Université de Caen **Année 2007-2008**
UFR de Sciences **1ère session**

EI61T Approfondissemnt en algorithmique
durée 2h00

Les notes de cours et TD sont autorisées.
Les deux parties sont indépendantes, correspondent aux parties du cours "cryptologie"" et "algorithmique répartie" et doivent être rendues sur des copies séparées.

Chaque candidat doit, au début de l'épreuve, porter son nom dans le coin de la copie qu'il cachera par collage après avoir été pointé. Il devra en outre porter son numéro de place sur chacune des copies, intercalaires, ou pièces annexées.

1 Cryptologie (à rédiger sur une copie séparée)

Soient a et b deux entiers tels que $a \geq b > 0$. En considérant la division euclidienne classique, on définit deux suites d'entiers $r_0, r_1, \ldots, r_{\ell+1}$ et $q_1, q_2, \ldots, q_\ell$ pour un entier $\ell > 0$ de la façon suivante :

$$\begin{array}{rcll} a & = & r_0, & \\ b & = & r_1, & \\ r_0 & = & r_1 q_1 + r_2 & (0 < r_2 < r_1) \\ & \vdots & & \\ r_{i-1} & = & r_i q_i + r_{i+1} & (0 < r_{i+1} < r_i) \\ & \vdots & & \\ r_{\ell-2} & = & r_{\ell-1} q_{\ell-1} + r_\ell & (0 < r_\ell < r_{\ell-1}) \\ r_{\ell-1} & = & r_\ell q_\ell & (r_{\ell+1} = 0) \end{array}$$

On a $r_\ell = \text{pgcd}(a, b)$.

On considère la suite de Fibonacci définie par $F_0 = 0$,$F_1 = 1$ et $F_{n+2} = F_{n+1} + F_n$ pour $n \geq 0$.

1. Calculez les 11 premiers termes de la suite, et calculez $\text{pgcd}(F_{10}, F_9)$.

   Vous allez maintenant montrer que le nombre d'itérations de l'algorithme d'Euclide ℓ vérifie

   $$\ell \leq \log b / \log \phi + 1 \tag{1}$$

   où $\phi := (1 + \sqrt{5}/2) \approx 1,62$ est le nombre d'or.
2. Quelle est l'équation du second degré vérifiée par ϕ ?
3. Montrez que pour $i = 2, \ldots, \ell - 1$, $r_{\ell-i} \geq r_{\ell-(i-1)} + r_{\ell-(i-2)}$.
4. Montrez par récurrence et en utilisant les deux questions précédentes, que pour $i = 0, 1, \ldots, \ell - 1$

   $$r_{\ell-i} \geq \phi^i.$$

5. Montrez le résultat annoncé (1) en posant $i = \ell - 1$ dans la relation précédente.
6. Connaissant le nombre d'itérations de l'algorithme d'Euclide, et en utilisant le fait que la complexité d'une division euclidienne est quadratique, donnez une estimation de la complexité de l'algorithme d'Euclide sur deux entiers a et b de k bits (donnez cette complexité en fonction de k).

7. Rappelez le système de chiffrement RSA.

   Vous allez montrer, en étudiant la relation $ed = 1 \mod \varphi(N)$, que le théorème précédent permet, grâce à l'algorithme de calcul des réduites, de casser RSA lorsque l'exposant secret est trop petit.
   Si $ed = 1 \mod \varphi(N)$, alors il existe $k \in \mathbb{Z}$ tel que $ed = 1 + k\varphi(N)$.

8. Rappelez la définition de $\varphi(n)$ et calculez $\varphi(77)$.

Quelques faits sur les fractions continues :
On note à présent

$$[q_0, q_1, \ldots, q_m] = q_0 + \cfrac{1}{q_1 + \cfrac{1}{\ddots \cfrac{}{q_{m-1} + \cfrac{1}{q_m}}}}.$$

Soit $f \in \mathbb{R}$, on pose

$$\begin{cases} q_0 = \lfloor f \rfloor, & r_0 = f - q_0 \\ q_i = \left\lfloor \dfrac{1}{r_{i-1}} \right\rfloor, & r_i = \dfrac{1}{r_{i-1}} - q_i, \quad i = 1, 2 \ldots \end{cases}$$

tant que r_{i-1} est non nul (notez que les q_i et r_i se calculent grâce à l'algorithme d'Euclide).

On a donc $f = [q_0, q_1, \ldots, q_{n-1}, q_n + r_n]$ $\forall$ $n \in \mathbb{N}$.

La suite, finie ou infinie, des q_i s'appelle le développement en fractions continues de f. *Les nombres rationnels $f_i = \dfrac{n_i}{d_i} = [q_0, q_1, \ldots, q_{i-1}, q_i]$ sont dits* réduites ou développement partiel *de f.*

Le théorème fondamental suivant (admis) va vous permettre de réaliser une cryptanalyse de RSA :

Théorème 1 *Supposons que* $\mathrm{pgcd}(a, b) = \mathrm{pgcd}(c, d) = 1$, *et*

$$\left| \frac{a}{b} - \frac{c}{d} \right| \leq \frac{1}{2d^2}.$$

Alors c/d est l'une des réduites du développement en fractions continues de a/b.

9. Montrez que

$$\left| \frac{e}{N} - \frac{k}{d} \right| \leq \frac{\alpha}{\sqrt{N}} \quad \text{pour une certaine constante } \alpha$$

   en supposant (c'est vrai en pratique) que p et q valent environ $\sqrt{N}$ tous les deux (utiliser une majoration grossière de $|N - \varphi(N)|$).

10. En déduire alors à quelle condition et comment d peut être retrouvée en utilisant le développement en fractions continues d'un certain rationnel et donnez une estimation de la complexité de l'attaque.

2 Algorithmique répartie (à rédiger sur une copie séparée)

2.1 Introduction

Une troupe de robots mobiles évolue sur un terrain. Ils communiquent entre eux par ondes radio (WiFi). Ces communications ne peuvent être reçues que si l'émetteur n'est pas trop éloigné, de la sorte que, selon leurs évolutions, deux robots pourront ou non communiquer entre eux. On dira qu'ils sont "voisins" s'ils peuvent communiquer. Cette relation est supposée symétrique.

Les robots ont des identités deux à deux distinctes, mais, au départ, chacun ne connaît que sa propre identité.

Chaque robot pourra :

- soit envoyer un message d'annonce, indiquant son identité, qui sera reçu par tous les autres robots à sa portée
- soit envoyer un message à un voisin dont il a appris et mémorisé l'existence.

Chaque robot enverra périodiquement un message d'annonce pour signaler sa présence. La réception d'un tel message permettra à un robot d'apprendre ou de confirmer la présence d'un voisin.

Chaque robot est muni d'une horloge réelle, qui lui permettra de dater la réception des messages (cf. ci-dessous).

2.2 Routage

Chaque robot mémorisera, indéfiniment, les noms des autres robots qu'il apprendra par réception de leurs messages d'annonce. Il associera à chacun de ces noms la date de dernière réception du message d'annonce correspondant au nom. Ceci lui permettra d'associer un statut à ce nom : ce sera un voisin si la date de réception est récente (différence avec l'heure actuelle inférieure ou égale à une constante `SilenceMax`).

Question 2.1 *On suppose pour le moment que les messages d'annonce ne contiennent que le nom du robot qui l'émet. Écrire la procédure qui permet à un robot de mettre à jour ses données à la réception d'un tel message, conformément aux exigences exprimées ci-dessus (préciser les structures de données utilisées).*

On aimerait à présent que chaque robot puisse connaître, outre ses voisins, les noms des autres robots ainsi qu'une route pour les atteindre (en utilisant un autre robot en relais lorsqu'ils ne sont pas à portée). Chaque robot va donc ajouter à ses messages d'annonce une liste de couples (robot.nom, robot.distance) de tous les robots qu'il connaît. Cette liste sera exploitée de la manière suivante lors de sa réception (principe de mise à jour des tables de routage vu au cours 3) :

```
Lors de la réception d'un message annonce(nom, listerobots)
Mémoriser ou confirmer nom comme voisin comme vu ci-dessus
Pour chaque element de listerobots faire
     si element.nom n'est pas parmi les noms mémorisés, l'ajouter
     mettre à jour la date de element.nom avec la date courante
     si la route vers element.nom est différente de nom
          alors si element.distance +1 < distance(element.nom)
               alors distance(element.nom) = element.distance +1
                     route(element.nom) = nom
               finsi
          finsi
fait
```

Par ailleurs, si l'existence d'un voisin n'est pas confirmée au bout du temps `SilenceMax`, les routes utilisant ce voisin comme relais seront supprimées.

Question 2.2 *Modifier et décrire précisément les structures de données des informations mémorisées afin de prendre en compte les données nécessaires, et écrire la procédure qui calcule la liste des robots connus dans le but de l'ajouter en paramètre au message d'annonce, pour exploitation comme indiqué ci-dessus.*

Question 2.3 *En utilisant ces structures de données, écrire la procédure - qui sera appelée périodiquement - permettant d'éliminer chaque route pour laquelle le voisin relais de cette route n'a pas donné d'information depuis un temps trop long.*

2.3 Diffusion

La maintenance, vue ci-dessus, de la table de routage permet à un robot de savoir par où envoyer un message destiné à un autre robot. Mais dans certains cas, un robot peut avoir besoin d'envoyer un message à tous les autres.

On pourrait pour réaliser cette opération :

- soit utiliser les tables de routage, et dupliquer le message en autant d'exemplaires qu'il y a de robots connus
- soit utiliser un algorithme de parcours standard tel que ceux vus au cours 1.

La première solution présente le défaut d'une duplication excessive du message, la seconde de ne pas utiliser du tout les informations mémorisés. D'où la :

Question 2.4 *Écrire une procédure de diffusion de message qui envoie ce message à tous les voisins, avec pour chacun d'eux la liste des autres robots auxquels il doit ré-expédier le message, liste calculée à partir de la table de routage mémorisée au moyen des structures de données précisées à la question 2.2 (préciser ce mode de calcul).*