Université Joseph Fourier
DESS Génie Informatique - année 2001-2002

Schéma de corrigé pour l'examen 2001

Problème 1.

Question 1. L'avantage principal des algorithmes de chiffrement à clés secrètes est l'efficacité. C'est ce qui justifie leur utilisation. Les algorithmes à clés publiques sont utilisés pour la transmission des clés secrètes et pour l'authentification, qui manipulent des données de taille réduite.

Question 2.

a) Question de cours (voir cours)
b) Il suffit d'ajouter à chaque chèque un identificateur unique (construit par ex . à partir du code de l'agence émettrice et d'un numéro d'ordre, ainsi que d'une estampille (date et heure locals). L'agence peut alors vérifier qu'un chèque donné a déjà été débité.

Question 3.

a) Le dernier message est
$B \rightarrow A$: $I, \{RA, K\}_{KA}, \{I, A, B\}_K$
b) L'authentification repose sur l'hypothèse que *A* et *B* ne communiquent respectivement leur clé *KA* et *KB* à personne sauf à *S*. *A* est authentifié pour *S* car il connaît *KA*. Le serveur *S* est authentifié pour *B* car il est le seul à connaître *KB*. Donc *A* est authentifié pour *B* à travers *S*.
B est authentifié pour *A* car il connaît la clé *K*, qui a été engendrée par *S* (car elle est communiquée à *A* chiffrée par *KA*, que *S* est seul à connaître). Donc *K* n'a pas pu être fabriquée par un autre que *S*.
Le rôle de *I* (qui peut être une estampille avec la date) est l'identification unique qui évite le rejeu de messages anciens. Le rôle de *RA* et *RB* est de servir à l'authentification de *S* pour *A* et *B*, car *RA* et *RB* sont chiffrés par *KA* que *S* est seul à connaître.

Problème 2.

Question 1.

a) Le service de gestion de noms sert au client (plus précisément au talon client) à connaître l'adresse réseau (par. ex. l'adresse IP et le numéro de porte) qu'il faut appeler pour exécuter la procédure. Ces informations ont été préalablement enregistrées par le talon serveur. Le service de gestion de noms peut résider sur le site du serveur ou sur un site indépendant. L'adresse réseau et le numéro de porte de ce service doivent être connus à l'avance (de la même façon que ceux du serveur DNS).
b) La modification demandée a un double intérêt. 1) Pour la tolérance aux fautes, en assurant que le service sera rendu même si *N* serveurs sont défaillants (si on en prévoit *N*+1). 2) Pour l'efficacité. En effet, si on a plusieurs serveurs, on peut choisir le moins chargé (si on peut connaître la charge) ou le plus voisin, pour réduire le temps d'attente (latence).

Plusieurs solutions sont possibles. Par exemple, on peut utiliser pour les serveurs de noms un schéma (serveur primaire, serveurs de secours). On a donc une liste fixée de couples (adresse IP, numéro de porte), connue de tous les clients et serveurs, le premier de la liste étant le serveur primaire, les suivants les serveurs de secours, dans l'ordre de remplacement. Dans ce cas, l'interface ne change pas, mais chaque primitive est appelée à l'intérieur d'une boucle comportant un délai de garde. Si le délai est écoulé sans réponse, on appelle le serveur suivant sur la liste. Si au contraire on utilise le schéma de la duplication active, on n'a rien à changer. On suppose dans les deux cas que les serveurs appliquent un protocole interne (voir cours) pour rester cohérents entre eux.

Un atre schéma non développé ici consiste à appeler le service à travers une procédure qui permet de déterminer le serveur le plus efficace.

NB. Il suffisait de donner une seule solution, les diverses possibilités sont données à titre indicatif.

Question 2.

a) L'intérêt de la modification proposée est d'une part l'indépendance entre service logique et gestion physique, d'autre part la facilité d'évolution et l'adaptabilité du service (indépendance mutuelle des différentes modifications), enfin la commodité pour l'utilisateur (le changement de support d'un fichier est invisible à l'utilisateur).

b) Une interface possible du service de noms est la suivante.
id = nouveau_id(*params*) : créer un nouvel identificateur interne. Les paramètres (facultatifs) peuvent comporter des préférences relatives au fichier (taille initiale, volume préféré).
associer (*f*, *id*) : associe l'identificateur interne *id* et le nom de fichier *f*. On peut aussi proposer une procédure qui combine *nouveau_id* et *associer*.
id = rechercher(*f*) : trouver l'identificateur interne *id* associé au nom de fichier *f* (ou renvoyer *nil* si pas trouvé).
supprimer (*f*) : supprime le nom de fichier *f* et libère l'identificateur *id* associé (utilisé lors de la destruction d'un fichier).
La liaison entre les service de noms et la gestion du stockage physique se fait uniquement par l'intermédiaire de l'identificateur interne, ce qui réalise l'indépendance de ces deux fonctions.

c) Il suffit d'avoir sur chaque serveur une table d'implantation des volumes qui sont actuellement stockés sur ce support, et sur chaque site utilisateur une table de correspondance entre numéro de volume et nom de serveur Ainsi, pour tout fichier, on peut déterminer son volume à l'aide de son id et ensuite le serveur sur lequel il se trouve grâce à la table de correspondance. C'est la solution adoptée dans AFS.

Pour déplacer un volume, il faut bloquer l'accès à tous les fichiers de ce volume, puis attendre que les accès en cours soient terminés. Ensuite, on transfère physiquement le volume (par recopie ou par montage d'un support physique), et on met à jour la table d'implantation sur le serveur. Enfin, on diffuse à tous les sites la modification à apporter à la table de correspondance (une association (volume, serveur) a changé). Puis on libère l'accès aux fichiers du volume. Ces

opérations sont réalisées par un programme du système, sous contrôle de l'administrateur.

d) Pour gérer des volumes dupliqués, il faut changer la structure de la table de correspondance, pour permettre d'avoir des entrées de la forme (n° de volume, liste de serveurs) au lieu de (n° de volume, serveur). L'accès à un volume nécessite un algorithme de choix entre les différents serveurs qui gèrent le volume (par exemple essayer le premier de la liste ; si pas disponible, prendre le suivant, etc ; ou des algorithmes plus raffinés prenant en compte la charge ou la proximité, ce qui suppose de tenir ces informations à jour).