Université de Caen **Année 2009-2010**
UFR de Sciences **1ère session**

EI61T Approfondissement en algorithmique
durée 2h00

Les notes de cours et TD sont autorisées, ainsi qu'une calculatrice.
Les deux parties sont indépendantes, correspondent aux parties du cours "algorithmique pour la 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 Algorithmique répartie (à rédiger sur une copie séparée)

1.1 Introduction

En algorithmique répartie, la structure d'anneau peut rendre bien des services, notamment pour le problème d'accès à une ressource en exclusion mutuelle. Il suffit en effet de faire alors tourner un jeton sur l'anneau, dont la possession par un site autorise ce site à accéder à la ressource. Si on impose à chaque site de passer le jeton à son successeur après avoir utilisé, s'il le souhaitait, le privilège qu'il porte, on assure que chaque site pourra, s'il le désire, accéder régulièrement à cette ressource. Sous réserve que les lignes soient fiables et en particulier ne perdent pas le jeton, la sécurité est absolue.

Bien entendu, mettre en place une telle structure d'anneau, équivalente à un circuit hamiltonien du graphe dont les sommets sont les sites du réseau et les arêtes les lignes de communication, n'est pas toujours possible. On a vu en T.D. un algorithme permettant de créer un "anneau virtuel" sur un réseau quelconque, en utilisant un parcours séquentiel de ce réseau (fin du T.D. 1).

L'objectif de ce problème est de partir du résultat obtenu par cet algorithme, et d'essayer d'optimiser le circuit virtuel obtenu.

1.2 rappel du résultat obtenu et utilisation de l'anneau virtuel

L'algorithme vu en T.D. permettait à chaque site i :
- d'une part d'enregistrer dans une variable `successeur` l'identité du successeur du site i dans l'anneau,
- d'autre part de mémoriser dans des variables `R[t]`, t étant un voisin de i, l'identité du site auquel i devait envoyer le jeton après l'avoir reçu du site t, ceci que le site i ait été autorisé à utiliser le privilège porté par le jeton ou non.

Le message de jeton lui-même portait en paramètre l'identité du site destinataire du jeton. Ainsi, le site recevant ce message jeton savait s'il pouvait utiliser ce privilège, ou bien s'il devait simplement le faire suivre pour qu'il atteigne le destinataire désigné.

Question 1.1 *Rappeler ce que le site i, recevant le message Jeton(destinataire) transmis par un de ses voisins k, peut et doit faire pour assurer la propriété d'accès en exclusion mutuelle à la ressource gérée par ce jeton.*

1.3 Optimisation

L'algorithme vu en T.D. n'a pas la prétention de trouver le "meilleur" anneau virtuel sur le réseau. De plus, on peut envisager le cas où un ou des ajouts de lignes de communications dans le réseau permettrait d'envisager un circuit plus court (comportant moins de liaisons).

On suppose dans ce qui suit que chaque site i a mémorisé, en plus des variables indiquées en section 1.2, l'identité de son voisin qui lui transmet le jeton avec droit d'utilisation, ceci dans la variable `pred`. Ceci permet au site i de savoir que, lorsqu'il doit transmettre le jeton à son successeur après l'avoir utilisé, il doit le transmettre à `R[pred]`.

Question 1.2 *En supposant que le site i s'aperçoive que son successeur dans l'anneau virtuel est un de ses voisins, indiquez comment il peut :*

- *voir si ce fait était déjà connu de lui ou pas, et mémorisé dans ses variables de gestion de l'anneau virtuel*
- *si ce n'était pas le cas, modifier ses variables pour s'adapter à sa nouvelle situation*

Adapter ses variables à sa nouvelle situation n'est pas suffisant. Il faut aussi que le ou les autres sites concernés modifient leurs variables en conséquence.

Question 1.3 *Indiquez comment, dans le cas où le site s'aperçoit de cette nouvelle situation, il peut avertir les autres sites concernés afin qu'ils mettent leurs variables à jour pour profiter de cette liaison directe entre le site i et son successeur dans l'anneau. Indiquer précisément les messages à envoyer, à qui, et les actions à effectuer sur les variables des sites recevant ces messages.*

NB :

- on supposera qu'on traite une telle situation en un seul endroit à la fois dans tout le réseau.
- on pourra s'aider du cas de la figure 1, mais la question ci-dessus requiert une réponse valable dans tous les cas de figure.

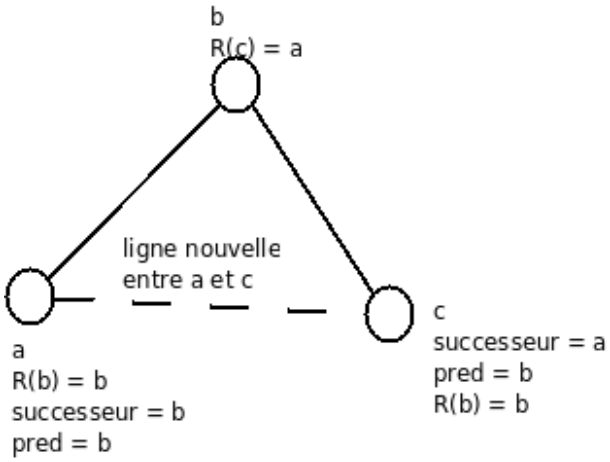

FIGURE 1 – Exemple de cas d'optimisation

2 Algorithmique pour la Cryptologie (à rédiger sur une copie séparée)

2.1 Introduction

Si le chiffrement RSA est le plus utilisé à l'heure actuelle, une autre technique est largement répandue : c'est le chiffrement Elgamal. Il fonctionne de la façon suivante :

Soit p un grand nombre premier (typiquement 1024 bits) de la forme $p = 2q + 1$, avec q lui aussi premier. On appelle les nombres premiers de cette forme des nombres premiers *forts*.

On considère un élément $g \in (\mathbb{Z}/p\mathbb{Z})^\star$ d'ordre q (c'est-à-dire que $g^q = 1 \pmod p$ et $g^i \neq 1 \pmod p$ pour tous $0 < i < q$).

- La clé secrète d'un utilisateur A est alors un entier $x_A \in [\![1, q-1]\!]$, et sa clé publique est $y_A = g^{x_A} \pmod p$.
- Le chiffrement d'un message $m \in (\mathbb{Z}/p\mathbb{Z})^\star$ se fait de la façon suivante :
  - un entier r est tiré aléatoirement dans $[\![1, q-1]\!]$,
  - le chiffré $c = (u, v)$ à destination d'Alice est composé des deux éléments suivants :

$$u = y_A^r \pmod p \quad \text{et} \quad v = m \cdot g^r \pmod p.$$

1. Quelle est la taille de q en bits si p est de taille n_p ?
2. Donnez un exemple de nombre premier fort plus grand que 15.
3. Donnez l'algorithme de déchiffrement.
4. Citez *tous* les algorithmes arithmétiques nécessaires pour mettre en œuvre un chiffrement Elgamal.
5. Montrez qu'un attaquant qui a accès à une machine qui déchiffre en boîte noire avec la clé secrète d'Alice peut arriver à retrouver le message $m^\star$ à partir de son chiffré $c^\star = (u^\star, v^\star)$ (destiné à Alice) en ne demandant pas le déchiffrement explicite de $c^\star$ à la machine. (Il peut demander n'importe quel couple (u, v) à la machine à condition que $(u, v) \neq (u^\star, v^\star)$, par contre, l'un des deux éléments peut être soit $u^\star$ ou $v^\star$).
6. Que fait l'algorithme suivant ? Quel est le nombre de multiplications qu'il fait ? Pourquoi pourrait-on vouloir cette propriété ?

Input: $x \in (\mathbb{Z}/p\mathbb{Z})^\star$ et $n = \sum_{i=0}^{\ell-1} n_i \times 2^i$

1. $x_1 \leftarrow x$
2. $x_2 \leftarrow x^2$
3. for i from $\ell - 2$ to 0 do
   - 3.1. if $n_i = 0$ then
     - 3.1.1. $x_1 \leftarrow x_1^2$
     - 3.1.2. $x_2 \leftarrow x_1 \times x_2$
   - 3.2. else
     - 3.1.1. $x_1 \leftarrow x_1 \times x_2$
     - 3.1.2. $x_2 \leftarrow x_2^2$
4. return x_1.

2.2 Algorithmes de calcul de logarithme discret

Pour que le chiffrement Elgamal soit sûr, le problème du logarithme discret doit être difficile. En d'autres termes, étant donné g et $y \in (\mathbb{Z}/p\mathbb{Z})^\star$, il doit être difficile de retrouver $x \in [\![1, q-1]\!]$ tel que $y = g^x \pmod p$.

1. Résolvez l'équation $6^x = 2 \pmod{23}$.
2. Donnez un algorithme trivial pour résoudre le logarithme discret et sa complexité.

Vous allez maintenant étudier deux algorithmes de calcul de logarithme discret et analyser leur complexité.

2.2.1 Pas de bébé, pas de géant

On cherche à résoudre l'équation $y = g^x \pmod p$ (1).
Soit $u = \lfloor \sqrt{p} \rfloor$, et soit la division euclidienne de x (inconnu) par u : $x = a \cdot u + b$.

3. Rappelez l'inégalité satisfaite par b.
4. Montrez que l'équation (1) se réécrit $y(g^{-u})^a = g^b \pmod p$.

Soit l'algorithme suivant :

Input: $y, g \in (\mathbb{Z}/p\mathbb{Z})^\star, p$

1. compute $\mathcal{B} = \{1, g, g^2, \ldots, g^{u-1}\}$ [baby steps]
2. $g' \leftarrow g^{-u} \pmod p$,
3. $t \leftarrow y$
4. for i from 0 to $p/u - 1$ do
   - 4.1. if $t \in \mathcal{B}$ return (? ?)
   - 4.2 else $t \leftarrow t \times g' \pmod p$ [giant step]

5. Que doit retourner l'algorithme (? ?) en 3.1 pour retrouver le logarithme discret de y ?
6. Quelle est sa complexité (en temps et en mémoire) ?

2.2.2 ρ de Pollard

On cherche toujours à résoudre l'équation (1).
Soit une suite $(a_i, b_i)_{i\in\mathbb{N}}$ d'entiers modulo $p-1$ et $(x_i)_{i\in\mathbb{N}}$ une suite d'entiers modulo p telle que $x_i = y^{a_i} g^{b_i} \pmod p$ avec $a_0 = b_0 = 0$ et $x_0 = 1$, définies ainsi :

$$(a_{i+1}, b_{i+1}) = \begin{cases} \Big(a_i + 1 \pmod{p-1}, b_i\Big) & \text{si} \quad 0 < x_i < \frac{1}{3}p \\ \Big(2a_i \pmod{p-1}, 2b_i \pmod{p-1}\Big) & \text{si} \quad \frac{1}{3}p < x_i < \frac{2}{3}p \\ \Big(a_i, b_i + 1 \pmod{p-1}\Big) & \text{si} \quad \frac{2}{3} < x_i < p. \end{cases}$$

1. Exprimez x_{i+1} en fonction de x_i.
2. En supposant que la suite $(x_i)_{i\in\mathbb{N}}$ se comporte de façon "aléatoire", vous pouvez appliquer le paradoxe des anniversaires, et espérer une collision "rapidement".
   Montrez qu'une collision permet de retrouver le logarithme discret de y et déduisez-en un algorithme et sa complexité (inspirez-vous de l'algorithme ρ pour la factorisation).

3 Corrigé de la partie 1

3.1 Utilisation de l'anneau virtuel

(réponse à la question 1.1)

Le site recevant le jeton :
- peut, s'il est le destinataire, l'utiliser pour accéder à la ressource, et doit ensuite prévoir son passage à son successeur
- dans tous les cas doit le faire suivre en se basant sur le routage mémorisé dans le tableau R.

Ceci peut s'écrire ainsi :

```
Le site i :
Sur réception de Jeton(destinataire) transmis par le voisin k
      si destinataire == i
            utiliser l'accès à la ressource si i le souhaite
            destinataire = successeur
      dans tous les cas, envoyer Jeton(destinataire) à R[k]
```

3.2 Optimisation

Question 1.2 :

Le site peut voir si le routage prenait déjà en compte le fait que son successeur est un voisin en comparant les variables `R[pred]` et `Successeur`. Si leurs contenus sont égaux, c'est que le site savait déjà que pour aller au successeur, le chemin était direct.

Si ce n'est pas le cas, alors le site devra prendre en compte cette nouvelle information, et notamment remplacer la valeur de `R[pred]` par le contenu de la variable `Successeur`.

Question 1.3 :

Mais ceci ne suffit pas. En effet si on se contente de la modification ci-dessus, le successeur de i, en recevant le jeton, ne saura pas à qui l'envoyer après son utilisation éventuelle car le tableau R n'aura pas l'information correspondante. De plus, le ou les sites figurant dans l'ancien chemin qui menait de i à son successeur (site b sur l'exemple) conserveront des valeurs devenues inutiles (R[c] dans l'exemple). Il faut donc envoyer un message spécial suivant cet ancien chemin, afin de "nettoyer" ces variables ou les mettre à jour.

On va donc envoyer un "jeton spécial", dont le trajet entre i et son successeur-nouveau voisin suivra l'ancien trajet du jeton, pour faire ces mises à jour. Ce jeton spécial portera en paramètre l'identité du successeur comem le jeton normal, mais en plus l'identité du site i qui lance l'opération afin de pouvoir mettre à jour la variable `pred` sur le site successeur.

```
Si le site i s'aperçoit que son successeur est désormais un voisin :
      Envoyer JetonSpecial(Successeur, i) à R[pred]
      R[pred] = Successeur // à faire après, sinon on perd l'ancien chemin
```

```
Sur le site j,
sur réception de JetonSpecial(destinataire, origine) venant du voisin k :
      si j != destinataire
             alors
                    Envoyer JetonSpecial(destinataire, origine) à R[k]
                    Effacer R[k] // ce chemin ne sert plus
             sinon // j est le successeur de origine, et le lien est direct
                    pred = origine
                    R[origine] = R[k] // conserver la suite du chemin du jeton
                    Effacer R[k] // l'ancienne provenance du jeton ne sert plus
```

3.3 Remarque complémentaire (hors sujet)

Le mieux est d'attendre le passage du jeton pour lancer le mécanisme d'optimisation de route. Ainsi, le jeton se transforme momentanément en jeton spécial, il continue de porter le privilège, et en plus, ceci assure que cette optimisation ne se produire qu'en un seul endroit à la fois, ce qui évitera d'éventuelles collisions dans les modifications de variables, au cas où plusieurs optimisations pourraient avoir lieu simultanément.