Rapport de projet pour le cours Introduction à la sécurité informatique

Florian Lefebvre

November 13, 2022

Contents

1 Etude bibliographique

Pour réaliser cette étude, nous avons commencé par faire une recherche bibliographique sur ce sujet. La carte heuristique ci-dessous montre les principales ressources que nous avons exploitées. C'est un sujet très connu qui a fait coulé beaucoup d'encre et qui continue a suscité aujourd'hui beaucoup d'intérêt.

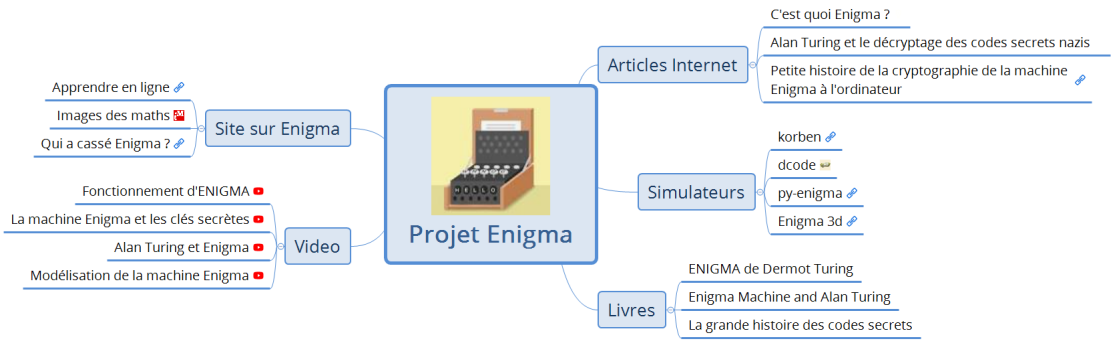

Figure 1: Ressources bibliographiques du projet

1.1 Historique de la machine Enigma

Figure 2: Une photo de la machine Enigma
source : https://www.apprendre-en-ligne.net/crypto/Enigma/

Enigma est une machine de cryptographie. Malgré son apparence de machine à écrire, elle a non seulement joué un rôle déterminant lors de la Seconde Guerre mondiale mais elle a aussi fait considérablement avancer l'histoire de l'informatique et de l'Intelligence Artificielle. Cette machine est d'abord commercialisée en Europe au début des années 1920. D'origine allemande, elle a été inventée par Arthur Scherbius, à des fins commerciales. Objectif, protéger les échanges d'informations entre banques grâce à un cryptage considéré à l'époque comme indéchiffrable. A partir de 1923, de nombreuses déclinaisons de cette machine seront commercialisées en Europe et dans le monde. Le nom « Enigma » sera alors employé de façon générique, pour désigner cette famille de machines. Ainsi, avant de devenir, grâce à Turing, un symbole du décryptage entre les mains des Alliés pendant la guerre, Enigma était surtout un outil efficace permettant de sécuriser les communications.

Figure 3: Enigma pendant la seconde guerre mondiale
source : https://interstices.info/turing-a-lassaut-denigma//

Si le mathématicien et cryptologue britannique Alan Turing est le plus connu pour avoir réussi à la décrypter, ce sont pourtant les Français et les Polonais qui sont à l'origine du déchiffrage d'Enigma. Dès 1931, les services français achètent des données secrètes d'Enigma grâce à Hans-Thilo Schmidt, un fonctionnaire allemand. En possession de ces élèments, ils vont alors les donner aux Polonais. "Les français vont rétrocéder une partie de leurs éléments aux Polonais qui ont rassemblé une équipe de mathématiciens très compétents." racontait Jean Lopez, directeur de la revue "Guerres et Histoire" et interviewé dans le journal du 13H de France 2 du 6 mai 2016. Au début de la Seconde Guerre mondiale, une équipe de Britanniques dont Alan Turing va alors bénéficier des travaux des Français et des Polonais. Grâce au déchiffrage de la machine Enigma, il a été estimé que le conflit en Europe a été raccourci de plusieurs mois.

1.2 Description de la machine d'Enigma

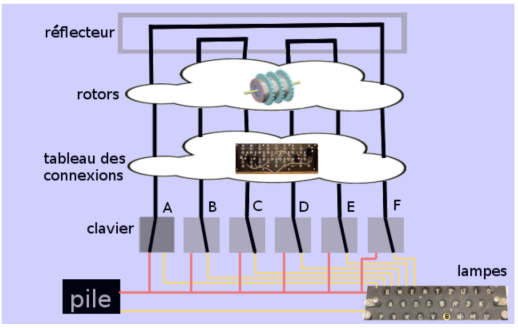

Figure 4: Schéma de la machine Enigma
source:https://images.math.cnrs.fr/La-machine-Enigma.html

Dans ce schéma, nous voyons les principaux éléments qui constituent la machine Enigma. Enigma chiffre les informations en faisant passer un courant électrique à travers une série de composants. Ce courant est transmis en pressant une lettre sur le clavier. Il traverse un réseau complexe de fils puis allume une lampe qui indique la lettre chiffrée. Le premier composant du réseau est une série de roues adjacentes, appelées « rotors », qui contiennent les fils électriques utilisés pour chiffrer le message. Les rotors tournent, modifiant la configuration complexe du réseau à chaque fois qu'une lettre est tapée. Enigma utilise habituellement une autre roue, appelée « réflecteur »,

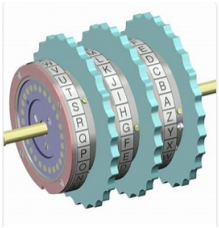

Figure 5: Les rotors de la machine Enigma

et un composant appelé « tableau de connexion », ce qui permet de rendre plus complexe encore le processus de chiffrement.

1.3 Présentation générale de son fonctionnement

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, suit le câblage interne, puis ressort à droite pour allumer la lettre S sur le tableau lumineux.

Autre principe de base: à chaque fois qu'une lettre est tapée au clavier, le rotor tourne d'un cran. Ainsi, E devient S la première fois, F la deuxième, P la troisième, etc. Par contre, 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.

1.4 Nos objectifs dans cette étude

Dans cette étude, nous souhaitons faire un état des lieux de cette machine Enigma, en collectant dans différentes ressources d'Internet, les informations qui émergent et nous semblent utiles. Ensuite, nous présenterons notre modélisation en s'appuyant sur une conception UML qui est une méthodologue très pratique pour modéliser un système complexe. Enfin, nous aborderons une perspective plus technique en présentant différents simulateurs que nous avons développés.

2 Modélisation de la machine Enigma

2.1 Approche théorique

Modélisation mathématique

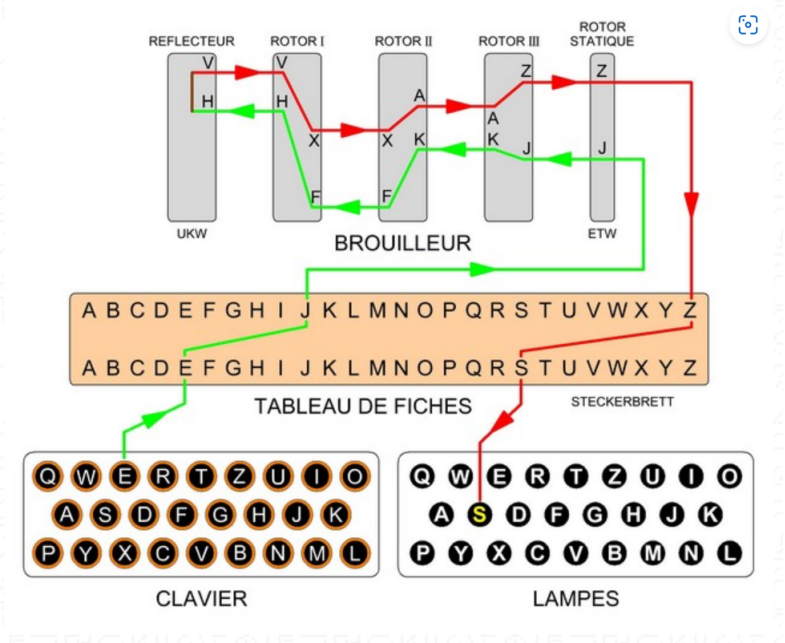

Figure 6: Les rotors de la machine Enigma

Le codage d'une lettre par une machine Enigma peut se traduire en une expression mathématique suivante : $MP_1P_1P_2P_3RP_3^{-1}P_2^{-1}P_1^{-1}$ où M est la lettre à coder, C est la lettre codée, P_1, P_2, P_3 sont les trois permutations de base et R le réflecteur. On constate que pour décoder un code d'Enigma, il suffit d'executer la même opération, ce qui se traduit par la formule ci-dessous. En effet R étant une permutation 2 à 2, on a $R^{-1} = \mathrm{R}$.

Ceci est pour le codage d'une lettre. Si l'on faisait cela pour le codage d'un message entier, chaque lettre serait toujours codé par la même lettre et donc une simple analyse statistique permettrait de décrypter le message. Pour éviter cela, Enigma, à chaque lettre tourne les permutations. La première permutation tournait à chaque lettre tapée. Pour la deuxième, toutes les 26 lettres et pour le troisième toutes les 676 lettres. Ainsi, pour un message de n lettres, le codage peut mathématiquement s'écrire ainsi :

$\mathrm{C} = M(p^iP_1p^{-i})(p^jP_2p^{-j})(p^kP_3p^{-k})R(p^k{P_3}^{-1}p^{-k})(p^j{P_2}^{-1}p^{-j})(p^i{P_1}^{-1}p^{-i})$

Avec :

p la permutation circulaire,

i le numéro de la lettre dans le message modulo 26,

j la partie entière du numéro de la lettre divisé par 676 modulo 26,

k la partie entière du numéro de la lettre divisé par 676 modulo 26,

P, M, R, C restant les mêmes que pour la première équation.

Calcul du dénombrement des possibilités

Pour dénombrer le nombre de combinaisons de la machine Enigma, partons du fait qu'elle utilise simultanément 3 rotors de 26 lettres chacun, c'est à dire 26 positions pour chaque rotors donc on a tout d'abord : 26 * 26 * 26 = 17576

Les 3 rotors (1,2,3) peuvent être placés selon les 6 dispositions suivantes: 123, 132, 213, 231, 312, 321. (Ce nombre passe a 120 avec 5 rotors à dispostions.)

Et il y a enfin le tableau de connexions à fiches qui est un système placé avant le premier rotor qui permet d'apparier 6 fois 2 lettres choisis parmi 26 en utilisant 6 câbles avant qu'elles ne rentrent dans le premier rotor.

Par exemple: en choisissant 2 lettres parmi 6 (donc avec un seul cable):

Première lettre => 6 choix possibles (par exemple j'ai choisi D parmi A,B,C,D,E,F)

Ensuite la lettre appariée à cette lettre "D" peut être A B C E F (on ne peut pas former un couple de 2 lettres identiques) ce qui nous donne 6*5 possibilités de couples de 2 lettres donc 30 choix . Toutefois des couples sont comptés en double par exemple les couples D/E et E/D sont équivalent donc le nombre de possibilités est divisé par 2. on a donc => (6*5)/2. Cela nous permet de calculer le nombre de possibilités pour 6 câbles parmi 26 lettres:

Première lettre choisie: 26 choix possibles de lettre appariée à la première: 25 choix possible, donc (26*25)/2 possibilitées différentes de placer le premier câble.

Ensuite il y a (24*23)/2 possibilitées pour le deuxième

puis (22*21)/2 possibilitées pour le troisième

puis (20*19)/2 possibilitées pour le quatrième

puis (18*17)/2 possibilitées pour le cinquième

et enfin (16*15)/2 possibilitées pour le sixième.

L'ordre des six cables n'a aucune importance (si l'on a choisi les couples: 1(A/B) 2(C/D) 3(E/F) 4(G/H) 5(I/J) 6(K/L) cela revient au même que si l'on avait choisi l'ordre 1(C/D) 2(I/J) 3(A/B) 4(E/F) 5(G/H) 6(K/L) .) pourtant selon notre calcul l'ordre des cables a une importance. Donc pour rectifier notre calcul, il faut diviser le tout par le nombre de combinaisons possible d'ordonner les 6 cables; c'est à dire par 6*5*4*3*2*1 = 6! =720 Il y a donc :

(26*25*24*23*22*21*20*19*18*17*16*15) / (2*2*2*2*2*6!)

= 100 391 791 500

Et enfin pour obtenir le nombre de combinaisons total, on doit multiplier les produits entre eux :

6 * 17576 * 100 391 791 500 = environ 10^{15} possibilités.

Etudions maintenant une approche logicielle de la machine Enigma en exploitant quelques diagrammes importants d'UML pour modéliser un système complexe.

2.2 Modèle de cas d'utilisation

Ce diagramme représente les différentes fonctionnalités de notre application. L'utilisateur peut utiliser notre simulateur de trois façon. La première d'entre elles est le lancement du simulateur sans artifice en cryptant un message ou en le décryptant. La seconde option est dite pédagogique. Le but est de permettre à un utilisateur de comprendre comment fonctionne cette machine en affichant tout ce qu'elle fait. La troisième option est le fait de tenter de faire le cryptage de la machine Enigma . Le but est de jouer à la remplacer. Mais en faisant cela, on apprend comment elle fonctionne. Ce qui est la logique d'un Serious Game : Apprendre en jouant. Dans notre architecture trois applications ont été développées représentant pour chacune d'entre elles ces trois options.

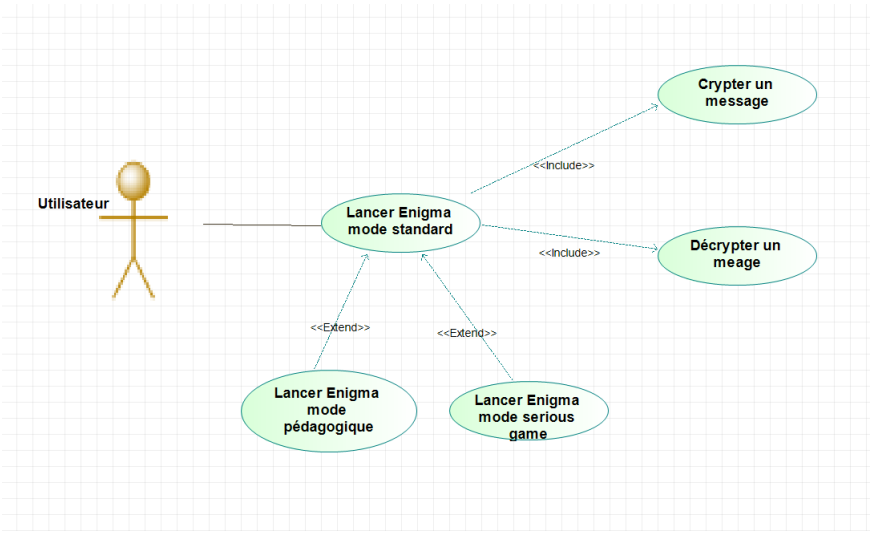

Figure 7: Le diagramme de cas d'utilisation de notre projet

2.3 Modèle statique

Un systeme complexe tel que la machine Enigma peut être étudié à l'aide d'une méthodologie telle qu'UML pour permettre ensuite de traduire cette analyse en conception. Le modèle qui suit a été réalisé sous Modélio qui est

un logiciel de conception orienté objet open source et présente la vision statique d'une machine Enigma. Elle s'appuie principalement sur le principe de décomposition où chaque composante se traduit par une classe en exploitant la relation d'agrégation d'UML. Dans ce schéma le clavier, comme les lampes

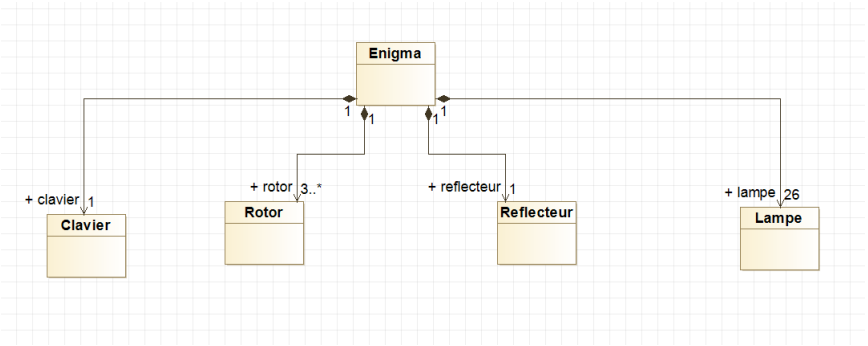

Figure 8: Le modèle statique de la machine Enigma

sont des éléments qui par la suite en développement se traduisent par des lectures clavier, et des affichages à l'écran. Ce modèle n'est pas universel mais permet de représenter les éléments essentiels de cette machine.

2.4 Modèle dynamique

Le modèle dynamique permet de mettre en évidence l'interaction des objets composant la machine Enigma. La figure ci-dessous est un scénario possible. Il matérialise les objets issus du modèle statique, des lignes de vie et des messages provoquant le passage à l'activité d'un d'autre objet. L'ordre des rotors dépend de la valeur de la clé qui détermine qui doit commencer à tourner en premier et dans quelle direction, il le fait, quel serait le deuxième et le troisième rotor à tourner et dans quelle direction. On voit sur la figure 9 l'aller et le retour du cryptage d'un caractère. Pour montrer qu'il faut faire cela sur l'ensemble des caractères du message, une boucle d'interaction de type "loop" en UML matérialise le fait qu'il faut bien sûr traiter l'ensemble des caractères du message. Le diagramme de séquence matérialise l'interaction entre les rotors mais ne montrent pas le changement de leur état interne.

Le diagramme d'état (figure 10) montre que le rotor 1, si c'est lui qui commence va à chaque cryptage d'un caractère tourner d'un cran. Dès qu'il fait un tour complet, c'est le rotor 2 qui fait la même chose alors que le rotor 1 est revenu à sa position initiale. La même chose se produit pour le rotor 3 et dès que le cycle est terminé, on repart avec un nouveau cycle.

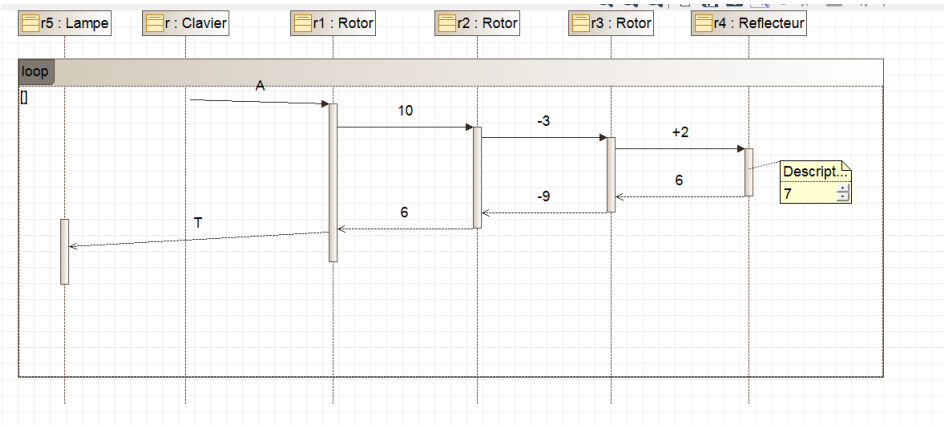

Figure 9: Le modèle dynamique de la machine Enigma

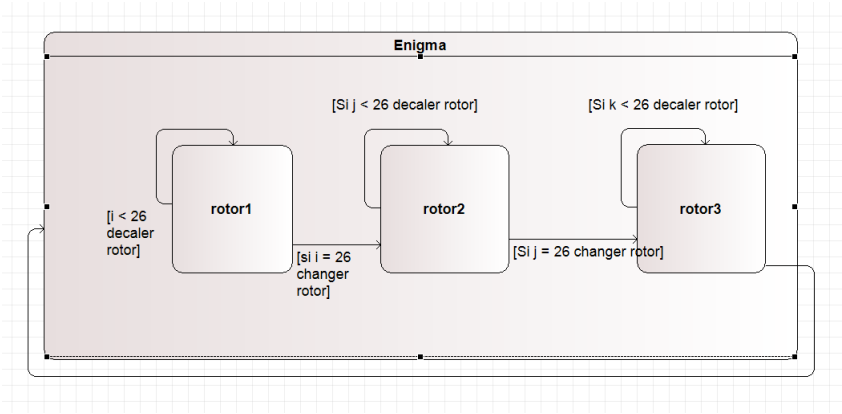

Figure 10: Le diagramme d'état d'un rotor

2.5 La cryptanalyse de la machine Enigma

Le site "https://interstices.info/turing-a-lassaut-denigma/" présente de façon très claire la cryptanalyse de la machine Enigma et son historique. L'extrait ci-dessous tiré de ce site résume les grandes lignes de cette cryptanalyse :

« Les premiers travaux de cryptanalyse de la machine Enigma ont été réalisés par des mathématiciens polonais, notamment Marian Rejewski, avant le début de la guerre. Leur attaque reposait sur une technique encore utilisée très fréquemment en cryptanalyse : la stratégie « diviser pour mieux régner ». Il s'agit de diviser le secret en deux parties (ici les rotors et le panneau de connexions) et de trouver un moyen de tester une de ces parties (par exemple, la position de départ des rotors) indépendamment de l'autre. Pour cela, Marian Rejewski exploitait la connaissance de quantité de couples correspondant à deux versions chiffrées d'une même lettre à trois positions d'écart. Ces

couples étaient obtenus car chaque communication allemande débutait par la transmission d'une nouvelle clé de trois lettres, qui était toujours répétée afin d'éviter les erreurs. Mais en 1939, peu de temps avant l'invasion de la Pologne, l'armée allemande décida de complexifier Enigma (en augmentant le nombre de rotors possibles et le nombre de transpositions dans le panneau de connexions), puis cessa de répéter les clés, rendant impossible l'attaque de Marian Rejewski. Les cryptanalystes polonais eurent néanmoins le temps de transmettre leurs travaux aux services de renseignement britanniques, qui lancèrent dans le plus grand secret l'opération dite « Ultra » pour décrypter les messages d'Enigma. Ces travaux étaient menés par une équipe de sept mille personnes, où Alan Turing se distingua.

Les cryptanalystes polonais eurent néanmoins le temps de transmettre leurs travaux aux services de renseignement britanniques, qui lancèrent dans le plus grand secret l'opération dite « Ultra » pour décrypter les messages d'Enigma. Ces travaux étaient menés par une équipe de sept mille personnes, où Alan Turing se distingua.

Mathématiciens, linguistes, ingénieurs... étaient regroupés dans un manoir anglais du nom de Bletchley Park.Ils utilisèrent le fait que certaines parties des messages étaient faciles à deviner. En particulier, les bulletins météo, transmis chaque matin à la même heure, contenaient tous le mot « Wetter » (qui signifie « temps » en allemand).

Alan Turing eut alors l'idée de rechercher des cycles de lettres. Pour qu'une suite de lettres forme un tel cycle, il faut que chaque lettre de la suite soit une version chiffrée de la lettre précédente, et que la version chiffrée de la dernière lettre corresponde à la première. Comme le tableau de connexions effectue les mêmes transpositions à la fin d'un chiffrement et au début du suivant, son effet s'annule complètement au sein d'un tel cycle. On dispose donc d'un motif caractéristique des rotors : il suffisait alors d'essayer toutes les possibilités pour les rotors pour trouver celle qui produisait le motif attendu. Pour tester le million de possibilités, Alan Turing fit construire des machines électromécaniques appelées « bombes », reproduisant les rotors d'Enigma et permettant d'essayer en parallèle jusqu'à vingt mille configurations par seconde. Une fois la position des rotors déterminée, il devenait possible de décrypter une partie du message (celle correspondant aux six lettres non affectées par le tableau de connexions) puis d'en déduire les transpositions.»

3 Implémentation de simulateurs de la machine Enigma

Il existe de nombreux simulateurs de la machine Enigma comme le montre la figure 13. Nous avons choisi dans cette étude de présenter trois approches possibles de la simulation de la machine Enigma.

3.1 Le simulateur graphique de Tom MacWrite

La première approche est un simulateur graphique de très grande qualité. Dans celui-ci vous pouvez chiffrer jusqu'à 20 caractères. Tout comme la machine d'origine, la ponctuation n'est pas prise en charge. Le plugboard peut être configuré avec des paires de caractères séparées par des espaces: par exemple: AB CD mappe A à B (et inversement) et C à D (et inversement). Les entrées non valides seront ignorées. Les rotors comme le réflecteur sont sélectionnables comme on le souhaite. La figure 11 montre l'interface graphique de ce simulateur avec de nombreuses options possibles. Une animation graphique permet de donner une approche très dynamique et très visuelle de ce simulateur. Il y a dans ce simulateur de nombreuses possibilités pour la configuration des rotors. L'utilisation en sens inverse ne permet pas de retrouver le message d'origine. C'est néanmoins l'un des plus beaux simulateurs que nous ayons eu l'occasion de tester.

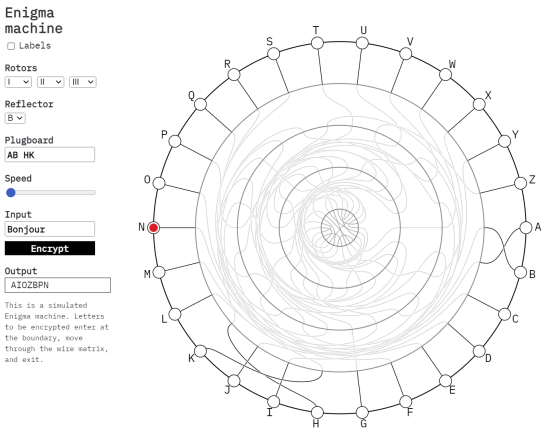

Figure 11: Schéma de la machine Enigma
source:https://observablehq.com/@tmcw/enigma-machine

3.2 Le simulateur Python py-enigma

Py-Enigma est une bibliothèque Python 3 pour simuler les machines Enigma utilisées par les forces armées allemandes (Wehrmacht) pendant la 2e Guerre mondiale. Py-Enigma permet à la fois de chiffrer et de déchiffrer les messages qui peuvent être envoyés ou reçus à partir de machines Enigma réelles utilisées par l'armée allemande (Heer), l'armée de l'air (Luftwaffe) et la marine (Kriegsmarine). Cet exemple montre comment cette librairie peut être

```
from enigma.machine import EnigmaMachine

# setup machine according to specs from a daily key sheet:

machine = EnigmaMachine.from_key_sheet(
       rotors='II IV V',
       reflector='B',
       ring_settings=[1, 20, 11],
       plugboard_settings='AV BS CG DL FU HZ IN KM OW RX')

# set machine initial starting position
machine.set_display('WXC')

# decrypt the message key
msg_key = machine.process_text('KCH')

# decrypt the cipher text with the unencrypted message key
machine.set_display(msg_key)

ciphertext = 'NIBLFMYMLLUFWCASCSSNVHAZ'
plaintext = machine.process_text(ciphertext)

print(plaintext)
```

Figure 12: Exemple de programme
source:https://pypi.org/project/py-enigma/

utilisée facilement. Elle peut simuler trois ou quatre rotors et possède une documentation au niveau de son api très fournie. Cela montre vraiment l'engouement de nombreux projets pour simuler cette machine.

3.3 Le simulateur 3d

Ce simulateur est le plus impressionnant car il permet d'utiliser la machine Enigma à l'aide d'une représentation 3d et donne lieu à un réalisme impressionnant. Il nécessite une petite phase d'adaptation pour apprendre à l'utiliser mais donne lieu ensuite à une expérience inoubliable comme si on revenait à l'époque de l'utilisation de cette machine. C'est un projet très original qui permet d'être comme dans un espace virtuel en trois dimensions où l'interaction offerte dans ce simulateur est renversant de réalité. Il est sûrement parmi tous les simulateurs le plus beau au niveau de son interface Homme-Machine. Il démontre encore une fois l'engouement qu'à générer

cette machine Enigma à travers l'envie de reproduire le plus fidèlement son fonctionnement.

Figure 13: Simulation 3d de la machine Enigma
source:https://enigma.virtualcolossus.co.uk/

De nombreux simulateurs ont été développé. Le tableau suivant présente un récapitulatif très intéressant de tous ces simulateurs.

Nom	Plateforme	Types	Uhr	UKW-D
Web Encryptor - The Online Encrypter[19]	React (développée par Facebook)	Enigma I, M3 (Army/Navy), M4 (Army/Navy), Railway, Tirpitz, Zahlwerk (Default/G-260/G-312), Swiss-K (Air Force/Commercial)	✗ Non	✓ Oui
Franklin Heath Enigma Simulator[20]	Android	K Railway, Kriegsmarine M3,M4	✗ Non	✗ Non
EnigmAndroid[21]	Android	Wehrmacht I, Kriegsmarine M3, M4, Abwehr G31, G312, G260, D, K, Swiss-K, KD, R, T	✗ Non	✗ Non
Andy Carlson Enigma Applet (Standalone Version)[22]	Java	Kriegsmarine M3, M4	✗ Non	✗ Non
Minarke (Minarke Is Not A Real Kriegsmarine Enigma)[23]	C/Posix/CLI (MacOS, Linux, UNIX, etc.)	Wehrmacht, Kriegsmarine, M3, M4	✗ Non	✗ Non
Russell Schwager Enigma Simulator[24]	Java	Kriegsmarine M3	✗ Non	✗ Non
PA3DBJ G-312 Enigma Simulator[25]	Javascript	G312 Abwehr	✗ Non	✗ Non
Virtual Enigma 3D [archive][26]	JavaScript	Wehrmacht, Kriegsmarine M4	✗ Non	✗ Non
Terry Long Enigma Simulator[27]	MacOS	Kriegsmarine M3	✗ Non	✗ Non
Paul Reuvers Enigma Simulator for RISC OS [archive][28]	RISC OS	Kriegsmarine M3, M4, G-312 Abwehr	✗ Non	✗ Non
Dirk Rijmenants Enigma Simulator v7.0[29]	Windows	Wehrmacht, Kriegsmarine M3, M4	✗ Non	✗ Non
Frode Weierud Enigma Simulators[30]	Windows	Abwehr, Kriegsmarine M3, M4, Railway	✗ Non	✗ Non
Alexander Pukall Enigma Simulator[31,32]	Windows	Wehrmacht, Luftwaffe	✗ Non	✗ Non
CrypTool 2 — Enigma component and cryptanalysis[33]	Windows	A/B/D (commercial), Abwehr, Reichsbahn, Swiss-K, Enigma M3, Enigma M4	✗ Non	✗ Non

Figure 14: Les simulateurs
source:https://enigma.virtualcolossus.co.uk/

4 Notre simulateur en Java

Le petit simulateur que nous décrivons maintenant n'a pas comme ambition de surpasser la qualité des simulateurs que nous venons de décrire rapidement, mais simplement de proposer un modèle simple de programmation orienté objet avec le langage Java et de le destiner plus à un environnement pédagogique pour faciliter l'appropriation du fonctionnement de la machine Enigma. Nous proposons trois modèles de simulateur pour illustrer ce fonctionnement. Le premier est le plus simple et consiste à crypter un message. Le second modèle est de fournir à l'utilisateur une trace détaillée de ce que fait le simulateur pour mieux comprendre son fonctionnement interne. Le troisième modèle permet de s'entrainer en demandant à l'utilisateur de faire lui-même le calcul des états de la machine Enigma pour voir s'il a bien compris son fonctionnement. Nous pouvons comparer ce troisième modèle à un Serious Game dont l'objectif est d'apprendre en jouant. Il y a eu dans cette approche un jeu très intéressant s'appelant "Turing Machine" qui est un jeu de déduction dont le but est de faire des hypothèses pour trouver le code de la machine Enigma.

Figure 15: Un jeu sur la machine Enigma
source://gusandco.net/2022/09/20/turing-machine-jeu-critique-deduction/

4.1 Etude de son fonctionnement

La machine que nous voulons programmer est composée de trois rotors et un réflecteur. Chaque rotor est modélisé par un tableau de deux lignes (1ère ligne–>ligne en haut; 2ème ligne –> ligne en bas) et de 26 colonnes. Lorsque nous tapons une lettre, les trois rotors sont parcourus de bas vers le haut jusqu'au réflecteur, puis de haut vers le bas jusqu'à trouver la lettre chiffrée correspondante.

Durant la phase d'aller (de la lettre tapée en direction du réflecteur), le chemin est calculé à partir de la deuxième ligne de chaque rotor. Ainsi lorsque nous tapons une lettre qui a l'indice i selon l'ordre alphabétique (A=0, B=1,. . .,Z=25), elle rentre dans la colonne i du rotor 1. Dans ce cas, la case [2, i]

(2ème ligne, i-ème colonne) du rotor 1, nous donne la valeur du décalage (à droite si la valeur est positive, à gauche si la valeur est négative) à faire pour trouver la colonne d'entrée du rotor 2. Par exemple, si nous avons une valeur v dans la case [2, i], cela veut dire que nous devrons entrer au rotor 2 par la colonne (i + v) mod 26. De la même manière, la colonne d'entrée du rotor 2 nous donne la colonne d'entrée du rotor 3 qui elle même va nous donner la colonne d'entrée du réflecteur. Finalement, la colonne d'entrée du réflecteur va nous donner la colonne d'entrée du rotor 3 pour la phase de retour.

Pendant la phase de retour, nous appliquons le même principe sauf que nous nous servirons de la 1ère ligne de chaque rotor pour calculer le chemin de retour. Le tableau (figure 21), à la fin de ce document, montre la configuration de notre machine Enigma.

4.2 Premier modèle : version console simple

Dans cette première version, l'utilisateur doit entrer un message. Le programme enlevera de ce message les espaces et affichera le message codée comme le montre la session suivante : Si nous reprenons le message codé et

```
Enigma version normale
Configuration du rotor 1 :
[17, 4, 19, 21, 7, 11, 3, -5, 7, 9, -10, 9, 17, 6, -6, -2, -4, -7, -12, -5, 3, 4, -21, -16, -2, -21]
[10, 21, 5, -17, 21, -4, 12, 16, 6, -3, 7, -7, 4, 2, 5, -7, -11, -17, -9, -6, -9, -19, 2, -3, -21, -4]
Configuration du rotor 2 :
[25, 7, 17, -3, 13, 19, 12, 3, -1, 11, 5, -5, -7, 10, -2, 1, -2, 4, -17, -8, -16, -18, -9, -1, -22, -16]
[3, 17, 22, 18, 16, 7, 5, 1, -7, 16, -3, 8, 2, 9, 2, -5, -1, -13, -12, -17, -11, -4, 1, -10, -19, -25]
Configuration du rotor 3 :
[12, -1, 23, 10, 2, 14, 5, -5, 9, -2, -13, 10, -2, -8, 10, -6, 6, -16, 2, -1, -17, -5, -14, -9, -20, -10]
[1, 16, 5, 17, 20, 8, -2, 2, 14, 6, 2, -5, -12, -10, 9, 10, 5, -9, 1, -14, -2, -10, -6, 13, -10, -23]
[25, 23, 21, 19, 17, 15, 13, 11, 9, 7, 5, 3, 1, -1, -3, -5, -7, -9, -11, -13, -15, -17, -19, -21, -23, -25]
Votre chaine à crypter ?
mieux vaut tard que jamais
chaine sans espace : mieuxvauttardquejamais
La chaine finale cryptée est : QPKETHPBDWDPUFIMXBTCYK
```

Figure 16: Une session avec le premier simulateur

que nous demandons à notre simulateur de crypter ce message, il est logique que nous retrouvions le message d'origine. Nous avons donc copier le message crypté dans le presse-papier et relancer notre simulateur en lui donnant comme message celui qui est crypté. La figure suivante nous montre bien que le simulateur a retrouvé le message d'origine sans les espaces puisque ceux-ci sont enlevé lors de la première étape.

```
Enigma version normale
Configuration du rotor 1 :
[17, 4, 19, 21, 7, 11, 3, -5, 7, 9, -10, 9, 17, 6, -6, -2, -4, -7, -12, -5, 3, 4, -21, -16, -2, -21]
[10, 21, 5, -17, 21, -4, 12, 16, 6, -3, 7, -7, 4, 2, 5, -7, -11, -17, -9, -6, -9, -19, 2, -3, -21, -4]
Configuration du rotor 2 :
[25, 7, 17, -3, 13, 19, 12, 3, -1, 11, 5, -5, -7, 10, -2, 1, -2, 4, -17, -8, -16, -18, -9, -1, -22, -16]
[3, 17, 22, 18, 16, 7, 5, 1, -7, 16, -3, 8, 2, 9, 2, -5, -1, -13, -12, -17, -11, -4, 1, -10, -19, -25]
Configuration du rotor 3 :
[12, -1, 23, 10, 2, 14, 5, -5, 9, -2, -13, 10, -2, -8, 10, -6, 6, -16, 2, -1, -17, -5, -14, -9, -20, -10]
[1, 16, 5, 17, 20, 8, -2, 2, 14, 6, 2, -5, -12, -10, 9, 10, 5, -9, 1, -14, -2, -10, -6, 13, -10, -23]
[25, 23, 21, 19, 17, 15, 13, 11, 9, 7, 5, 3, 1, -1, -3, -5, -7, -9, -11, -13, -15, -17, -19, -21, -23, -25]
Votre chaine à crypter ?
QPKETHPBDWDPUFIMXBTCYK
chaine sans espace : QPKETHPBDWDPUFIMXBTCYK
La chaine finale cryptée est : MIEUXVAUTTARDQUEJAMAIS
```

Figure 17: le message d'origine

4.3 Deuxième modèle : version console pédagogique

Dans ce deuxième modèle de simulateur, le programme est beaucoup plus verbeux et visualise toutes les étapes de transformation du message d'origine au message crypté. La figure suivante présente une session de ce simulateur mais avec un message court pour limiter la taille de la session à intégrer dans ce document. Les deux figures suivantes montrent les deux phases de cette sessions, la première montre le passage aller et la seconde le passage retour.

```
Enigma version pédagogique
Les valeurs du Rotor 1
[17, 4, 19, 21, 7, 11, 3, -5, 7, 9, -10, 9, 17, 6, -6, -2, -4, -7, -12, -5, 3, 4, -21, -16, -2, -21]
[10, 21, 5, -17, 21, -4, 12, 16, 6, -3, 7, -7, 4, 2, 5, -7, -11, -17, -9, -6, -9, -19, 2, -3, -21, -4]
Les valeurs du rotor 2
[25, 7, 17, -3, 13, 19, 12, 3, -1, 11, 5, -5, -7, 10, -2, 1, -2, 4, -17, -8, -16, -18, -9, -1, -22, -16]
[3, 17, 22, 18, 16, 7, 5, 1, -7, 16, -3, 8, 2, 9, 2, -5, -1, -13, -12, -17, -11, -4, 1, -10, -19, -25]
Les valeurs du rotor 3
[12, -1, 23, 10, 2, 14, 5, -5, 9, -2, -13, 10, -2, -8, 10, -6, 6, -16, 2, -1, -17, -5, -14, -9, -20, -10]
[1, 16, 5, 17, 20, 8, -2, 2, 14, 6, 2, -5, -12, -10, 9, 10, 5, -9, 1, -14, -2, -10, -6, 13, -10, -23]
Les valeurs du réflecteur
[25, 23, 21, 19, 17, 15, 13, 11, 9, 7, 5, 3, 1, -1, -3, -5, -7, -9, -11, -13, -15, -17, -19, -21, -23, -25]

Votre chaine à crypter ?
ab
chaine sans espace : ab
caractère à crypter :a
Passage du crypatage aller des rotors 1 à 3

rotor 1 :
indice d'entrée du rotor 1 : 0 valeur : 10
indice d'entrée du rotor 2 :10

rotor 2 :
indice d'entrée du rotor 3 : 7 valeur : 2

rotor 3:
indice d'entrée du réflecteur : 9 valeur : 7

aller: 0 10 7 9
```

Figure 18: passage aller du simulateur version pedagogique

```
Passage du crypatage retour des rotors 3 à 1

rotor 3 :
indice d'entrée du rotor 3 : 16 valeur : 6
indice d'entrée du rotor 2 : 22 valeur : -9

rotor 2 :
indice d'entrée du rotor 1 : 13 valeur : 6
rotor 1 :
fin du chiffrement d'une lettre
Valeur crypté : T
decaler le rotor actif d'une position selon l'ordre et la direction indiques dans la clef
Les valeurs du rotor 2
[-16, 25, 7, 17, -3, 13, 19, 12, 3, -1, 11, 5, -5, -7, 10, -2, 1, -2, 4, -17, -8, -16, -18, -9, -1, -22]
[-25, 3, 17, 22, 18, 16, 7, 5, 1, -7, 16, -3, 8, 2, 9, 2, -5, -1, -13, -12, -17, -11, -4, 1, -10, -19]
caractère à crypter :b
Passage du crypatage aller des rotors 1 à 3

rotor 1 :
indice d'entrée du rotor 1 : 1 valeur : 21
indice d'entrée du rotor 2 :22

rotor 2 :
indice d'entrée du rotor 3 : 18 valeur : 1

rotor 3:
indice d'entrée du réflecteur : 19 valeur : -13

aller: 1 22 18 19

Passage du crypatage retour des rotors 3 à 1

rotor 3 :
indice d'entrée du rotor 3 : 6 valeur : 5
indice d'entrée du rotor 2 : 11 valeur : 5

rotor 2 :
indice d'entrée du rotor 1 : 16 valeur : -4
rotor 1 :
fin du chiffrement d'une lettre
Valeur crypté : M
decaler le rotor actif d'une position selon l'ordre et la direction indiques dans la clef
Les valeurs du rotor 2
[-22, -16, 25, 7, 17, -3, 13, 19, 12, 3, -1, 11, 5, -5, -7, 10, -2, 1, -2, 4, -17, -8, -16, -18, -9, -1]
[-19, -25, 3, 17, 22, 18, 16, 7, 5, 1, -7, 16, -3, 8, 2, 9, 2, -5, -1, -13, -12, -17, -11, -4, 1, -10]
la chaine finale cryptée est :  TM
```

Figure 19: passage retour du simulateur version pedagogique

4.4 Troisième modèle : version serious game

Cette dernière version est une pratique du fonctionnement de la machine Enigma. Le but est de proposer une valeur possible des rotors en appliquant l'apprentissage que l'on a effectué dans le simulateur précédent. A la fin de cette simulation, le programme classifie le joueur dans un niveau de compétences selon son nombre d'erreurs. Un joueur qui commet très peu de fautes sera jugé comme très bon, mais s'il commet trop d'erreur, on lui recommendera de réviser le fonctionnement de la machine Enigma. La figure suivante illustre un exemple de session de ce simulateur.

```
Passage du crypatage retour des rotors 3 à 1

rotor 3 :
Veuillez saisir l'indice du rotor 3 :
6
Bravo, vous avez trouvé la bonne valeur
indice d'entrée du rotor 3 : 6 valeur : 5
Veuillez saisir l'indice du rotor 2 :
9
Non, ce n'est pas la bonne valeur
indice d'entrée du rotor 2 : 11 valeur : 5

rotor 2 :
Veuillez saisir l'indice du rotor 1 :
1
Non, ce n'est pas la bonne valeur
indice d'entrée du rotor 1 : 16 valeur : -4
rotor 1 :
fin du chiffrement d'une lettre
Veuillez saisir la valeur de la lettre crypté ? :
e
Non, ce n'est pas la bonne valeur
Valeur crypté : M
decaler le rotor actif d'une position selon l'ordre et la direction indiques dans la clef
Les valeurs du rotor 2
[-22, -16, 25, 7, 17, -3, 13, 19, 12, 3, -1, 11, 5, -5, -7, 10, -2, 1, -2, 4, -17, -8, -16, -18, -9, -1]
[-19, -25, 3, 17, 22, 18, 16, 7, 5, 1, -7, 16, -3, 8, 2, 9, 2, -5, -1, -13, -12, -17, -11, -4, 1, -10]
la chaine crypter vaut TM
nombre de question : 16 nombre d'erreurs : 11
Mauvais score, continuer à vous entrainez !
```

Figure 20: exemple de session du simulateur version serious game

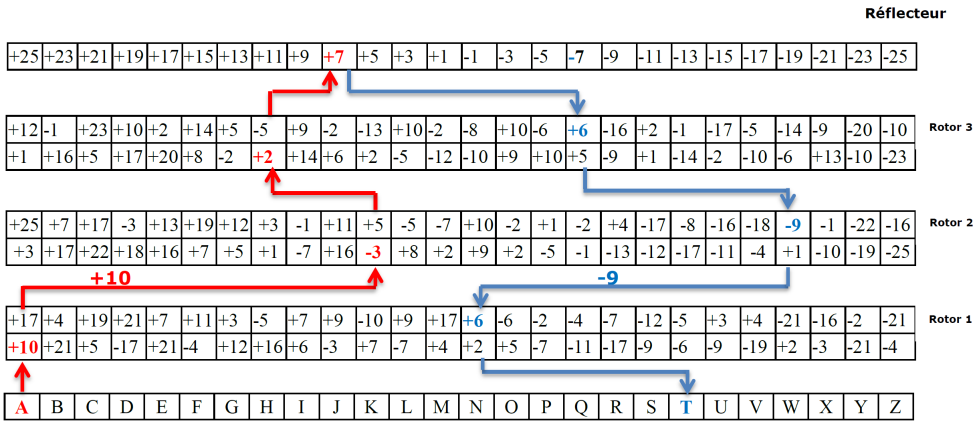

Figure 21: fonctionnement de notre machine Enigma

5 Conclusions

La machine Enigma a joué un rôle considérable pendant la seconde guerre mondiale. Son invention en 1918 constitue une véritable innovation technologique puisque cette machine est le premier système électromécanique de cryptage.

D'un point de vue historique, il est évident qu'une partie du combat va se jouer sur le terrain du déchiffrement des données cryptées. Les Anglais comprennent que décrypter rapidement les message passés par Enigma est un enjeu prioritaire. Un groupe de cryptanalystes se réunit secrètement dans le domaine de Bletchley Park, au Royaume-Uni. Alan Turing, cryptanalyste britannique les rejoint en 1939. En 1942, lui et ses collègues parviennent à déchiffrer les messages du régime nazi. Ce qui a changé le cours de l'histoire.

Le nombre d'articles, de livres, de simulateurs développés montrent bien l'importance de cette machine dans le domaine de la sécurité informatique et suscite encore aujourd'hui beaucoup d'intérêt et de curiosité. C'est dans cet esprit, et sans aucune prétention que ce document a été rédigé, en remerciant la qualité d'enseignement de ce module sur la sécurité enseignée à l'Université Paris 8 et la liberté de choisir en fin de ce module un thème nous tenant à coeur comme celui de la machine Enigma en ce qui nous concerne.