Tables des matières

1 Fondement du projet

1.1 Introduction

La cryptologie est dans sa définition formelle la plus récente la science qui enclôt d'une part la cryptographie fondée sur la protection de messages confidentiels ainsi que le maintien de leurs authenticités à l'aide d'une clé et d'une autre la cryptanalyse, qui elle consiste à retrouver le message en clair à partir d'un texte chiffré, et ce sans clé. Cette discipline a énormément servi depuis l'antiquité et l'intérêt que suscite la protection de données a poussé les cryptographes à travers, le temps a repoussé à chaque fois les limites imposées par leurs prédécesseurs. En effet, la cryptologie a permis par exemple à l'Allemagne nazie pendant la Seconde Guerre mondiale de communiquer des messages sensibles à ses alliées à l'aide de la célèbre machine de chiffrement et de déchiffrement *Enigma* attaquée et mise à nu en 1936 par le cryptographe et mathématicien britannique *Alan Turing*. Aujourd'hui, la cryptologie ne relève plus seulement du prysme mathématique, mais également de l'informatique et est utilisée à des fins différentes. Les ambassades de tout pays par exemple utilisent exclusivement le chiffrement pour faire transiter les informations dites secrètes et les entreprises commerciales quant à elles en viennent à entretenir des Cryptolecte pour protéger leurs données des éventuels piratages informatiques provenant souvent de la concurrence.

1.2 But du projet

1.2.1 Contexte du projet :

Dans le cadre de notre projet final d'étude en licence informatique à l'université de *Versailles Saint Quentin En Yvelines*, la réalisation d'un outil automatique d'aide au décryptage nous a été confié par le professeur responsable du module *projet* durant le sixième semestre de la licence.

1.2.2 Objectif du projet :

Le but du projet est de créer un produit de type logiciel *dekstop* servant à faciliter le chiffrement, le déchiffrement et la cryptanalyse de textes en français, axé sur le chiffre de Vigenère et le chiffrement par substitution. Le logiciel offrira une interface interactive, moderne et simplifiée à la fois permettant d'insérer des textes, de choisir l'opération à appliquer sur ces derniers et d'afficher ou non le processus lié à l'opération choisie en temps réel.

1.3 Personnes et organismes impliqués dans les enjeux du projet

1.3.1 Maître d'ouvrage :

Le logiciel est réalisée pour des informaticiens dotés de connaissances maigres ou insuffisantes en cryptologie souhaitant se frotter à la cryptographie et à la cryptanalyse dans une démarche pédagogique.

1.3.2 Utilisateurs du produit :

Tout utilisateur désirant chiffrer, déchiffrer ou cryptanalyser des textes en français.

2 Contraintes sur le Projet

2.1 Contraintes non négociables

2.1.1 Contraintes sur la conception :

- L'outil doit être équipé d'une interface graphique ergonomique permettant de suivre le déroulement des différentes opérations effectuées sur les textes dans un temps réel.
- L'outil doit fournir des fonctionnalités divisées en modules indépendants les uns des autres.
- L'outil doit être entièrement réalisé en utilisant seulement des bibliothèques gratuites.
- L'outil doit permettre l'exportation de fichiers sous différents formats.

2.1.2 Environnement de fonctionnement :

L'outil doit être exploitable sur les trois systèmes d'exploitation : Windows, MacOS et Linux.

2.1.3 Contrainte de temps :

L'outil doit être remis avant le 21 Mai 2019.

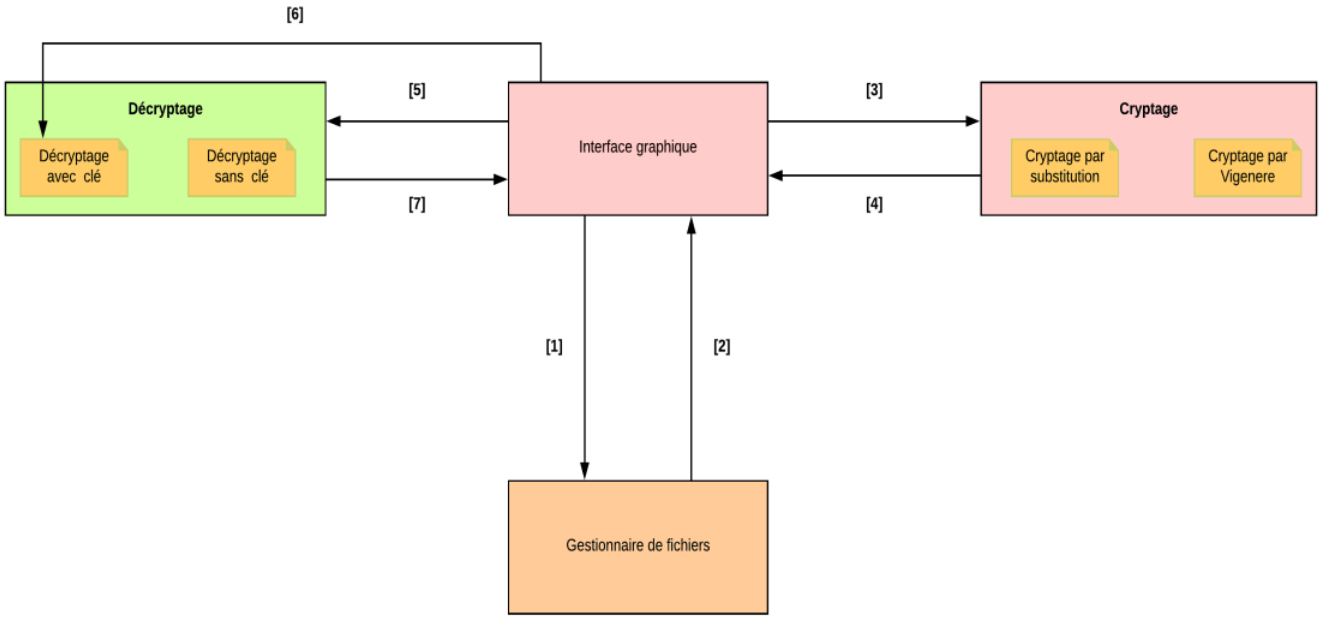
2.2 Glossaire et conventions de dénomination

- Clé : Chaîne de caractères servant de secret pour chiffrer et déchiffrer un texte sensible.
- Crypter : Action de chiffrer un texte en utilisant une clé afin de protéger son contenu.
- Décrypter : Action de traduire un texte chiffré en utilisant la clé appropriée.
- Cryptanalyse : Action de trouver la clé de chiffrement d'un texte chiffré.
- Fonctionnalité : Désigne une action qui rend un service particulier en informatique.
- Module : Un module représente un ensemble de routines qui ont une fonction précise.

3 Exigences fonctionnelles

3.1 Organigramme et fonctionnalités des modules

3.1.1 Organigramme et données échangées :



[1] Texte à sauvegarder dans un fichier + chemin du fichier de destination.
[2] Texte clair ou chiffré chargé depuis un fichier.
[3] Texte clair à chiffré + la méthode de chiffrement (Vigenère/Substitution) + la clé de chiffrement.
[4] Texte chiffré en utilisant la méthode et la clé choisies.
[5] Texte chiffré à déchiffrer + la méthode utilisée lors du chiffrement.
[6] la clé utilisée lors du chiffrement du texte.
[7] Texte résultant du processus de déchiffrement.

3.1.2 Fonctionnalités des modules :

L'outil comprendra principalement quatre modules :

1. interface graphique :

Description: Module permettant la gestion de l'interaction entre l'utilisateur et l'outil, et ce de manière conviviale et intuitive.

Liste des Fonctionnalités:

- choisir l'action à effectuer (Cryptage/Décryptage).
- choisir le type de cryptage/décryptage (Substitution/Vigenère).
- introduire un texte manuellement.
- choisir un fichier source pour importer le texte contenu dans le fichier.
- choisir la clé avant de lancer le cryptage.
- choisir de visualiser le déroulement du cryptage.
- choisir de visualiser le déroulement du décryptage.
- visualiser le texte en sortie.
- choisir d'exporter le texte résultant vers un fichier.

2. Cryptage

Description: Module chargé d'assurer le cryptage d'un texte donné en entrée, et ce en utilisant l'une des deux méthodes de cryptage Substitution/Vigenère.

2.1 Cryptage par substitution :

Description : Ce sous-module est chargé d'assurer le cryptage d'un texte donné en entrée en utilisant la méthode de cryptage par substitution et en se basant sur la clé choisit par l'utilisateur.

Liste des fonctionnalités :

- Crypter un texte par substitution.
- Visualiser le déroulement du cryptage étape par étape.

2.2 Cryptage de Vigenère :

Description : Ce sous-module est chargé d'assurer le cryptage d'un texte donné en entrée en utilisant la méthode de cryptage de Vigenère et en se basant sur la clé choisit par l'utilisateur.

Liste des fonctionnalités :

- Crypter un texte par la méthode de blaise Vigenère.
- Visualiser le déroulement du cryptage étape par étape.

3. Décryptage

Description : Ce module est chargé d'assurer le décryptage d'un texte donné crypté auparavant soit par la méthode de substitution soit par la méthode de Vigenère ter un texte par la méthode de blaise Vigenère.étape. est chargé d'assurer le décryptage d'un texte donné crypté auparavant soit par la méthode de substitution soit par la méthode de Vigenère.

3.1 Décryptage avec clé :

Description : Ce sous-module assurera le décryptage d'un texte donné en entrée, et ce, en utilisant deux paramètres qui sont le type du cryptage ainsi que la clé utilisée lors du cryptage. Remarque : ce sous-module devra assurer que le texte résultant est identique à 100% au texte original.

Liste des fonctionnalités :

- Retrouver le texte original à partir d'un texte crypté par substitution et ce en utilisant une clé de décryptage.
- Retrouver le texte original à partir d'un texte crypté par la méthode de Vigenère et ce en utilisant une clé de décryptage.
- Visualiser le déroulement du décryptage étape par étape.

3.2 Décryptage sans clé :

Description : Ce sous-module devra assurer le décryptage d'un texte donné en entrée, en utilisant un seul paramètre qui est le type du cryptage, ce sous-module devra aider à retrouver une grande partie du texte original à la différence du précédent qui avait pour rôle de retrouver 100% du texte original.

Liste des fonctionnalités :

- Retrouver une grande partie du texte original à partir d'un texte crypté par substitution.
- Retrouver une grande partie du texte original à partir d'un texte crypté par la méthode de Vigenère.
- Visualiser le déroulement du décryptage étape par étape.

4. Gestionnaire de fichiers

Description : Module permettant la gestion du texte en entrée/sortie, d'importer un texte contenu dans un fichier et de sauvegarder un texte résultant d'un des deux processus cryptage/décryptage dans un fichier.

Liste des fonctionnalités :

- Charger un texte depuis un fichier pdf/word/txt vers l'interface.
- Sauvegarder un texte affiché sur l'interface dans un fichier pdf/word/txt.

3.2 Algorithmes utilisés

Chiffrement par substitution monoalphabétique :
Principe : c'est un algorithme qui prend en entrée une clé de taille 26 en plus du texte clair et renvoie en sortie un texte crypté. cet algorithme remplace chaque occurrence de lettre par une autre lettre a partir de la clé.
Compléxité : O(n).

Déchiffrement par substitution monoalphabétique :
Principe : c'est un algorithme qui prend en entrée une clé de taille 26 en plus du texte crypté et renvoie en sortie un texte décrypté. cet algorithme remplace chaque occurrence de lettre par la lettre avec laquelle elle a été remplacé en cryptant le message, en d'autre terme on utilisera la clé dans le sens inverse.
Compléxité: O(n).

Chiffrement de vigenère :
Principe : Après la réception du texte et de la clé de chiffrement, cet algorithme découpe le texte en blocs de sorte à ce que chaque bloc ait la meme taille que la clé, additionne ensuite le rang de chaque lettre du texte (sachant que le rang appartient a Z26) avec le rang de la lettre de la clé qui lui correspond le tout modulo 26.
Compléxité : O(n).

Déchiffrement de vigenère :
Principe : Ce dernier Prend en entrée le texte crypté et la clé de chiffrement et renvoie un texte clair en sortie, au début il fait correspondre chaque bloc du texte crypté a la clé et en soustrayant le rang de la lettre de la clé (telle que le rang Z26) du rang de la lettre du texte chiffré le tout modulo 26. on obtient le texte déchiffré en sortie
Compléxité : O(n).

Algorithme d'analyse des fréquences :
Principe : Cet algorithme prend en entrée un texte chiffré, et produit une structure de données sous forme d'arbre, contenant les lettres du texte, les digrammes et les trigrammes avec leurs fréquences et positions dans le texte .
Compléxité : O(n).

Cryptanalyse de la substitution monoalphabétique :
Principe : Cet algorithme prend en entrée un texte crypté préalablement par substitution monoalphabétique et nous fournit en sortie un texte décrypté. Il commence par calculer les fréquences des polygrammes du texte, pour ensuite remplacer chaque occurence de lettre par celle qui convient le mieux, en comparant sa frequence dans ce texte et les fréquence des lettres de la langue, notamment dans le cas de collision entre les lettres, on passe a une comparaison de fréquences par digrammes. si le problème de collision perciste on passe au trigrammes.
Compléxité : O(n^3).

Algorithme de Kasiski :
Principe : l'objectif de cette méthode est de Permettre la détermination de la longueur de la clé. Cet algorithme extrait les distances qui séparent chaque groupe de polygrammes, il calcule leurs diviseurs et les PGCD s'il le faut, et finit par retourner la valeur la plus répétée comme étant la taille de la clé.
Compléxité : O(n).

Indice de coïncidence :
Principe : Cet algorithme découpe le texte en blocs valant la taille clé trouvée, il calcule l'indice de coïncidence de chaque groupe, si les valeurs sont approximativement égales à 0,077 il retourne un Vrai, si non un Faux qui aboutit à un nouveau choix de la longueur de la clé.
Compléxité : O($\frac{n}{m}$) pour un texte de n caractère et une clé de taille m.

Algorithme de cryptanalyse de Vigenère :
Principe : Avec la possession de la longueur de la clé, cet algorithme fait une attaque statistique mono-alphabétique sur les groupes précédemment repartitionnés, et ensuite il les fusionne pour avoir un texte clair et compréhensible.
Compléxité : O($\frac{n}{m}$) pour un texte de n caractère et une clé de taille m.

4 Exigences non fonctionnelles

4.1 Ergonomie et convivialité du produit

L'outil devra offrir une interface graphique interactive, moderne et facile d'accès dès la première prise en main.

4.2 Performance et fiabilité

L'application devra être sensiblement performante afin d'éxécuter toutes les tâches sollicitées par les utilisateurs de façon rapide et optimale; ainsi les différentes manipulations du logiciel devront être fluides et toute interaction devra obligatoirement être instantané et efficace.

4.3 Maintenabilité et scalabilité

Le code source de notre outil devra être lisible et compréhensible afin d'assurer un état évolutif et extensible par rapport aux besoins du marché. En outre l'ajout ou la modification de fonctionnalités seront facilement implémentable.

5 Autres aspects du projet

5.1 Estimation des coûts et répartition des tâches

Modules & sous modules	Estimation des coût	Répartition des tâches
Cryptage	300 lignes	Said & Faycal
Décryptage avec clé	300 lignes	Yasmine & Ismail
Cryptanalyse	600 lignes	L'ensemble du groupe
Gestionnaire de fichiers	200 lignes	Amine & Akram
Interface graphique	1200 lignes	Ala Eddine & Maher

5.2 Possibilités d'améliorations

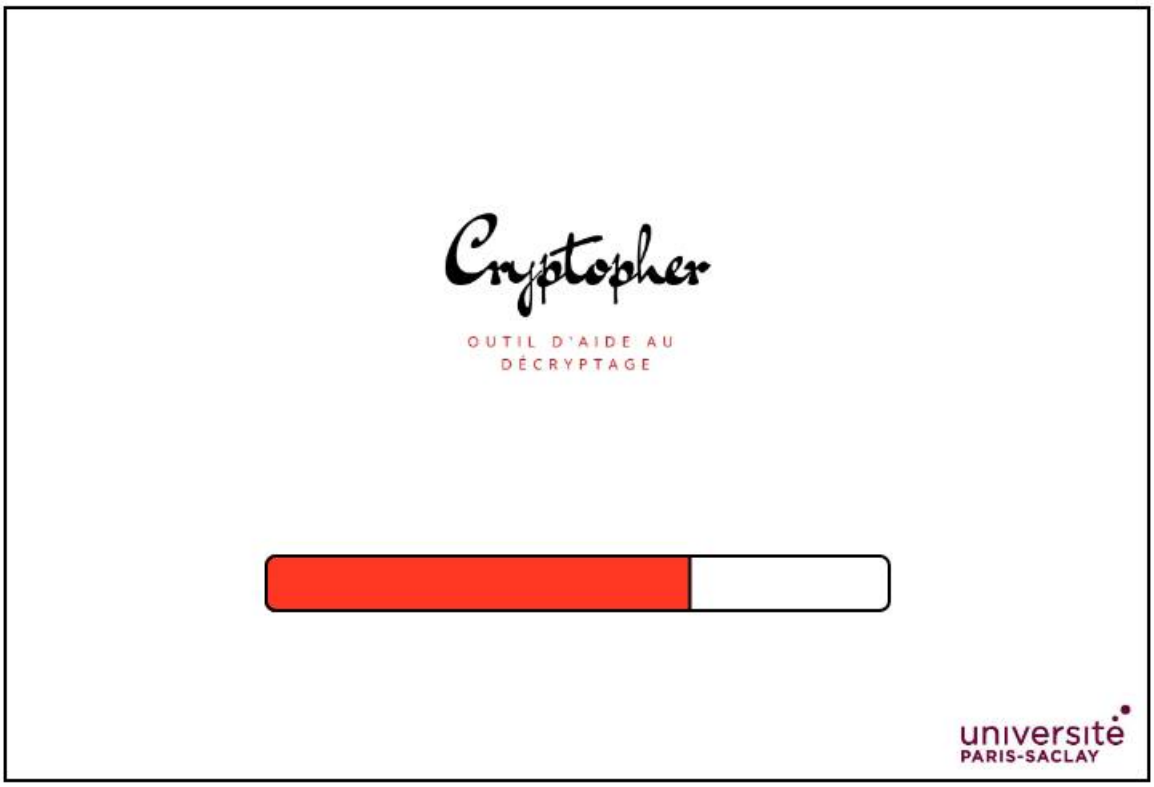
L'application ne permet de manipuler que deux types de chiffrement/déchiffrement, on peut alors proposer dans un premier temps la possibilité de manier d'autres type d'Algorithmes tel qu'*Enigma*, *Hill* ou encore *César* afin d'enrichir les fonctionnalités de l'outil et de le rendre plus interssant.Dans un second temps permettre à l'utilisateur de travailler sur des textes écrits en d'autres langues que le français lui donnerait un côté universel.

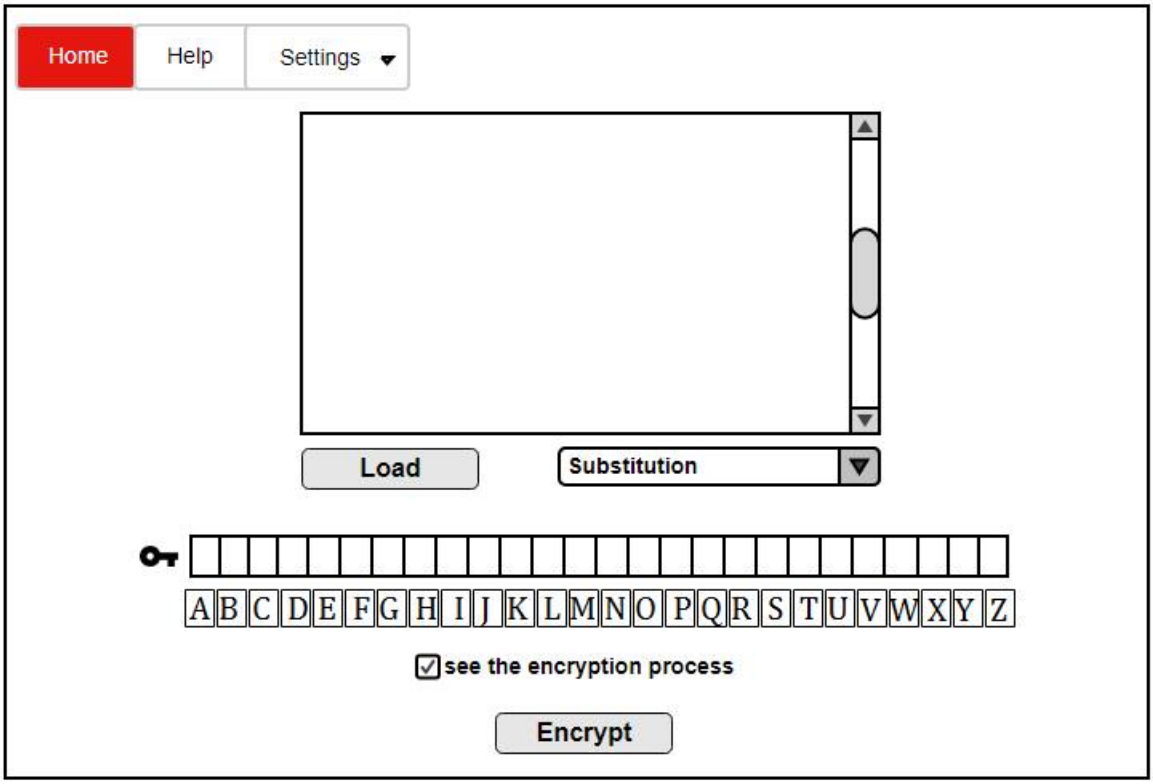
6 Conclusion

La rédaction du cahier des charges nous a permis de comprendre méthodiquement les spécificités de notre outil et d'apprivoiser les notions primordiales à sa conception malgré les divergences intellectuelles de tout un chacun. Cette démarche nous a amené à choisir avec conviction ce qui nous semble être le langage de programmation le plus adapté à la réalisation de ce projet, le C++. Le multithreading et les routines mathématiques qu'impose cet outil nous ont poussé à opter pour un langage procédural. Néanmoins, la complexité de certains modules et notamment la cryptanalyse nous a obligé à chercher une conception algorithmique claire et organisée. En effet, étant donné que la répartion des taches s'est faite par modules, nous créerons du code modulaire permettant d'isoler chaque module et d'en créer séparément de nouveaux qui viendront s'ajouter au fur et à mesure de l'implémentation. Nous avons donc commencé à réfléchir objet et à entrevoir les aspect de la POO, en remarquant qu'une clé de chiffrement est à titre d'exemple une donnée sensible dont la structure ne doit en aucun cas être modifié ce qui relèverait de l'encapsulation. Ayant par conséquent besoin d'un langage qui soit procédural et orienté objet à la fois, nous avons pensé au C++ car en plus de son aspect hybride, il permet d'utiliser un nombre très important de bibliothèques gratuites ainsi que le somptueux Framework Qt afin de réaliser une fenêtre graphique en bonne et due forme.

7 Annexes

7.1 Apparence présumée de l'application





7.2 Références

- *Cryptographie Théorie et pratique, Dougals Stinson (1995).*
- *Algorithmique et cryptographie, Robine Guy (1991).*
- *La cryptologie : L'art des codes secrets, Philippe Guillot (2013).*
- *Fous de codes (secrets), Mark Frary (2017).*
- *Google scholar An Interactive Cryptanalysis Algorithm for the Vigenere Cipher.*
- *Initiation à la cryptographie : théorie et pratique https://www.di.ens.fr/ ferradi/cours.pdf*
- *Long métrage "Imitation Game" Netflix.*