École Supérieure en Sciences et Technologies de l'Informatique et du Numérique

Rapport Projet N°1 Sécurité Informatique: Machine Enigma

Höhere Schule für Computer- und digitale Wissenschaft und Technologie

Projektbericht Nr. 1 Computersicherheit: Maschinenrätsel

Nom/Name : Djermoune
Prénom /Vorname: Amayes
Section/Abschnitt : 1CS
Groupe/Band : 01

Année Universitaire : 2021/2022
Chargé du module : Pr. Kamel Adi

Sommaire

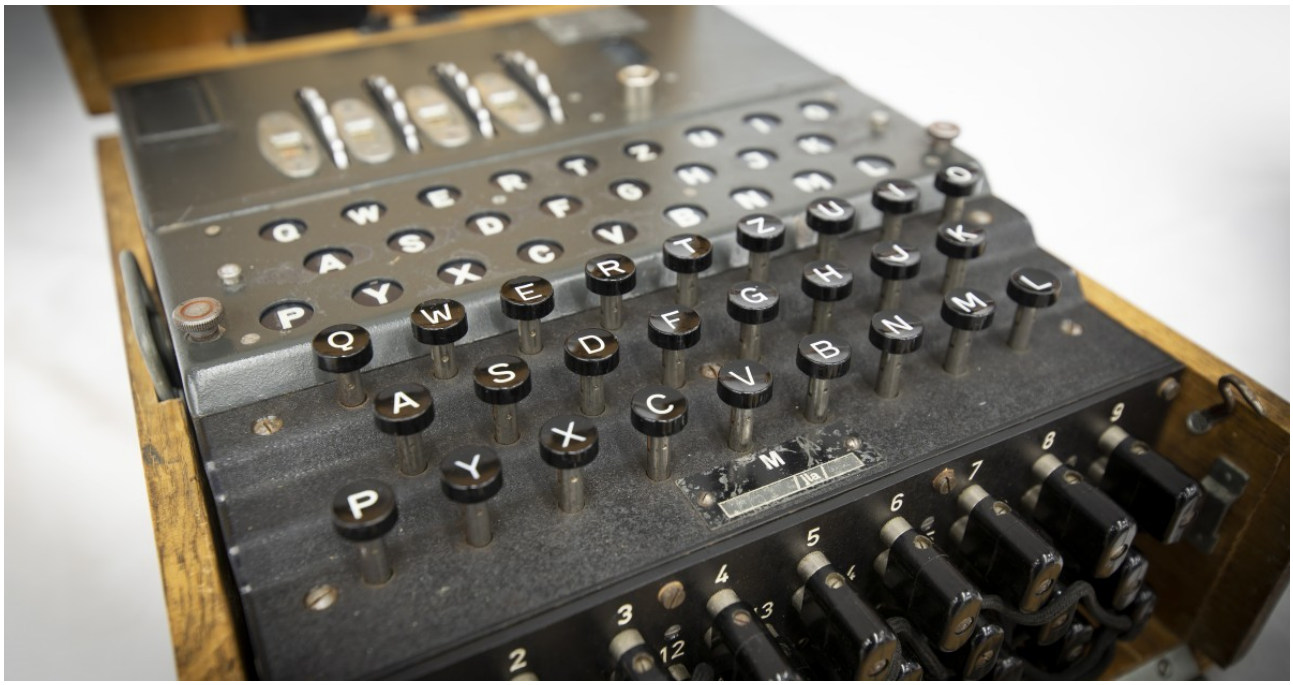
« Je dédie ce projet à ma mère que je ne l'ai plus revu il y'a 4 ans de cela que j'aime tant…... »

Introduction :

Depuis la nuit des temps, la cryptographie a toujours été un outil puissant dans divers domaines, surtout si ça parle de domaine militaire, étant donné la sensibilité des données et des informations transmises. Ce principe, Jules César l'a bien compris et a établit une Chiffrement nommé « Chiffrement de César » qui fut efficace. Sauf qu'en fonction du temps, certains ont pu déchiffrer et casser le code César, ce qui était inquiétant dans le sens ou le concept d'un système cryptographique parfait n'était rien de plus qu'un mythe, du moins pas les allemands .

Lors de la seconde guerre mondiale, les allemands gagnait en terre et en ressources en jouant sur les petits détails qui étaient décisives et si on remonte aux racines, on va retrouver que les allemands ont assuré la sécurité des informations confidentiels et sensibles.

Comment ? La réponse se trouve derrière une des machines cryptographiques les plus sophistiquées et classiques de l'histoire : Enigma

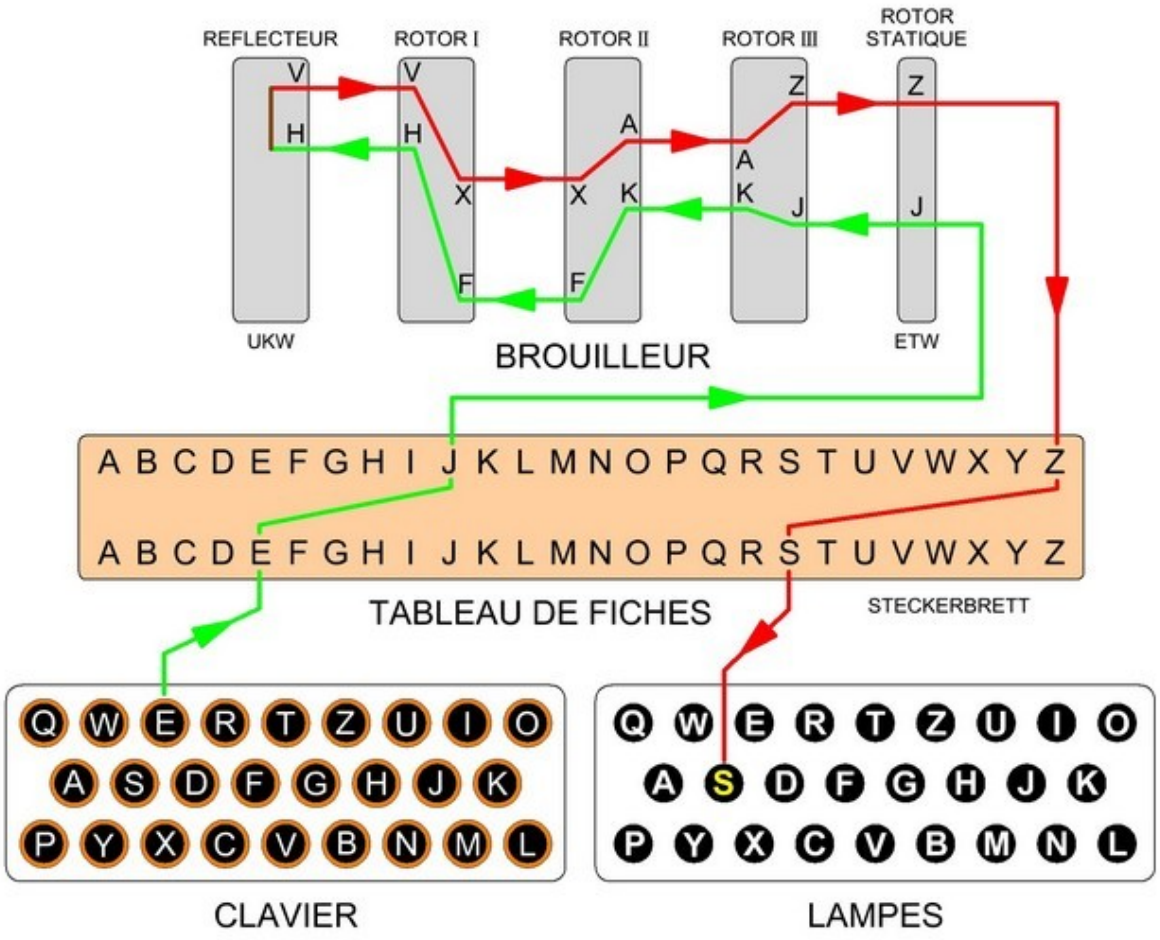


C’est quoi l’Enigma ? :

Enigma est la machine à chiffrer et déchiffrer qu'utilisèrent les armées allemandes du début des années trente jusqu'à la fin de Seconde Guerre Mondiale. Elle automatise le **chiffrement par substitution**. Cette machine ressemble à une machine à écrire. Quand on presse sur une touche, deux choses se passent. Premièrement, une lettre s'allume sur un panneau lumineux: c'est la lettre chiffrée. Deuxièmement, un mécanisme fait tourner le rotor de droite d'un cran; toutes les 26 frappes, le deuxième rotor tourne d'un cran, toutes les 676 frappes (26 au carré), c'est le troisième rotor qui tourne d'un cran. Certaines Enigmas avaient 3 rotors, celles de la **Kriegsmarine** en avaient 4 ou 5. Ces rotors tournants modifient les connexions électriques dans la machine, ce qui fait que la touche "A" allumera peut-être le "B" la première fois, mais le "X" la deuxième, le "E" la troisième, etc.

Comment Fonctionne l’Enigma ? :

Le principe de base des machines Enigma conçues par **Scherbius** repose sur l'utilisation de rotors qui transforment l'alphabet clair en alphabet chiffré :



Si on frappe la lettre **E** sur le clavier, un courant électrique est envoyé dans le rotor qui suit le câblage interne, puis ressort à droite pour allumer la lettre **S** sur le tableau lumineux ce qui est en soi un fonctionnement basique.

Autre principe de base: chaque fois qu'une lettre est tapée au clavier, le rotor tourne d'un cran (ce qui est l'objectif du TP) . Ainsi, **E** devient **S** la première fois, **F** la deuxième, **P** la troisième, etc. Par contre, et c'est une faiblesse de la machine qui sera exploitée pour la casser, **E** ne sera jamais chiffré **E**.

Le tableau de fiches (Steckerbrett) permet de brouiller les pistes en reliant deux lettres du clavier entre elles (ici **E** et **J**). Ainsi, quand on tape **E**, le courant prend en fait le circuit prévu pour **J**.

Les trois rotors multiplient ainsi le nombre de combinaisons. Le deuxième et le troisième avancent respectivement d'un cran quand le premier et le deuxième ont fait un tour complet.

Quant au réflecteur, il renvoie le courant dans le dispositif jusqu'au panneau lumineux où la lettre cryptée s'affiche. Son rôle n'est pas d'augmenter le nombre de combinaisons possibles, mais de faciliter considérablement la tâche du destinataire. En effet, si **E** devient **S** dans notre exemple, on a aussi **S** devient **E**. Et c'est valable pour toutes les paires de lettres claire/cryptée. Conséquence: si le mot **EFFACE** est chiffré **ACBFEB** par l'émetteur, il suffira à l'opérateur qui reçoit le message crypté de taper **ACBFEB** sur son clavier pour voir les lettres **E**, **F**, **F**, **A**, **C**, **E** s'allumer. Seule condition: les deux opérateurs distants doivent avoir réglé leur machine Enigma de la même façon.

Pourquoi Est-ce difficile de casser l'Enigma ?

Le côté sécurisé de cette machine est que même si elle tombe entre les mains ennemies, sa sécurité n'est pas compromise. En effet, c'est le nombre faramineux de réglages de la machine qui fait sa force et les réglages changeaient évidemment chaque jour. On peut en effet changer l'ordre des rotors, leur orientation initiale et les branchement du tableau de connexions. Par exemple, on pouvait spécifier la clé du jour ainsi:

- Position des rotors : C - D - A
- Orientations des rotors : C - X - E
- Branchements des connexions : A/L - P/R - T/D - B/W - K/F - O/Y
- Indicateurs : B - W - E

Au final, on a:

- 26 x 26 x 26 = 17576 combinaisons liées à l'orientation des chacun des trois rotors,
- 6 combinaisons possibles liées à l'ordre dans lequel sont disposés les rotors,
- 100391791500 branchements possibles quand on relie les six paires de lettres dans le tableau de connexions.

Les machines Enigma à 3 rotors peuvent donc chiffrer un texte selon 17576 x 6 x 100391791500 = 10000000000000000 combinaisons différentes!

Ainsi, connaître le fonctionnement de la machine n'aide (presque) pas à décrypter les messages qu'elle produit. Tout le problème est de retrouver le bon réglage et encore, faut etre rapide pour

décrypter le message à temps car les allemands changeaient de clé suivant un rythme hebdomadaire .En conséquence , les **bombes de Turing** ont vu le jour .

Outils:

Langages de programmation utilisés : Python, C++, JavaScript .

Python :

Une des raisons pour lesquelles j'ai opté pour Python est le fait que Python est riche question librairie y compris une librairie particulière : py-enigma du MIT. Ça m'a aussi beaucoup aidé afin de se focaliser beaucoup plus sur la sémantique de se soucier de la syntaxe .

C++ :

Alors initialement, je comptais travailler avec du C++ pour profiter de sa rapidité y compris dans le contexte de la cryptographie, mais je me suis retrouvé avec un code hyperbolisant mais sans interface graphique, ce qui était problématique . Du coup, j'ai utilisé le C++ mais pas sous forme d'un code que vous trouverez dans mon application, mais plutôt comme un intermédiaire me permettant de se documenter sur la bibliothèque py-enigma , étant donné qu'elle est écrite en C++.

JavaScript :

JavaScript m'a été utile surtout dans l'implémentation de l'interface graphique de l'Enigma y compris divers fonctionnalités et une flexibilité unique, à savoir allumer les lettres encryptés, interface graphique avec l'aide du CSS et ainsi de l'animation afin de montrer que le roter marche et tourne lorsqu'on tape une lettre, sans oublier l'implémentation d'un cahier afin de noter ce qu'on a écrit.

PS : J'ai pas noté HTML et CSS car ce ne sont pas des langages de programmation:)

Références :

https://web.stanford.edu/class/cs106j/handouts/36-TheEnigmaMachine.pdf

http://stanford.edu/class/archive/cs/cs106a/cs106a.1164/handouts/31-EnigmaSlides.pdf

https://web.stanford.edu/class/cs106j/assignments/Assignment-05/

https://pypi.org/project/py-enigma/

https://py-enigma.readthedocs.io/en/latest/

https://www.youtube.com/watch?v=ybkkiGtJmkM

The Imitation Game Film

Difficultés Rencontrés :

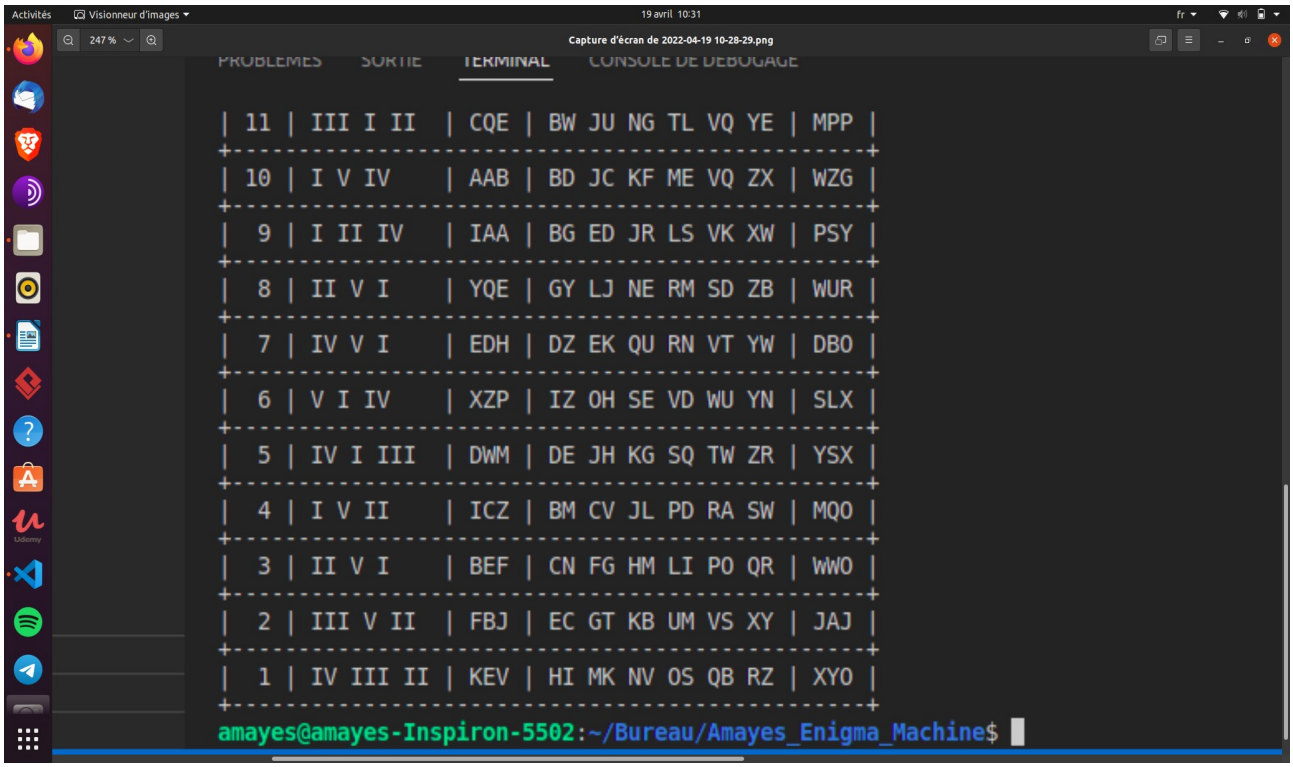
1. Choix des langage de programmation .
2. Choix de la structure de donnée (Tableau circulaire ou une Liste Linéaire Bidirectionnelle).
3. Implémentation de la fonction d'encryption et de décryption.
4. La rotation et fonctionnalité des roters (Configuration Initiale, Rotation entre eux
5. Implémentation des étapes d'encryption.
6. Mise en œuvre de l'interface graphique.
7. Ajout d'améliorations sans sortir du contexte du cahier de charge (Plugboard, Interface graphique différente et plus originale par rapport à celle de l'énoncé ….)

Application :

1. dailySettings :

Alors si on se mets dans le contexte de guerre, il est à noter que les allemands changent de configuration de façon hebdomadaire et afin d'assurer le bon passage du message, il faut que les deux machines soient configurés avec la même configuration. D'où ce programme va nous permettre de générer une liste contenant les configurations hebdomadaires de façon aléatoire . Par exemple, le 1er jour d'avril je vais utiliser une telle configuration, le deuxième une autre etc.

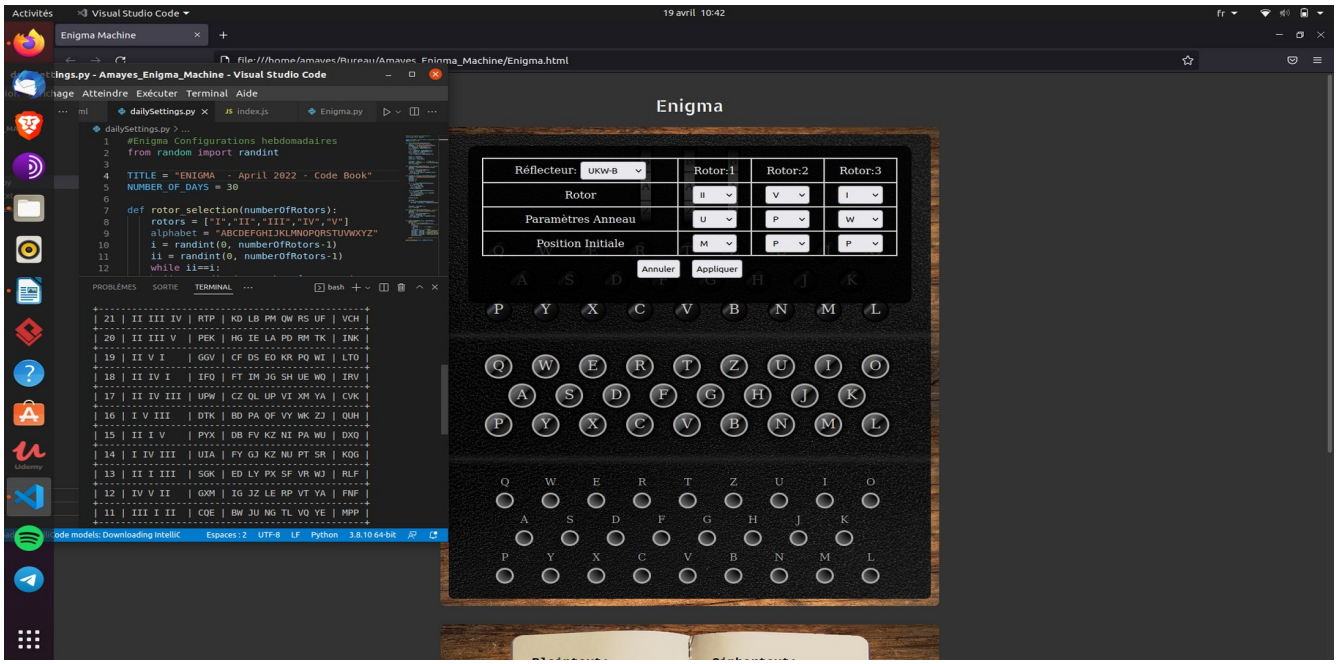
Pour faciliter l'exécution de la tache, j'ai opté pour un makefile afin de générer ceci en tapant dans le terminal la commande make . La première ligne représente les roters que je vais utiliser, la deuxième colonne représente la position de l'anneau. La Troisième est faite pour déclarer les permutations entre les lettres et pour finir, la dernière colonne est faite pour indiquer la position initiale des roters.



Enigma.html (Programme Principal):

Exemple de déroulement :

1. Nous sommes le 19 Avril , donc on prends la configuration du 19 avril après avoir généré les configurations initiales



2. Passons à la configuration :

Maintenant que nous avons finit la configuration de nos roters ainsi que la machine de l'Enigma, on passe à l'opération d'encryption/décryption :

On désire encrypter la phrase : Hello World

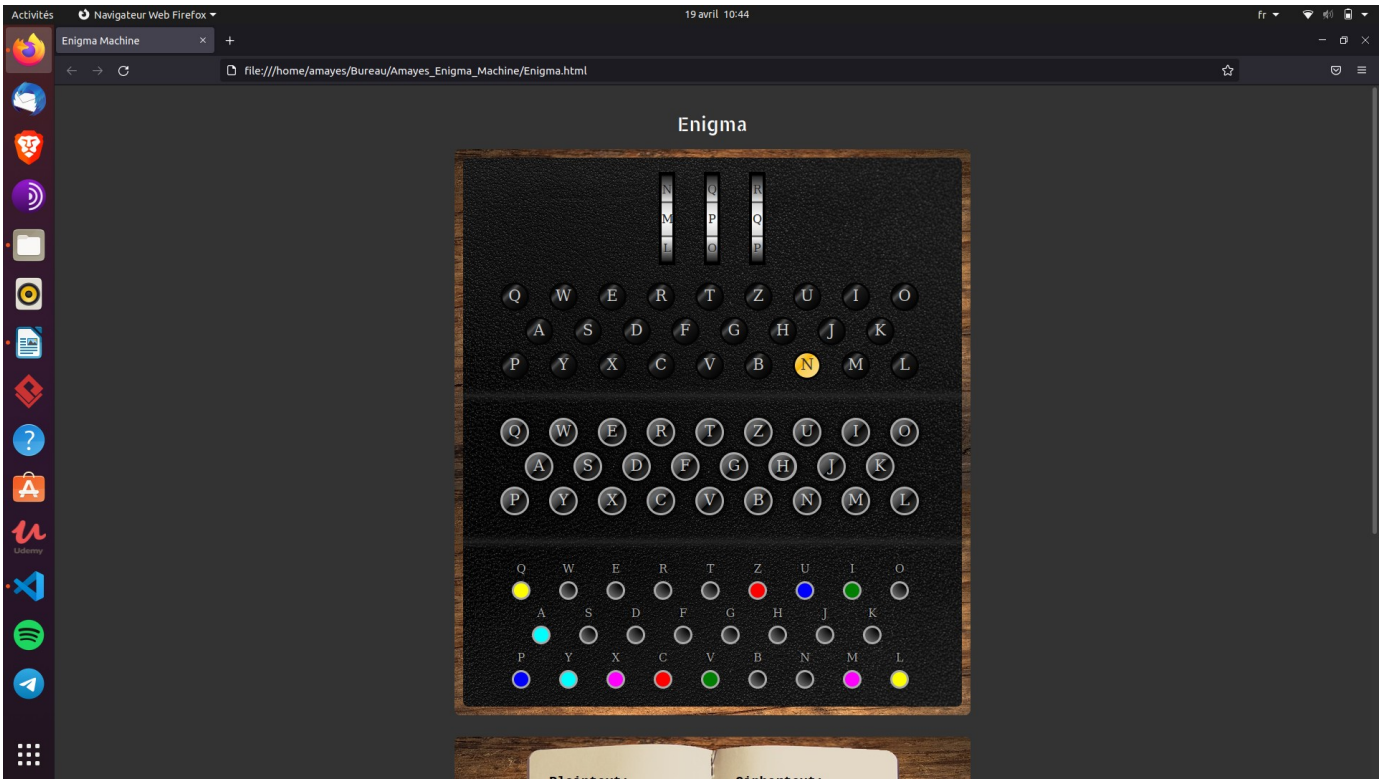
Ce qui va se passer c'est que je tapes la lettre H et on m'affiche la lettre encrypté qui est N et le roter à droite effectue une rotation et passe de P à Q .

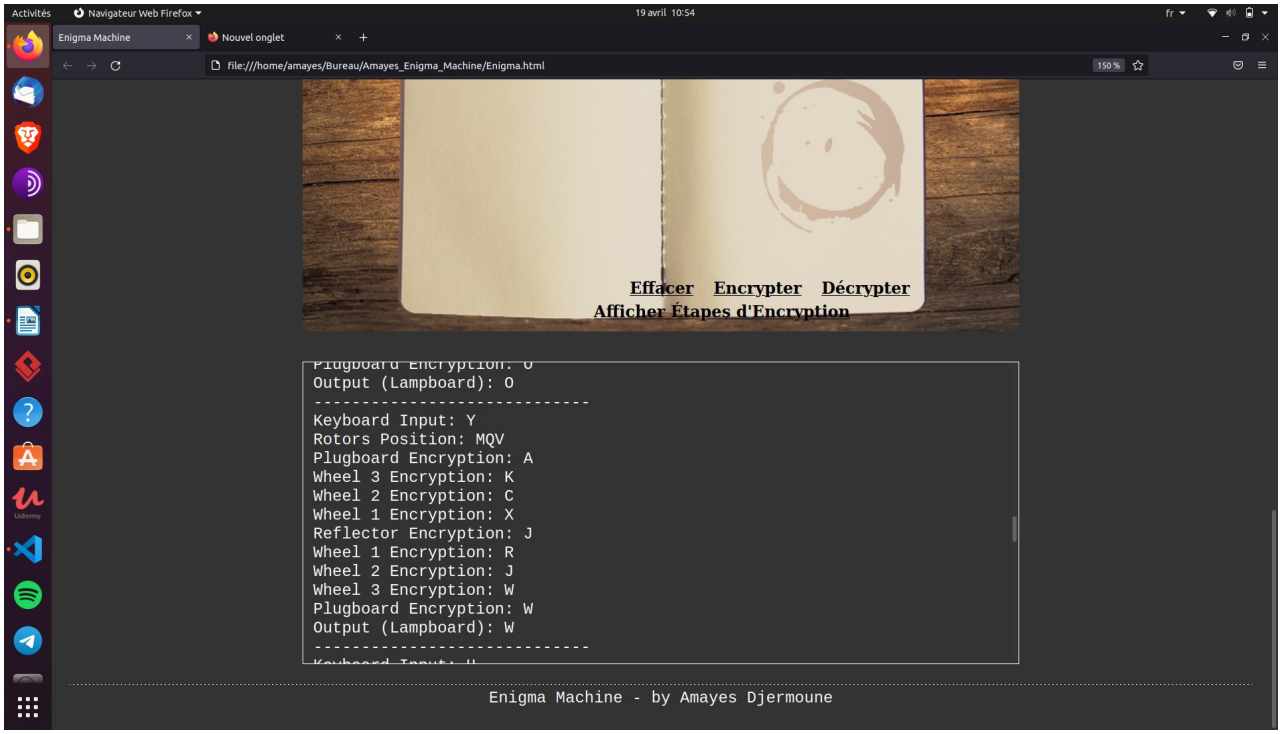
On fait pareil jusqu'à la fin du message.

Au final, dans mon cahier, j'aurais le texte que j'ai inséré ainsi que le texte encrypté ^^

Maintenant si je veux etre plus curieux et suivre les étapes d'encryption/décryption, avant de commencer le processus, je cliques sur le bouton pour m'afficher les étapes d'encryption. Au fur et à mesure, on va nous afficher les étapes d'encryption en détail:)

Bien sur, sans oublier que lorsqu'on veut passer à la décryption d'un message, on doit remettre les roters à la position initiale afin de refaire le meme processus d'encryption .





Ce que vous allez trouver dans ma solution :

1. dailySettings.py : Qui représente un programme qui va nous permettre d'afficher les configurations hebdomadaires
2. enigma.py : Ce code représente l'implémentation de l'Enigma mais sans interface graphique via python .
3. index.js : Ce programme représente la même implémentation de l'Enigma mais en JavaScript tour en ajoutant certains fonctionnalités et de l'animation pour l'interface graphique .
4. Enigma.html : Qui représente le programme principal

PS : Chacun d'entre eux est accompagné d'un fichier txt qui nous qui décrit les fonctionnalités du code en entier ainsi que certains fonctionnalités : Encryption, Décryption, Affichage des étapes d'encryption , affichage des couleurs, effacer le message du cahier, comment se déroule l'encryption et la liste est longue et riche .

Conclusion :

Au final, je tiens à vous remercier Mr.Adi , de m'avoir donné l'opportunité de travailler sur l'Enigma, qui est un projet qui me passionne et qui m'a permis d'apprendre d'implémenter les concepts de cryptographie reliés explicitement à la Cybersécurité et ma formation en tant qu'Ingénieur. Je tiens aussi à vous informer que j'ai eu pas mal d'inspiration concernant les projets de l'Enigma étant donné qu'il a déjà été donné comme TP à Stanford voir à 101Computing (surtout l'interface), mais j'ai préféré le faire à ma sauce, en prenant comme source d'inspiration les deux interfaces de Stanford et 101Computing ainsi que d'ajouter ma touche personnelle en terme de fonctionnalités étant donné que je me suis documenté sur la librairie py-enigma.

L’objectif derrière ce projet est d’approfondir ses connaissances en ce qui concerne la cryptographie, apprendre à effectuer ses recherches en profondeur, respecter son cahier de charge et surtout gagner en expérience concernant la réalisation des projets informatiques et ainsi exploiter ses connaissances et son savoir à son paroxysme.

D’où je vous invite à bien vouloir prendre mon travail en considération et m’attribuer la note que je mérite .