L3 Calcul Formel
Université de Lorraine

TP 2 : Cryptographie à clés publique et privée

Clément Dell'Aiera

1 Complexité

Au premier TP, vous avez implémenté les algorithmes d'Euclide et d'Euclide étendu. Ces algorithmes se basent sur la propriété $pgcd(x, y) = pgcd(y, r)$ où r est le reste de la division euclidienne de x par y. La terminaison de l'algorithme fournit entre autre une preuve de l'identité de Bézout : il existe des entiers u et v tels que $ux + vy = pgcd(x, y)$. Nous allons nous intéresser au coût de ces algortihmes.

1. Soient x et y deux entiers. Montrer que x et y sont premiers entre eux ssi il existe u et v dans $\mathbb{Z}$ tels que $ux + vy = 1$.

   La suite de Fibonacci est définie par

   $$\begin{cases} F_0 = 0, F_1 = 1 \\ F_{n+1} = F_n + F_{n-1} \end{cases}$$

2. Calculer les 7 premiers termes de la suite.
3. Montrer que, si $n > 0$, alors

   $$\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^n = \begin{pmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{pmatrix}$$

   En déduire $F_{n+1}F_{n-1} - F_n^2 = (-1)^n$.
4. Montrer que $F_{n+m} = F_{n+1}F_m + F_nF_{m-1}$. En déduire : $pgcd(F_{n+m}, F_m) = pgcd(F_m, F_n)$, puis $pgcd(F_n, F_m) = F_{pgcd(n,m)}$.
5. Soit $\varphi = \frac{\sqrt{5}+1}{2}$. Montrer que

   $$F_n = \frac{1}{\sqrt{5}}(\varphi^n - (-\varphi)^{-n}),$$

   et en déduire que F_n est l'entier le plus proche de $\frac{\varphi^n}{\sqrt{5}}$.

6. Montrer la proposition suivante, que l'on doit à Lamé (1845).
   Soient x et y deux entiers tels que $0 < y < x$ et soit d leur *pgcd*. Si l'algorithme d'Euclide partant de (x, y) s'arrête au bout de n pas, on a

$$x \geq dF_{n+2}, \quad y \geq dF_{n+1}.$$

   Est-ce optimal ?

7. Montrer que $n \leq \frac{3}{2} log(F_{n+1}) + 1$. En déduire une majoration du nombre de pas de l'algorithme d'Euclide.

   Pour estimer le coût de l'algorithme d'Euclide, nous allons d'abord définir le modèle dans lequel nous nous plaçons, i.e. établir le coûts de chaque opération arithmétique.

   • Le coût d'une addition ou soustraction $m \pm n$ est majorée par $c_+ \max(\log|m|, \log|n|)$.
   • Le coût d'une multiplication $m \times n$ est majorée par $c_\times \log|m| \log|n|$.
   • Le coût d'une division euclidienne de m par n, où $0 < n \leq m$, est majorée par $c_\% log|m| \log|m/n|$.

   Ce modèle s'appelle le modèle à coûts bilinéaires. Montrer que dans ce modèle, le coût du pgcd de x et y avec $0 \leq x < y$ est majoré par $c_\%(\log y)^2$.

2 Cryptosystème de Rabin

Alice et Bob veulent échanger des messages cryptés. Ils utilisent pour cela le crytposystème à clé publique de Rabin. Voici les 3 étapes de ce protocole.

•Génération des clés
A choisit 2 grands entiers premiers p et q, et calcule $N_A = pq$. La clé publique de A est N_A, sa clé privée (p, q).

•Chiffrement
B souhaite envoyer un message crypté à A. Il récupère sa clé publique N_A, représente le message comme un entier m entre 0 et $N_A - 1$, et envoie m^2 modulo N_A à A.

•Déchiffrement
A reçoit c, et souhaite calculer une racine carrée. A calcule donc les racines carrées $\pm r_p$ de c modulo p, et $\pm r_q$ de c modulo q. Le théorème chinois donne alors les racines de c modulo N_A : m_1, $N_A - m_1$, m_2, $N_A - m_2$. A priori, A ne peut décider lequel de ces 4 messages est celui que B veut transmettre. Toutefois, ce problème peut être corrigé par redondance : B peut copier les 6 derniers bits de son message à la fin d'icelui, et A choisit le message qui présente une redondance sur ses 6 derniers bits.

Pour calculer une racine carrée de c modulo p ou q, nous utiliserons l'algorithme de Tonelli-Shanks.

Tonelli-Shanks
Entrées : p nombre premier impair, a carré modulo p.
Sortie : une racine carrée de a modulo p.

1. Trouver b entre 2 et $p-1$ qui ne soit pas carré modulo p.
2. Calculer $s \in \mathbb{N}$ et t impair tels que $p = 2^s t$.
3. $z \leftarrow b^t$
4. $m \leftarrow 0$
5. Pour j de 0 à $s-1$ faire :
   si $(a^t z^m)^{2^{s-1-j}} = -1 \ (mod\ p)$ alors $m \leftarrow m + 2^j$ fin si.
   fin pour
6. Retourner $a^{\frac{t+1}{2}} z^{\frac{m}{2}}$.

1. Implémenter l'algorithme de Tonelli-Shanks.
2. Implémenter des fonctions qui génère des clés, chiffre et déchiffre un message dans le protocole de Rabin.