PIF1006 – Mathématiques pour informaticiens II

Session Automne 2021

TRAVAIL #3

Indications principales	
Date de présentation de l'énoncé	25 novembre 2022
Date de remise du travail	22 décembre 2022 à 23h59
Politique sur remises tardives	-25% par jour de retard
Éléments et format de remise	Rapport et fichiers complet de la solution ou du projet remis sous forme électronique sur le portail de cours dans un seul fichier compressé (.zip)
Pondération	10%
Nb d'étudiants par équipe	2 à 5 personnes

INTRODUCTION

Nous avons vu en classe quelques bases en cryptographie et quelques méthodes courantes de chiffrement et déchiffrement par blocs avec une clé secrète, comme les méthodes *Electronic Code Book (ECB)* et *Cipher Block Chaining (CBC)*.

Nous avons également vu que pour le chiffrement des blocs à proprement dit, nous pouvons utiliser un chiffrement par substitution ou par permutation (ou un produit des deux). Il est notamment important vu l'aspect symétrique du chiffrement et déchiffrement que ces clés ou fonctions de chiffrement respectent le principe de la bijection.

Il est très intéressant de coder de telles méthodes et de telles fonctions de chiffrement afin de pouvoir apprécier le processus de cryptographie et les algorithmes qu'il implique. C'est ce que vous serez d'ailleurs invités à faire dans le cadre de ce travail.

DESCRIPTION

Il vous est demandé de coder une classe permettant de chiffrer un message de texte en utilisant la méthode ***Cipher Block Chaining*** (*CBC*), et comme fonction de chiffrement des blocs un **chiffrement par transposition** et ce, en utilisant exactement la stratégie décrite dans les notes de cours (voir exemple « *ce cours de mathématiques est très intéressant* »). Nous allons cependant introduire une variante afin de simplifier quelque peu le processus.

Pour cela, vous devez créer une classe statique « `Chiffrement` » possédant deux méthodes :

- `String Chiffrer(String message, String cle)`
- `String Dechiffrer(String message, String cle)`

Pour le chiffrement :

- Dans un premier temps, le message au complet devra subir un chiffrement par transposition en utilisant une clé de transposition au choix de l'utilisateur, composée d'une chaîne de caractères composée d'une série de nombres séparés par un caractère d'espacement (p. ex. « 1 4 6 5 3 2 »). La quantité de nombres dans la série donnera par le fait même le nombre de colonnes de transposition.

  Considérez que l'utilisateur entrera une série adéquate.

- Dans un deuxième temps, vous devez utiliser la méthode de chiffrement par bloc CBC sur le message précédemment transposé en considérant que 1 caractère (1 octet) = 1 bloc. La clé (fonction) de chiffrement ayant déjà été appliquée préalablement sur le message en entier, vous n'avez qu'à effectuer l'opération **XOR** entre la valeur du bloc clair avec la valeur du bloc chiffré précédent (ou le vecteur d'initialisation pour le premier bloc à chiffrer).

  *Cela implique que vous devrez **transformer** la **chaîne de caractères** comportant le message en un **tableau d'octets** (byte[]) qui constituera l'ensemble des blocs clairs. Le **tableau d'octets chiffrés** que vous aurez produit suite au chiffrement par CBC **devra être reformé** en une **chaîne de caractère** que vous retournerez à l'appelant et qui formera ainsi le message chiffré.*

Pour le déchiffrement :

- Vous devez d'abord décomposer le message chiffré en un tableau d'octets, sur lequel vous appliquez l'algorithme de déchiffrement avec la méthode *CBC* sur chacun des blocs d'octet. Cela aura alors pour effet de reconstituer le message transposé.

- Puis, vous devez appliquer la transposition inverse en respectant la clé de transposition fournie par l'utilisateur et retourner le message résultant. Avec la même clé de

transposition utilisée pour le chiffrement (et un peu de « chance » !), le message retourné devrait être celui d'origine.

Précisions supplémentaires:

Le **vecteur d'initialisation** (VI), doit être déterminé par l'utilisateur et pour le déchiffrement et pour le chiffrement.

La **clé de transposition** sera la clé transmise lors du chiffrement et **la même clé devra être fournie lors du déchiffrement afin de pouvoir bien ordonner les colonnes et reconstituer le message d'origine ligne par ligne**.

CONSEIL : Ne négligez pas d'ajouter tout commentaire pertinent à votre code. Ils ne seront pas évalués comme tel, mais ils pourront tout de même aider lors de la correction à comprendre ce que vous vouliez faire en cas d'erreur.

Bien évidemment, vous aurez besoin d'une classe « ***`Main`*** » ou l'équivalent qui permet d'interagir avec votre classe de chiffrement afin de tester vos algorithmes et vérifier leur bon fonctionnement en affichant à la console ou dans des contrôles utilisateurs les résultats de l'utilisation des méthodes de chiffrement et de déchiffrement.

N.B. Vous êtes tenu de faire en sorte que l'utilisateur ait à entrer les messages à l'exécution, soit à la console, soit dans une interface utilisateur : on doit voir au minimum l'affichage du message original, crypté et décrypté, afin de pouvoir bien observer que le tout se déroule tel que prévu.

Enfin, vous pourriez avoir besoin des éléments qui suivent pour mener à bien votre travail :

- L'opérateur *XOR* en C# est « `^` » et doit être utilisé sur des variables de types `int` ou des bits (valeurs booléennes) (vous aurez donc probablement des opérations de *casting* à faire);
- Pour passer d'un `String` vers un `byte[]`, une des façons de faire consiste à instancier un objet de classe `ASCIIEncoding` ou `Encoding` et appeler la méthode `GetBytes()`.
- Pour transformer un *String* en un tableau de jetons (sous-chaînes), vous pouvez utiliser la méthode `Split()` et spécifier un séparateur (*p. ex.* un caractère d'espacement).

EXIGENCES ET INSTRUCTIONS SUPPLÉMENTAIRES

Langage de programmation

Au niveau du langage de programmation, vous devez utiliser soit l'une ou l'autre des options suivantes, **sans aucune exception** :

- C#/VB/C++ avec Microsoft Visual Studio .NET Core
- Java avec NetBeans.
- Autres options à valider avec l'enseignant

Éléments à remettre

Vous devrez remettre l'ensemble des extrants exigés qui seront énumérés ci-dessous dans un seul fichier compressé (en format .zip) dans la section de dépôt des travaux sur le portail du cours **avant la date et heure limite.**

Dans le cas où vous remettiez le travail en retard, vous ne pourrez alors le déposer sur le portail et vous devrez me le retourner via courriel. Une pénalité de 25% par jour de retard, à compter de l'heure de remise, serait alors appliquée.

Les éléments à remettre sont les suivants :

- Tous les **fichiers relatifs au projet** de l'application :
    - Si vous travaillez avec Visual Studio .NET, vous devez remettre le répertoire contenant la solution, le ou les projets associés, les fichiers contenant le code source, les exécutables, etc.
    - Si vous travaillez avec Java NetBeans, vous devez remettre le projet NetBeans, ainsi que les fichiers .java et .class, puis le .jar généré.

  Dans tous les cas, assurez-vous que le projet soit prêt à être exécuté directement à l'intérieur de la plateforme de développement, c'est-à-dire sans erreur de compilation et sans configuration spéciale non explicitement donnée. Aucun débogage ne sera fait lors de la correction.

- Un **rapport Word** ou PDF dans lequel vous devez au minimum indiquer les éléments suivants :
    - Page de présentation;
    - Problèmes et difficultés rencontrés (s'il y a lieu);
    - Instructions spéciales d'exécution du programme (s'il y a lieu);

- **Guide d'utilisation avec instructions d'utilisation et impressions d'écran** :
    - Vous devez, à l'intérieur de ce guide, montrez 3 exemples d'utilisation du chiffrement :
        - Un exemple qui fonctionne avec une clé de transposition quelconque;
        - Un second exemple qui fonctionne avec une autre clé de transposition avec une quantité différente de nombres;
        - Un troisième exemple pour lequel la clé tentée pour le déchiffrement est différente de celle utilisée pour le chiffrement;
        - Un quatrième exemple pour lequel le VI utilisé pour déchiffrer est différent que celui pour chiffrer.

GRILLE D'ÉVALUATION

Voici les critères d'évaluation et la pondération associée à chacun d'entre eux :

Sujet d'évaluation	***Pts (/10)***
Rapport complet/guide utilisateur (incl. les résultats pour les cas demandés)	1
Classe *Main* (utilisation de la classe Chiffrement et affichages)	1

Classe *Chiffrement :*
- Chiffrement : 4 pts
  - Transposition : 2 pts
  - CBC : 2pts
- Déchiffrement : 4 pts
  - Transposition : 2 pts
  - CBC : 2pts
	8