2014-2015

Projet Enigma

Compte-Rendu



DUT Informatique (S4)

TROUVE Robin

KISSI Naïm

DE LIMA Marianna

Encadré par:

M.ROY

Sommaire

**Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2**

1. **Enigma : Machine de cryptage . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3**
   1. Création d’Enigma (remise en contexte) **. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3**
      1. Remise en contexte
      2. Utilisation d’Enigma
   2. Fonctionnement d’Enigma (plugboard, rotors, circuit… + chiffrage/déchiffrage)
      1. Les rotors
      2. Le plugboard
      3. Le réflecteur
   3. Cryptanalyse par Turing
2. **Enigma : Simulation informatique**
   1. Organisation du projet (architecture mvc, méthodes de travail, outils)
   2. Représentation d’Enigma (ex : rotor = objet … comment on le conceptualise et comment il fonctionne dans le code, comment on a essayé de reproduire la machine physique)
   3. Difficultés rencontrées et solutions retenues
3. **Manuel utilisateur**

**Conclusion**

**Annexes**

Introduction

Dans le cadre de notre DUT Informatique, en semestre 4, nous sommes amenés à réaliser un projet alliant mathématiques et informatique afin d’appliquer nos connaissances et la méthodologie que nous avons pu acquérir au cours de notre formation.

Le sujet de ce projet est la machine Enigma, machine de cryptage la plus connue et utilisée lors de la Seconde Guerre Mondiale. Il s’agit donc ici de comprendre son fonctionnement, ses mécanismes et de pouvoir reproduire son comportement de manière informatisée.

Ce compte-rendu détaille notre démarche pour aborder le sujet, le comprendre et enfin réaliser techniquement le projet en s’organisant au sein du trinôme, en utilisant des outils et les méthodes appris en cours.

Nous allons donc voir dans une première partie ce qu’est la machine Enigma, son rôle, son fonctionnement et nous nous intéresserons aussi à sa cryptanalyse par Alan Turing pour comprendre ses failles. Nous verrons ensuite comment nous nous sommes organisés pour mener le projet à bien en mettant toutes nos connaissances en œuvre et nous verrons aussi les difficultés nous avons rencontrés et comment nous les avons surmontées.

1. **Enigma : Machine de cryptage**
   1. Création d’Enigma

Ecrire en 12 Times New Roman

Pour 1.1.1 remise en contexte :

Créateur ? Date ? Quelle méthode de cryptage existait avant? (voir méthode de vigenère) Quel était le but d’Enigma ?

Pour 1.1.2 Utilisation

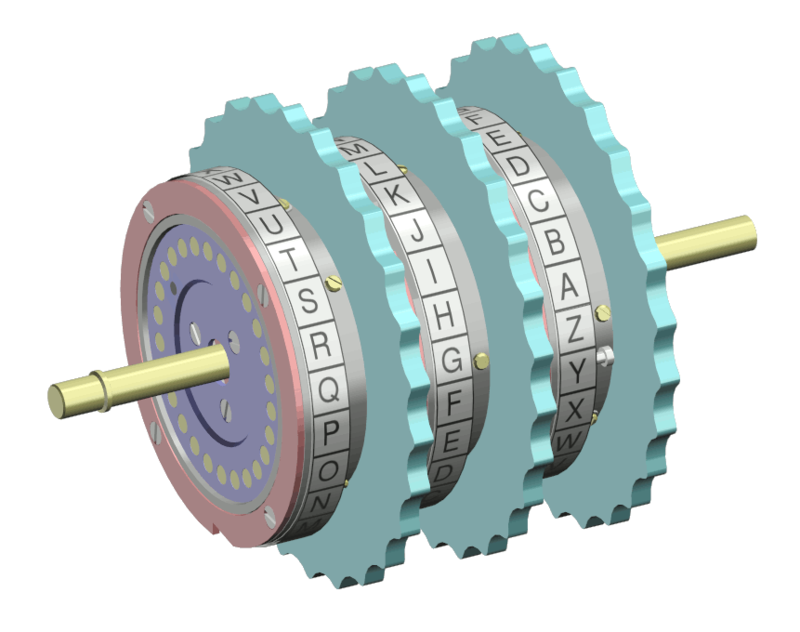
Qui s’en servait ? Comment il s’en servait (il s’envoyait le code pour paramétrer la machine, les messages étaient codés en morse, les 2 machines devaient être identiques)? Ne pas parler du fonctionnement interne mais externe si tu veux

* 1. Fonctionnement d’Enigma

1.2.1. Les rotors

Nous allons à présent voir en détail comment Enigma fonctionne, quelles sont ses particularités qui l’ont rendu si difficile à « casser ».

La puissance d’Enigma réside dans le fait qu’elle est imprévisible en terme de chiffrement des mots rentrés et cela est directement permis par sa structure interne. C’est la combinaison de ses composants qui la rend si performante mais quels sont-ils ? En tant que machine électromécanique, Enigma est composée de rotors fonctionnant avec des contacts électriques permettant la subsitution poly-alphabétique, c’est d’ailleurs une de ses particularités.

Les rotors servent de connexion électriques, ils sont au nombre de 3 dans la machine et pouvaient être choisis parmis 5 rotors en tout. Leur particularité vient du fait qu’ils sont cylindriques et fixés sur un axe autour duquel ils peuvent tourner. A chaque lettre tapée, le rotor effectue une rotation qui change complétement la donne pour la lettre suivante car du coup, la permutation sera différente pour chaque lettre même si celle-ci est retapée.

C’est un des points forts d’Enigma car de ce fait, l’analyse par fréquence qui consiste à repérer les lettres qui reviennent fréquemment dans un message crypté est impossible. Mais comme signalé, il y a 3 rotors en tout. Chaque rotor représente les lettres/chiffres que l’on veut crypter. Par exemple, si on prend l’alphabet de 26 lettres, chaque rotor aura 26 positions. Le 1er rotor tourne d’un cran à chaque fois qu’une lettre est tapée, de sa 26e position à sa position initiale, il déclenche le 2e rotor qui lui aussi tourne d’un cran. Lorsque le 2e rotor effectue sa rotation de sa 26e lettre à sa position initiale, il déclenche à son tour le 3e rotor. Les rotors reviennent à leur position initiale lorsqu’ils ont tous parcouru leur 26 positions. Chaque rotor pouvait être positionné de 1 à 26 avant de commencer à taper le message.

Nous avons schématisé les 3 rotors avec 4 lettres selon un circuit électrique de cette manière :

Rotor 1 Rotor 2 Rotor 3

A A

B B

C C

D D

On peut voir ici que si on tape lettre « A », elle sera permutée en « B » à l’issu du rotor 1 puis en « D » et enfin en « B ». Le rotor 1 tournera d’un cran et ses sorties seront décalées ce qui permettra d’avoir une lettre cryptée différente même si on retape à nouveau « A ».

Si on calcule le nombre de possibilités de cryptage qu’offrent les rotors, nous avons :

5x4x3=60

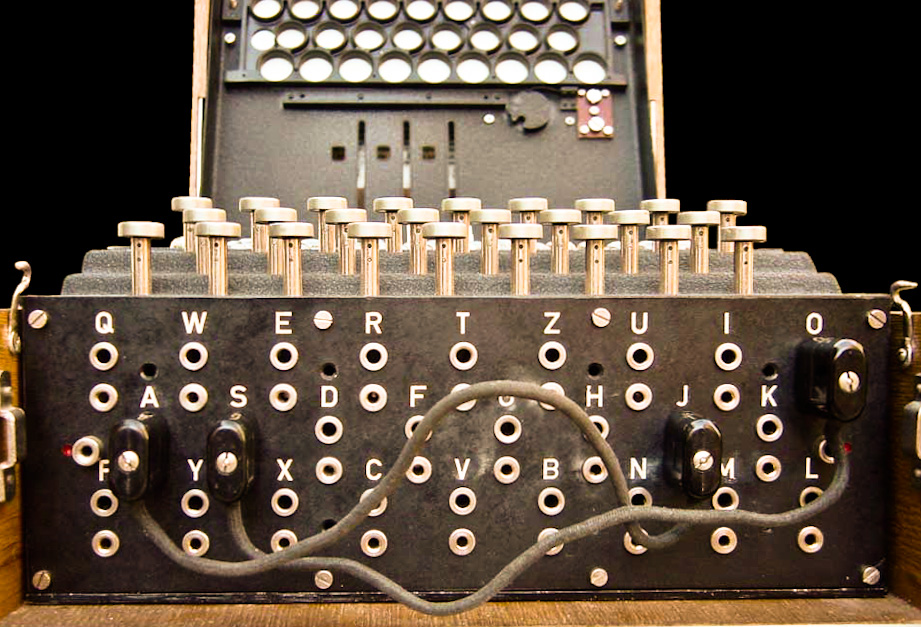
26x26x26=17 576

Soit : 60x17 576= 1 054 560

Jusqu’ici nous avons donc 60 possibilités de choix de rotors (3 parmis 5) et 17 576 (pour un alphabet de 26 lettres) positions possibles des rotors soit un total de 1 054 560 possibilités de cryptage. Si l’alphabet comporte plus de caractères, cela augmentera le nombre de positions possibles des rotors et donc le nombre de possibilités total.

1.2.2. Le plugboard

Mais Enigma ne s’arrête pas à ces trois rotors tournant, en effet, un des composants central de cette machine est le tableau de connexion (plugboard), situé devant la machine et permettant d’effectuer 10 paires de permutation, les autres restant invariantes. Le plugboard était spécialement conçu pour les machines utilisées à des fins militaires, les versions commerciales en étaient dépourvues.



Les permutations sont commutatives c’est-à-dire que si la lettre « A » est reliée à la lettre « B » alors la lettre « A » sera permutée avec la lettre « B » mais la lettre « B » sera permutée avec la lettre « A ».

Il y a donc 20 lettres permutées et 6 inchangées dans un alphabet de 26 lettres.

Le but du plugboard était de brouiller les pistes car une lettre tapée était d’abord permutée suivant le tableau de connexion puis codée.

Concrètement, cela signifie que si « A » était permutée avec « B », c’est la lettre « B » qui serait effectivement cryptée en passant dans les rotors.

Nous avons donc enrichi notre schéma (cf rotors ci-dessus) avec le plugboard ce qui nous donne sur un alphabet de 4 lettres :

Plugboard Rotor 1 Rotor 2 Rotor 3

A A

B B

C C

D D

Dans ce cas-ci, les lettres « A » et « B » sont permutées alors que les lettres « C » et « D » restent inchangées. Si on veut crypter la lettre « A », c’est donc en fait la lettre « B » qui va être cryptée et on obtiendra donc en permutations successives: A->B, B->A, A->B, B->D.

Si nous calculons les possibilités qu’offrent les branchements frontaux pour un alphabet de 26 lettres nous avons :

26 !/(6 !10 !2^10) = 150 738 274 937 250

En effet, nous avons 26 combinaisons de lettres (26x25x24…x1) ce qui justifie le « 26 ! ». Cependant, nous ne voulons faire que 10 paires de lettres donc 6 lettres restent non permutées. Etant donné que l’ordre des combinaisons n’importe pas, nous pouvons diviser par « 6 ! » et multiplier par le nombre de combinaison possible pour les 10 paires soit « 10 ! ». Enfin les paires sont constituées de deux lettres interchangeables, on peut alors diviser par « 2 » et comme il y a 10 paires de lettres cela nous donne « 2^10 »

C’est donc le plugboard qui permet d’avoir le nombre de possibilités le plus important.

//Si on rajoute les rotors : 158 962 555 217 826 360 000

Annexes :

<https://www.youtube.com/watch?v=G2_Q9FoD-oQ> par « numberphile »

<https://www.youtube.com/watch?v=7dpFeXV_hqs> par « e-penser »