = 11$ est une clé valide car $PGCD(840,11) = 1$. On numérise le texte HACK qui donne 70 210

La clé secrète est $a = 11^{-1} \bmod 840 = 611$

$70^{11} \bmod 889 = 597$ et $210^{11} \bmod 889 = 197$

Du point de vue MEPS, le chiffrement par blocs (ou paquets) permet d'améliorer le débit. Du point de vue sécurité, un caractère n'est pas toujours chiffré de la même façon.

Symétrique sont réputés pus rapide que l'asymétrique, par contre il faut un canal sûr (donc une autorité de confiance) pour l'échange des clés secrètes. Avec l'asymétrique, l'échange de clés se fait sans aide extérieure.

Question 2(2 points). Donner les similitudes et différences entre les algorithmes (tests) de primalité et les algorithmes de factorisation.

Sol.

Test de primalité : sert à déterminer si un grand entier est premier

Algorithme de factorisation : sert à déterminer au moins un facteur premier d'un entier.

Un test de primalité ne permet pas de trouver concrètement un facteur premier. Mais s'il détermine qu'un entier est premier, c'est qu'il n'est pas factorisable. Un algorithme de factorisation, s'il founit un facteur premier, c'est que l'entier n'est pas premier.

Question 3(2 points). Expliquer pourquoi si dans l'utilisation du RSA on fait fuiter le nombre généré par la fonction d'Euler, alors « casser » le RSA réputé NP-sûr devient NP-facile.

Sol. Il suffit de résoudre le système de deux équations à deux inconnues : (i) $n = pq$; (ii) $\varphi(n) = (p-1)(q-1)$ (rappelons que n est publique et donc connu). En écrivant $q = n/p$ et après substitution dans (ii), on obtient une équation du second degré : $p^2 - (n - \varphi(n) + 1)p + n = 0$ qui admet deux racines p et q. Si un cryptanalyste parvient à

connaître $\varphi(n)$, il peut donc retrouver la factorisation. Cela signifie que le calcul de $\varphi(n)$ n'est pas plus facile que la factorisation.

Question 4. (6 points). L'une des attaques (théorique) sur le système RSA est la factorisation du modulus. Montrer commment procéder à ce type d'attaque à l'aide de la technique (« grossière ») rho de Pollard si $n = 247$. Pourquoi le fait de trouver un facteur non trivial du modulo permet-t-il de « casser » le RSA ?

Question 5. (4 points). On a intercepté deux messages chiffrés $y_1 = 2$ et $y_2 = 8$ avec le RSA de modulo public $n = 15$ envoyés à deux utilisateurs différents de clés publiques respectives $b_1 = 5$ et $b_2 = 3$. (i) Retrouver le message clair. (ii) Expliquer pourquoi cela est possible ?

Sol. Sécurité. Casser le RSA revient à factoriser de gands entiers ou ce qui est équivalent au problème de racine modulaire ce qui est réputé calculatoirement difficile.

L'algorithme rho de Pollard avec la fonction $f(x) = x^2 + 1 mod 247$donne

a	2	5	26	183
b	5	26	183	145
p $= PGCD(a - b)$	1	1	1	19

Et on retrouve la factorisation 247=19x13

Sol. b_1 et b_2 sont premiers entre eux, alors O peut retrouver le message clair x sans connaître la factorisation de n. En effet, il suffit de calculer $u = -1, v = 2$ (à l'aide de l'algorithme d'Euclide étendu) tels que $ub_1 + vb_2 = 1$. Par suite, $x = x^1 = x^{ub_1 + vb_2} = \left(x^{b_1}\right)^u \left(x^{b_2}\right)^v = y_1^u y_2^v = 2$. Il suffit donc à O de calculer $y_1^u y_2^v$ pour retrouver le message clair.