Rapport

Bourgeois Noé

22 mai 2021

Contents

1 Introduction

Le projet de cette année avait pour objectif de nous familiariser avec les concepts appris durant le cours en comparant deux langages : Python et C++. Il consistait en l'implémentation d'un système de communication chiffré.

Le chiffrement est un moyen de transformer une suite de caractères, en une autre suite, de telle sorte que cette dernière soit codée à l'aide d'une information secrète. Sans cette information, il serait impossible de déchiffrer la suite de caractère.

Ici, l'échange de messages s'effectue entre deux programmes, l'un en Python et l'autre en C++. Ces deux programmes communiquent via un fichier, fourni en paramètre. Le programme C++ envoie les messages sur le fichier entré par l'utilisateur, tandis que le programme Python lit les messages arrivant sur le fichier.

Les messages échangés via le fichier sont chiffrés et dechiffrés en utilisant l'une ou l'autre des fonctions présentées ci-dessous.

2 Fonctions

2.1 Caesar

Le Code de César , également appelé le chiffrement par décalage était utilisé par Jules César.

Le texte chiffré s'obtient en décalant, vers la gauche ou vers la droite, l'alphabet tout en gardant son ordre. Le chiffrement peut être représenté par l'alignement de deux alphabets : l'alphabet chiffré est équivalent à l'alphabet initial décalé d'un certain nombre de positions vers la gauche ou vers la droite.

2.2 Enigma

Enigma est une machine qui était utilisée par l'armée allemande pendant de la Seconde Guerre mondiale.

Lorsqu'un caractère est fourni à la machine, chaque composant qui la constitue est chargé de remplacer le caractère reçu en entrée par un autre en sortie. Il existe deux types de composants utilisés par la machine : Les rotors : Un rotor est un composant chargé de faire correspondre chaque lettre fournie en entrée par une autre lettre en sortie en se basant sur une permutation fixe. Le réflecteur : Le réflecteur se situe au bout de la chaîne, et est une dernière permutation effectuée

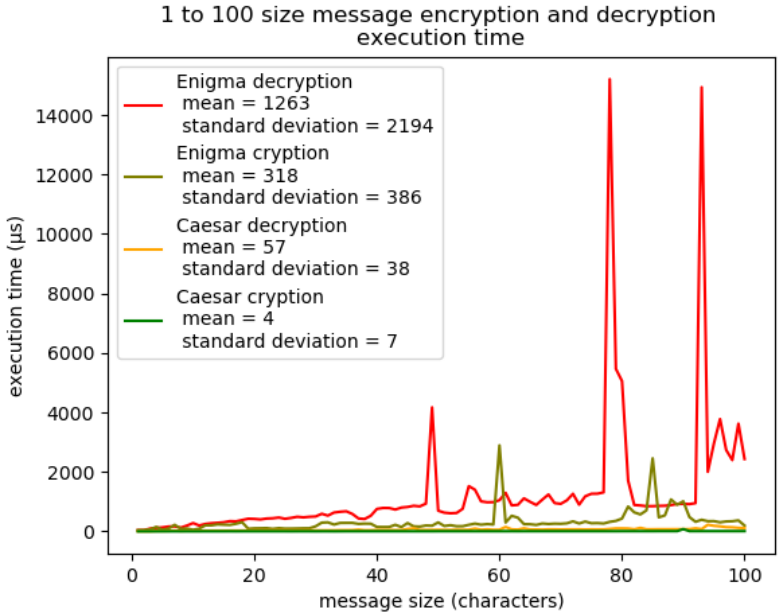
La machine effectue trois étapes : 1. Le signal entrant traverse les permutations de chaque rotor de droite à gauche. 2. La lettre entrante du rotor de gauche est envoyée au réflecteur qui retourne une nouvelle lettre et refléte le résultat sur le rotor de gauche. 3. La dernière étape correspond au signal renvoyé par le réflecteur aux rotors en lisant la table des permutations de manière inverse. Finalement, le signal sortant du rotor de droite indique la lettre chiffrée.

3 Performances et encapsulation

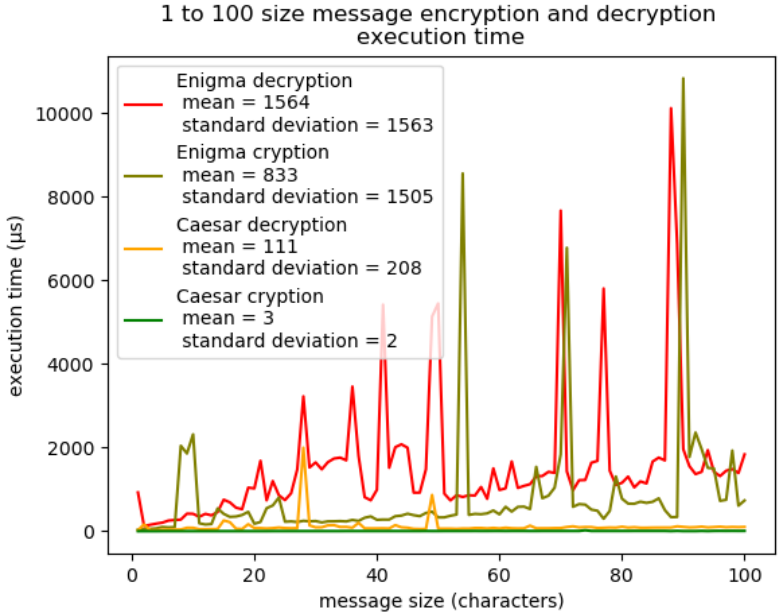
Nous avons donc implémenté les programmes puis évalué les temps d'executions des 2 fonctions dans les 2 programmes. Nous avons ensuite encapsulé les fonctions dans des classes et réévalué les performances.

4 Résultats

before encapsulation:



after encapsulation:



5 Discussion

Nous pouvons tout d'abord constater que le temps d'exécution d'enigma est plus élevé que celui de caesar à cause du nombre d'opérations pour simuler chaque rotor et le réflecteur.

Le temps moyen maximum est celui d'"enigma" appelé par "recv" car le message doit être codé en bytes pour être transmis de recv à enigma puis décodé à son retour.

Après encapsulation, les temps moyens ont tous augmenté, même plus que doublé pour enigma appelée par send, sauf caesar appelée par send qui était déjà la plus rapide (4µs).

L'écart-type a lui aussi augmenté dans tous les cas sauf pour enigma appelée par recv.

Il est intéressant de constater que les pics des mêmes fonctions ne correspondent pas aux mêmes mots mais bien aux mêmes quantités de mêmes lettres dans les mots à crypter ou décrypter.

6 Conclusion

Enigma pourtant codée en c++ est plus lente, peu importe depuis quelle autre fonction elle est appelée, encapsulation ou non, car le nombre d'opérations est plus élevé.