Rapport projet transverse

CryptImage

Hermann HUANG

Victorin HUET

Luka KUPATADZE

Romain PASQUIER

Yoan SAPIENZA

2021-2022

Sommaire :

I. Introduction

A. Présentation générale du projet

CryptImage est un projet d'application qui permet à l'utilisateur de crypter ses images de manière fiable et sécurisée afin de les envoyer en toute sérénité à ses contacts sans craindre que ces informations soient vendues à des tiers.

Les réseaux sociaux sont incontournables de nos jours, mais quelle garantie avons-nous que les géantes multinationales derrières ces applications ne vendent pas nos informations à notre insu ? En réalité cette vente des informations personnelles constitue une majeure partie des revenues de ces entreprises. Si certaines informations collectées peuvent être anodines, d'autres sont bien plus sensibles comme des photos personnelles. Or à l'heure actuelle, il n'existe pas encore d'application qui permet de crypter facilement et rapidement des images et permettre leur envoie en toute confidentialité. Il s'agit évidemment d'un enjeu de société qui touche à la protection de la vie privée de chacun et du droit au secret, à l'heure où les informations de tous sont collectées et utilisées à des fins commerciales, économiques ou plus traitreusement pour manipuler les opinions.

B. Théorie

Le projet a pour domaine la cryptographie et plus particulièrement la cryptographie des images. Le principe général consiste à travailler sur le position et l'information contenue dans les pixels d'une image : échange de position et permutation, changement des valeurs RGB des pixels ou encore utilisation de portes logiques sur les bits des valeurs RGB. La notion de clé reste commune à toutes ces propositions. La clé se doit d'être générée aléatoirement et est utilisée à la fois pour le cryptage et le décryptage : il s'agit d'un chiffrement symétrique. L'utilisateur A qui a cryptée une image doit alors envoyer sa clé à l'utilisateur B qui souhaite la décrypter.

C. Méthodes choisies

La première méthode consiste a travaillé sur une clef de 16 bits pour en extraire des coefficients. Ces coefficients serviront à travailler sur la position des pixels grâce à des fonctions affines et des modulos en nombre premier. Cette méthode permet de lisser notre image pour éviter de reconnaitre n'importe quelle autre méthode de cryptage.

La deuxième méthode choisie est celle dite de la matrice Householder. L'idée principale est d'utiliser une matrice comme clé. Cela peut paraître contreproductif sachant que le produit matriciel n'est pas un algorithme particulièrement rapide mais la matrice Householder est une matrice orthogonale symétrique c'est-à-dire que l'inverse de la matrice est elle-même : c'est un gain de temps considérable lorsque l'on fera le décryptage. Cette matrice sera utilisée pour changer la valeur RGB des pixels donc l'image cryptée aura des couleurs totalement aléatoires.

D. Objectifs et cahier des charges

Les objectifs fixés sont les suivants :

- Réaliser une application de cryptage d'image puis son décryptage en Java
- Un maximum de format d'image peut être pris en compte dont les principales (PNG, JPEG, BMP, etc…)
- Le temps de cryptage et de décryptage doit être raisonnable.
- La qualité de l'image doit être sauvegardée

Après avoir retracé la réalisation et le déroulé du projet, nous allons procéder à un rapide bilan de ce que nous avons appris, ce que nous avons pu améliorer et les erreurs que nous avons commises. Enfin nous allons conclure ce rapport ; en annexe se trouvera les sources et le glossaire.

II. Réalisation et déroulé du projet

Le thème du projet s'est très vite porté sur le domaine de la cryptographie, cependant nous avons hésité entre crypter des images que l'on envoie ou des enregistrements audios. Nous avons choisi les images car nous voyons mieux comment procéder et de plus une image peut transmettre plus d'information : l'image elle-même ou un texte photographié par exemple. Enfin on a choisi de coder l'application en Java.

En premier lieu nous avons d'abord effectué des recherches très générales sur la cryptographie et appris les concepts et notions de base.

Nous sommes tombés sur un papier traitant d'une méthode pour cryptage d'image appelée méthode Householder. C'était une méthode intéressante complémentaire à la première car elle travaille sur la couleur des pixels. La partie visant à crypter une image a d'abord été réalisée et testée ; puis une deuxième partie pour décrypter a été implémentée.

III. Bilan

La méthode Householder s'est avérée particulièrement ardue à implémenter car il a fallu optimiser les calculs matriciels qui peuvent prendre énormément de temps. Il y également beaucoup de matrices à calculer à la fois dans le cryptage et puis le décryptage. Tester et vérifier chacune des méthodes s'est révélé extrêmement chronophage et lent. Il y a également eu des soucis au niveau des valeurs flottantes non exactes pour le décryptage.

Par ailleurs, la combinaison des deux méthodes a eu pour contrainte d'avoir une image carrée de taille n égale à un nombre premier. Il a donc fallu créer un algorithme de redimensionnement qui ajoute des bandes noires. Bien sûr il a fallu inclure l'information de redimensionnement dans la clé.

Bien que nous ayons rapidement pris de l'avance, il a été difficile au début de s'organiser efficacement.

IV. Impact RSE

La création de cette application n'a pas eu d'impact environnemental et son utilisation n'en aura pas non plus. Cette application devrait rendre un grand service à la société en permettant à tous de pouvoir facilement protéger sa vie privée et ses photo personnelles. Il est en effet capital de ses rendre compte de nos jours que nous sommes quotidiennement espionné à notre insu et des personnes peuvent chercher à nous nuire. Il s'agit donc de responsabiliser chacun sur la protection de sa vie personnelle et une telle application peut aider à sensibiliser le plus de gens possible.

V. Conclusion

Le projet a été de manière générale mené à bien. Une application en Java est fonctionnelle et permet le cryptage et décryptage de manière rapide. Cependant nous avons encore besoin d'améliorer les temps de traitement sur les très grandes résolutions au-dessus de la full HD (1920*1080) qui prennent également beaucoup de place mémoire. Les différents formats d'image PNG, JPEG, BMP sont compatibles et il n'y a aucune perte d'information ou de qualité au cours du processus. En somme le projet a été assez formateur sur le travail en groupe, une condition indispensable à la réussite d'un tel projet. Par ailleurs nous avons graduellement amélioré notre niveau de codage. Enfin, nous projetons d'implémenter cette application sur d'autres appareils comme les téléphones Android avec Android Studio. A terme, on aimerait que cette

application puisse directement s'intégrer avec les réseaux sociaux et ainsi faciliter et accélérer l'utilisation de l'application.

VI. Annexe : sources et bibliographies :

Méthode Householder :
https://dspace.univ-ouargla.dz/jspui/bitstream/123456789/21951/1/M%C3%A9thode%20de%20cryptage%20d%E2%80%99image%20bas%C3%A9e%20sur%20la.pdf

Produit de convolution

Opérations sur une image :
https://qastack.fr/codegolf/35005/rearrange-pixels-in-image-so-it-cant-be-recognized-and-then-get-it-back

Cours crypto :
https://lipn.univ-paris13.fr/~poinsot/SEC/Chapitre_7_Printable.pdf

Générer nombre aléatoire :
http://www.codeurjava.com/2015/03/generer-nombres-aleatoires.html

Algorithme de CW :
https://fr.wikipedia.org/wiki/Algorithme_de_Coppersmith-Winograd
https://www.geeksforgeeks.org/implementing-coppersmith-winograd-algorithm-in-java/

Autres pistes de produit matriciel :
https://javawithus.com/fr/multiplication-de-deux-matrices-en-java/
https://www.it-swarm-fr.com/fr/java/performances-des-java/958281085/

Cours Android Studio :
https://openclassrooms.com/fr/courses/4517166-developpez-votre-premiere-application-android

Optimisation de l'algorithme de multiplication des matrices :
https://martin-thoma.com/matrix-multiplication-python-java-cpp/#java_1

http://www.codeurjava.com/2015/09/difference-entre-arraylist-et-linkedlist-en-java.html
https://www.jmdoudoux.fr/java/dej/chap-math.htm#math-8

https://stackoverflow.com/questions/41029236/how-do-i-embed-a-java-program-into-my-website

https://github.com/kiprobinson/BigFraction
https://commons.apache.org/proper/commons-math/javadocs/api-3.3/org/apache/commons/math3/fraction/BigFraction.html

Technique de cryptographie :

http://deptinfo.unice.fr/twiki/pub/Linfo/PlanningDesSoutenances20032004/blanc-degeorges.pdf

Livre : Histoire du chiffrement et de ses méthodes : Synthèse chronologique du chiffrement a travers les âges

Méthode de cryptage d'image basée sur la permutation :

https://dspace.univouargla.dz/jspui/bitstream/123456789/21951/1/M%C3%A9thode%20de%20cryptage%20d%E2%80%99image%20bas%C3%A9e%20sur%20la.pdf

Images cryptées avec un masque jetable :

https://www.apprendre-en-ligne.net/crypto/images/index.html

Transfert sécurisé par combinaison de CRYPTAGE et de TATOUAGE D'IMAGES :

https://www.lirmm.fr/~wpuech/enseignement/old/DEA_info/05_cours/03_AS_contenu_securise_puech.ppt