Partie Langage C

Compétence C2

Contexte

Avec la clé de déchiffrement, il est possible de récupérer le contenu des fichiers en inversant le processus. Les analystes de votre équipe ont identifié la manière avec laquelle le chiffrement est construit, en voici le détail.

Un fichier chiffré subit les transformations suivantes :

1. Encodage en base64
2. Chiffrement par le chiffre de Vigenère avec une clé en base64, excepté pour les caractères de bourrage (« padding »), côté données et côté clé
3. Décodage depuis base64

Note : il est nécessaire de se renseigner sur le fonctionnement du chiffre de Vigenère avant de poursuivre.

Objectifs

Chiffrement

Construire le programme `cipher` qui chiffre un fichier encodé en base64, dont le nom est passé en argument à l'appel du programme. Le fichier chiffré en sortie remplace le fichier en clair en entrée. La clé de chiffrement en base64 est fournie en argument.

Exemple d'utilisation

```
./cipher macledechiffrement monfichier
```

Note : un programme déjà rencontré en Système d'Exploitation permet d'encoder un fichier en base64.

Déchiffrement

À partir du programme précédent, construire le programme `decipher` qui déchiffre un fichier encodé en base64, dont le nom est passé en argument à l'appel du programme. Le fichier en clair en sortie remplace le fichier chiffré en entrée. La clé de déchiffrement en base64 est fournie en argument.

Exemple d’utilisation

```
./decipher macledededéchiffrement monfichier
```

Note : un autre programme déjà rencontré permet de décoder un fichier en base64. La clé de chiffrement correspond aussi à la clé de déchiffrement.

Détermination de la clé de déchiffrement

En utilisant un fichier en clair et sa version chiffrée, construire le programme `findkey` qui détermine la clé de chiffrement utilisée. La clé est affichée sur la sortie standard, aucun autre affichage n’est réalisé sur cette sortie. Sur la sortie erreur, la taille de la clé est affichée.

Exemple d’utilisation

```
./findkey monfichierenclair monfichierchiffre
```

[BONUS] Simplification de l’assemblage des programmes

Plutôt que d’utiliser directement la commande `gcc`, il est proposé de construire un Makefile qui permet de construire chacun des outils ou les 3 en appelant la commande `make`.

[BONUS] Bibliothèque statique de fonctions

À l’avenir, il est souhaité d’intégrer les fonctionnalités de déchiffrement et de détermination de la clé de déchiffrement dans un plus gros programme. Il est demandé de préparer une bibliothèque statique prête à l’emploi pour ces deux fonctionnalités. Son assemblage doit être ajouté au Makefile.

Historique des modifications du sujet

Version 1.0.1 – 202511140935

Correction des coquilles.

Version 1.0.0 – 202511041900

Version initiale du sujet.