IPIPIP : Alice, Bob et Eve

Sophie.Demassey@mines-nantes.fr

1 Problématique

La cryptologie est une discipline scientifique qui étudie l'écriture, la transmission et l'analyse des messages secrets. Dans ce projet, nous laisserons de côté les questions de transmission, pour nous focaliser sur l'écriture/lecture de messages codés (cryptographie) et, dans une moindre mesure, sur l'attaque de codes (cryptoanalyse).

La problématique se schématise alors comme suit : Alice doit envoyer un message confidentiel à Bob. Eve peut éventuellement intercepter le message durant sa transmission. Alice doit s'assurer qu'Eve ne pourra pas prendre connaissance du contenu du message. Le rôle d'Alice est donc de transformer le **message en clair** en un **message crypté** avant de transmettre ce message crypté. Elle s'assure que Bob connaisse bien la transformation inverse (**déchiffrage**), au contraire d'Eve. L'objectif d'Eve va être de deviner cette transformation inverse (on dit **analyser**), afin de prendre connaissance du message en clair, à partir seulement du message crypté et, éventuellement, connaissant le système de cryptage utilisé.

2 Solution

Il s'agit de développer un logiciel, utilisable par chacune de ces trois personnes, Alice, Bob et Eve. Une fois connectée, après avoir précisé son nom en paramètre du programme, la personne disposera des fonctionnalités suivantes :

- Alice pourra : choisir un système de cryptage, générer des clefs de cryptage secrètes et publiques, communiquer avec Bob pour l'échange des clefs, et crypter n'importe quel message textuel au moyen des clefs ;
- Bob pourra : générer des clefs de cryptage secrètes et publiques, communiquer avec Alice pour l'échange des clefs, et décrypter le message codé par Alice au moyen des clefs ;
- Eve pourra : tenter de décrypter le message codé par Alice sans la clef, mais connaissant éventuellement le système de cryptage utilisé par Alice. Eve pourra également chercher la valeur de la clef secrète d'Alice si elle parvient à intercepter à la fois un message en clair et son message crypté (attaque à clair connu).

Toute communication (secrète ou publique) se fera par l'intermédiaire de fichiers texte : *message-clair.txt*, *message-crypto.txt*, *clef-session-alice-pour-bob.txt*, *clef-secrete-bob.txt*, etc.

L'interface se fera dans le terminal, en ligne de commande ; un exemple de session est présenté dans l'encadré de la page suivante.

3 Systèmes de cryptographie

De nombreux systèmes de cryptographie existent. Beaucoup sont recensés et expliqués sur le site suivant : http://www.apprendre-en-ligne.net/crypto/menu/index.html Vous êtes libres d'implémenter tout système cryptographique de votre choix en plus des systèmes ci-dessous.

```
$> java crypto Alice
Bonjour Alice !
Choisissez votre systeme de cryptage:
> Vigenere
Systeme de cryptage Vigenere... OK
Precisez la longueur de la clef (defaut: entier aleatoire entre 2 et 10):
>
generation de la clef de session... OK
enregistrement du systeme et de la clef: clef-session-pour-bob.txt... OK
Entrez le message a coder ou le nom du fichier a coder:
> message-clair.txt
lecture du message en clair... OK
cryptage du message... OK
enregistrement du message crypte: message-crypte.txt... OK
Aurevoir Alice !
$>
$> java crypto Bob
Bonjour Bob !
lecture du systeme et de la clef: clef-session-pour-bob.txt... OK
lecture du message crypte: message-crypte.txt... OK
Souhaitez-vous afficher ou enregistrer le message decrypte ?
> enregistrer message-decrypte.txt
enregistrement du message decrypte: message-decrypte.txt... OK
Aurevoir Bob !
$> $> java crypto Eve
Bonjour Eve !
lecture du message crypte: message-crypte.txt... OK
Listez les systemes de cryptage a tester (defaut: tous):
> Vigenere
Sorry: la cryptanalyse de Vigenere n'est pas implementee.
Voulez-vous essayer d'autres systemes ?
> oui
analyse rot13: message-analyse-rot13.txt... OK
analyse cesar (frequentielle/francais): recherche clef probable... NON
Sorry: la cryptanalyse du message a echouee.
Aurevoir Eve !
$>
```

3.1 Rot13

L'algorithme Rot13 consiste à remplacer chaque lettre du message (ex : 'A') par la lettre se situant 13 places après, de manière cyclique (par rotation), dans l'alphabet (ex : 'N'). Une majuscule est remplacée par une majuscule, et une minuscule par une minuscule. Tout autre caractère qu'une lettre non-accentuée reste inchangé. Ce système ne nécessite pas de clef de session. Les algorithmes de cryptage, de décryptage et de cryptanalyse de ce système sont identiques. Exemple :

message-clair.txt	clef-session-pour-bob.txt	message-crypte.txt
En 1999, Rot13 était utilisé par Netscape pour encoder des mots de passe !	Rot13	Ra 1999, Ebg13 égnvg hgvyvfé cne Argfpncr cbhe rapbqre qrf zbgf qr cnffr !

3.2 Chiffre Rot47

L'algorithme Rot47 est une variante de Rot13 portant également sur les chiffres et certains symboles, en assimilant chaque caractère à la valeur décimale de son code ASCII. Chaque caractère, autre que

l'espace, dont le code ASCII est compris entre 33 et 126, est transformé en un caractère de code compris entre 33 et 126, au moyen d'une rotation de 47 (ex : 'A' de code 65 est transformé en 'p' de code 112, et 'a' de code 97 en '2' de code 50). Les accents sur les lettres sont aussi préalablement supprimés (ex : 'é' est donc traduit comme 'e' (code 101) par '6' (code 54)). Tout autre caractère reste inchangé.

3.3 Chiffre de César

Le chiffre de César est une généralisation de Rot13 où la valeur de la clef de la rotation est n'importe quel entier compris entre 1 et 25. Cette valeur doit être préalablement déterminée et partagée par Alice et Bob : on parle d'un algorithme à clef privée. L'algorithme de décryptage est évident. Il existe deux types d'algorithmes de cryptanalyse. Le premier *brute-force* consiste à tester toutes les clefs de rotation possible. Eve doit alors considérer le message décrypté qui a le plus de sens (pour des textes courts, il peut y en avoir plusieurs). Le second algorithme repose sur l'*analyse fréquentielle* d'apparition des lettres, en les comparant à la table de probabilité de la langue du message. Par exemple, on pourra trouver les probabilités d'apparition des lettres dans un texte français sur la page suivante : http://fr.wikipedia.org/wiki/Fréquence_d'apparition_des_lettres_en_français

`message-clair.txt`	`clef-session-pour-bob.txt`	`message-crypte.txt`
Attaquez Asterix	Cesar 3	Dwwdtxhc Dvwhula

3.4 Chiffres de Playfair et de Beale

Spécifiez et implémentez les algorithmes de cryptage et décryptage de ces deux systèmes.

4 L'implémentation

L'implémentation se fera en trois étape, parallèlement à votre apprentissage de la programmation. Dès maintenant, vous pouvez commencer à écrire et implémenter les algorithmes de cryptage, de décryptage et, éventuellement de cryptanalyse, pour les systèmes énumérés ci-dessus, et d'autres de votre choix, en prenant soin de factoriser au maximum les fonctions communes au différents systèmes.

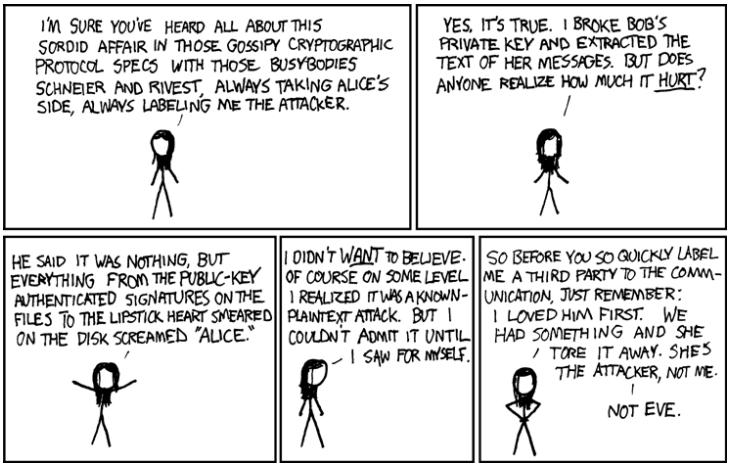
Dans un second temps (d'ici la fin du module de programmation), vous pourrez aborder :

- la conception et l'implémentation de l'interface et des lectures/écritures dans un fichier
- l'implémentation de systèmes basés sur le chiffrement des caractères en bits

Enfin, vous appliquerez les concepts d'héritage, vus dans le module de programmation par objets, pour concevoir l'architecture complète de votre programme.

5 La difficulté

Algorithmique : *** (bonus :*****)
Programmation : **
Modélisation du problème : *
Modélisation objet : ***



www.xkcd.com