

Projet : C++



Axel ACUNA
Olivier KLAK
Année Universitaire 2019-2020
Université Paris-Dauphine

Introduction :

Dans le cadre de la réalisation du Master MIAE Informatique pour la Finance, nous avons été amené à suivre le module de découverte C++. Pour valider ce dernier, la réalisation d'un projet a dû être menée. Le projet consistera en la réalisation et l'implémentation d'un algorithme de chiffrement pour chiffrer et déchiffrer un fichier. Ce projet permettra de mettre en avant les différentes notions vu en cours comme l'utilisation de pointeurs, de l'héritage, de liste et de vector pour les plus importants.

Au travers de ce projets, les classes de bases seront implémentées en premier lieu qui contiendront les différentes séquences utilisées lors des phases de l'algorithme de chiffrement.

Première Partie :

Dans l'optique de pouvoir tester et valider notre code au fur et à mesure de l'avancement du projet, nous avons choisi de rompre la consigne initiale avec le principe de l'encapsulation. Néanmoins, il est toujours possible de revenir à la consigne en mettant le tout dans un private.

Cette première partie établit les bases pour l'ensemble du projet.

Les fonctions principales de la Séquence et Séquence D sont :

- ❖ Décalage : permet de décaler les bits d'une séquence avec un paramètre
- ❖ Right and Left : permet d'obtenir la séquenceD en 2

sous séquences

- ❖ Opérateur >> et << : permet l'écriture et la lecture dans différents flux.
- ❖ Write et read : permet de transformer et d'interpréter une saisie ascii en valeur binaire et de
- ❖ Affichage : permet d'afficher le contenu de 2 types de séquences

Deuxième Partie :

Pour cette partie, nous avons investi pas mal de temps notamment pour de la recherche pour arriver au bout de la fonction Permutation, on ne savait pas ce qu'était un patron de classe fonction. Par ailleurs, cette même classe reprenait de base plusieurs choses des classes permutation de la première partie mais cette fois, les différents paramètres permettaient de préciser le début et la fin de la permutation de la séquence.

Pour la classe KeyGen, nous allons recevoir une séquence d'un certains nombre de bits (64), que nous allons permuter pour en retirer une séquence de 56 bits (tableau permtuchoice_pc1.tableS.3).

```

#include "KeyGen.h"
#include <Permutation.h>
KeyGen::KeyGen(SequenceD<64>& sequence) {
    Permutation<64,56> perm;
    this->round = 0;
    this->sequenceD = perm(sequence,PC1);
}

SequenceD<48> KeyGen::next() {
    //get bits rotated by Round number
    int step = LS[this->round];
    this->round++;
    //applying left shift of the sequence.
    this->sequenceD.decalage(step);
    Permutation<56,48> perm;

    return perm(sequenceD,PC2);
}

```

Concernant le next, nous avons pris le nombre de rotation en fonction du tour et on applique une permutation pour passer d'une séquence de 56 à 48 bits, à la fin la méthode renvoie les différentes clés séquences.

Notre exécution des clefs fonctionne bien, la capture ci-après en fait état :

```

19 int main() {
20     /**
21      * Génération de 16 clés à partir d'une clé random
22      */
23     SequenceD<64> sequenceD;
24
25     KeyGen keyGen( &sequenceD);
26
27     for(int i = 0; i<16;i++){
28         cout<<"Cle " << i << "\t" << keyGen.next().to_string() <<endl;
29     }
30 }

```

Run: MAIN_RUN x MAIN_RUN x

C:\Users\axel\CLionProjects\projet-cpp\cmake-build-debug\src\main\MAIN_RUN.exe

```

Cle 0  010011111011010011110101 001110111000011010111111
Cle 1  100010011110010100111101 101000100111011101101110
Cle 2  110001010000111110111011 001111001011101110100110
Cle 3  111101111011101010100001 111101000100110011110011
Cle 4  100110111011111011000010 010011111010101001011011
Cle 5  011110000111011011011110 10110111111010101011000
Cle 6  011101001101010101001100 001010011001011101100110
Cle 7  010000101100100101110111 110111001110110010100110
Cle 8  110111101110011001110001 000100011111100101101101
Cle 9  110011111101011100001110 111000101011110010110000
Cle 10 011010101001001111001111 111010010010111100111111

```

Conclusion :

Pour ce projet, nous avons été amené à manipuler et utiliser des fonctions de hauts niveaux, à plusieurs reprises nous avons dû faire appel à des recherches sur différents sites spécialisés sur les fonctions et les exemples qui leur étaient associés. Quand nous nous sommes inspirés de fonctions, nous avons indiqué la source originale en commentaire.

Nous avons aussi procédé à plusieurs arbitrages concernant notamment : les tests via le framework de GOOGLE, afin de créer des exécutables. Nous avons décidé séparer les fichiers includes et fichier de code.cpp afin qu'à la lecture notre code soit aéré. Dans nos fichiers.h, nous retrouvons des fonctions templates.

Il est recommandé d'utiliser un éditeur comme Clion ou C++clipse, car ils interprètent les makefiles. Par ailleurs, dans certains cas, il nous est venu à l'esprit d'utiliser certaines bibliothèques du langage C pour pouvoir utiliser des fonctions, à ce sujet, nous ne trouvons pas de fonctions équivalentes et simples à mettre en place d'où ce choix.

Ce projet a été assez long et fastidieux, malgré le fait qu'il ne soit pas entièrement fini, nous en avons gardé une bonne expérience, le niveau demandé et le travail en équipe pour ce projet ont été très appréciables.