Université de Caen **Année 2011-2012**
UFR de Sciences **1ère session**

EI61T Approfondissement en algorithmique
durée 2h00

Seules les notes de cours et TD, manuscrites ou imprimées sont autorisées. L'utilisation d'appareils électroniques est limitée à la fonction qui fournit l'heure.

Les deux parties correspondent aux parties du cours "algorithmique répartie" et "algorithmique de la cryptographie" et doivent être rendues sur des copies séparées.

Chaque candidat doit, au début de l'épreuve, porter son nom dans le coin de chaque copie qu'il cachera par collage après avoir été pointé.

1 Algorithmique répartie (à rédiger sur une copie séparée)

1.1 Introduction

Dans un système réparti, on a souvent besoin qu'un des sites du réseau joue un rôle particulier, car certains algorithmes doivent être lancés par un site et un seul.

Certaines circonstances peuvent nécessiter de désigner un nouveau site particulier. On fait appel pour ceci à un algorithme d'élection, qui doit être symétrique et qui peut être lancé par un nombre quelconque de sites du réseau.

On a vu en cours un tel algorithme, basé sur le lancement par un site du réseau au moins d'un parcours portant l'identité de son initiateur, seul le parcours ayant l'identité la plus forte étant acquitté, cet acquittement permettant au site concerné de savoir qu'il est l'élu.

Nous nous plaçons ici dans des hypothèses un peu différentes de celles des systèmes vus en cours, et nous allons construire un algorithme d'élection adapté à ce cas.

1.2 Hypothèses

On se place dans un système où tous les sites ont des identités deux à deux distinctes.

On suppose de plus, et c'est là la principale différence avec les systèmes étudiés en cours, que les messages sont transmis en un temps borné et que les sites connaissent la taille maximale du réseau, ce qui leur permet de connaître le temps maximum que doit mettre un message pour atteindre tout site du réseau, ainsi que le temps maximum que met la réponse à lui parvenir, lorsqu'il y a lieu.

On supposera que chaque site peut "armer un timer" T avec une valeur de durée, et que ce timer provoquera un événement interne au site (expiration du timer T) au bout du temps ainsi programmé.

1.3 Envoi d'informations aux sites du réseau

Question 1.1 *Écrivez un algorithme réparti permettant à tout site i du réseau d'envoyer un message :*

- *soit à tous les sites du réseau,*
- *soit aux sites du réseau qui ont une identité strictement supérieure à i,*
- *soit à un site désigné par son identité j.*

L'algorithme devra de plus positionner dans les différents sites une variable permettant ultérieurement à chaque site de pouvoir envoyer un message à l'initiateur i de cet algorithme.

Indication : on utilisera un parcours parallèle avec un message de diffusion comportant des paramètres supplémentaires permettant de préciser l'émetteur du message ainsi que les destinataires. Cet algorithme mémorisera sur le site i l'identité du site lui ayant transmis le message du site j pour la première fois, ceci dans la variable pere[j].

Question 1.2 *Décrivez comment le site i qui lance un tel message peut s'assurer du moment où son message a été transmis à tous les destinataires, et peut donc passer à l'opération suivante s'il y a lieu.*

Dans la suite, on supposera que chaque site peut appeler la fonction lui permettant d'envoyer un message à un ou des destinataires sélectionnés comme ci-dessus (on ne précisera plus les détails de l'algorithme qui le permet), et de positionner un timer lui permettant d'être sûr d'avoir reçu les réponses à ses questions.

1.4 Algorithme d'élection

Le réseau est supposé sujet à des modifications comportant entre autres le retrait, volontaire ou pour cause de panne, de certains sites y compris celui qui aurait un rôle de leader.

On suppose que, à un certain moment, un ou plusieurs sites s'aperçoivent qu'il y a un problème dans le réseau et qu'il convient de lancer un algorithme d'élection pour désigner l'un des sites comme leader qui "reprendra la main". Dans un premier temps, on supposera que le réseau est fiable pendant la durée d'exécution de cet algorithme au moins. En section 1.5 nous étudierons le cas contraire.

Cet algorithme d'élection, adapté aux caractéristiques du réseau, sera basé sur les principes suivants :

- Chaque site i qui démarre l'algorithme d'élection envoie aux sites ayant une identité plus forte que lui même (en espérant qu'il n'y en ait pas pour être l'élu) un message contenant le mot "Élection", et précisant son identité (ici i). De plus il arme un timer avec une valeur lui permettant d'attendre de recevoir toutes les réponses éventuelles.
- Chaque site j qui reçoit un tel message initié par le site i (et donc dont l'identité j est supérieure à i) envoie un acquittement négatif à i et, si ce n'était pas déjà fait, démarre pour lui-même l'algorithme d'élection.
- Si un site qui a lancé l'algorithme d'élection ne reçoit pas d'acquittement négatif dans le temps programmé, il se considère comme l'élu, et envoie à tous les sites du réseau un message le signalant (contenu : "Élu") et comportant son identité.
- Les sites recevant un message signalant qu'un site i est élu enregistrent cette information.

Question 1.3 *Mettre en forme cet algorithme, en écrivant les traitements à effectuer pour chaque événement possible, à savoir la réception des messages "Élection" ou "Élu", ou le débordement d'un timer précédemment armé.*

1.5 Problèmes éventuels

Cet algorithme fonctionne correctement si tous les messages sont effectivement transmis dans les délais prévus. On peut envisager par contre que certains messages soient perdus, messages d'acquittement ou de signalisation de l'élu notamment.

Question 1.4 *Pouvez-vous imaginer dans un de ces cas de perte de messages ce qu'il advient de l'algorithme ? Et dans ce cas, pensez-vous qu'il est impossible de se rattraper (à justifier) ou bien pouvez-vous imaginer une parade (à préciser) ?*

N.B. : Cette dernière question étant un peu "ouverte", toute bonne idée est susceptible d'y rapporter un peu de points, même si les détails des algorithmes implémentant ces idées ne sont pas précisés.

2 Algorithmique de la cryptographie (à rédiger sur une copie séparée)

Exercice

Dans tout l'exercice l'entier n désigne le produit de deux nombres premiers p et q gardés secrets.

1. Soit m un entier positif. A quelle condition un élément de $\mathbb{Z}/m\mathbb{Z}$ est inversible ?
2. On rappelle que l'exposant du chiffrement e est un élément inversible modulo $(p-1)(q-1)$ et l'exposant du déchiffrement d est l'inverse de e modulo $(p-1)(q-1)$:

$$ed = 1 \mod (p-1)(q-1).$$

La clé publique est alors la paire (n, e) et la clé privée correspondante est d. Soit m un message clair (converti en un entier inférieur à n) et c le message chiffré correspondant. On rappelle que $c = m^e \mod n$. On suppose ici qu'Alice a choisi les nombres premiers $p = 11$ et $q = 13$. Pour accélérer la fonction de chiffrement dans le cryptosystème RSA (par courtoisie envers les gens qui vous écrivent), chacun est tenté de choisir un petit exposant public. Ainsi Alice choisit le plus petit exposant public e possible. Donner la clé publique et la clé privée choisie par Alice en expliquant soigneusement et de manière détaillée l'algorithme utilisé pour calculer la clé privée et donner sa complexité.

3. Rappeler la méthode naïve et la méthode efficace pour calculer une exponentiation modulaire et donner les complexités des deux méthodes.
4. Bob veut chiffrer le message $m = 5$ à destination d'Alice. Calculer le chiffré correspondant.
5. On rappelle que $m = c^d \mod n$. Alice reçoit le chiffré $c = 16$. Déchiffrez le message c en utilisant une méthode efficace et en donnant la trace d'exécution de l'algorithme utilisé.

> Dans la suite, on étudie une technique basée sur le théorème des restes chinois pour accélérer par exemple le déchiffrement. Plus précisément, on étudie une méthode efficace pour calculer $m = c^d \quad [n]$.

On note :

$$\begin{array}{rcll} d_1 & = & d & [p-1] \\ d_2 & = & d & [q-1] \end{array}$$

6. On rappelle que $m_1 = m\ [p]$ et $m_2 = m\ [q]$ s'expriment respectivement en fonction de c, d_1 et p et en fonction de c, d_2 et q, de la manière suivante :
   On note :

$$\begin{array}{rcll} m_1 & = & c^{d_1} & [p] \\ m_2 & = & c^{d_2} & [q] \end{array}$$

Résoudre le système suivant où l'inconnu est m ($m < n$). et en déduire une méthode de déchiffrement pour RSA.

$$\left\{ \begin{array}{rcll} m & = & m_1 & [p] \\ m & = & m_2 & [q] \end{array} \right.$$

7. Appliquer cette méthode pour calculer le message m à partir du chiffré $c = 16$. En évaluant la complexité de cette méthode, comparer la méthode basée sur le théorème des reste chinois avec la méthode classique.