

Projet Enigma

2014-2015


Naïm Kissi

Robin Trouve

Marianna De Lima



Plan

- ▀ Enigma: Machine de cryptage
 - ▀ Les rotors
 - ▀ Le plugboard
 - ▀ Le réflecteur
 - ▀ Enigma: Simulation informatique
 - ▀ Organisation du projet
 - ▀ Logique de fonctionnement
 - ▀ Difficultés et solutions retenues
- 

Les rotors



- Au nombre de trois parmi 5
- Forme cylindrique et fixé sur un axe où ils peuvent tourner
- Forme cylindrique et fixé sur un axe où ils peuvent tourner
- Une lettre ne sera pas cryptée deux fois par la même lettre

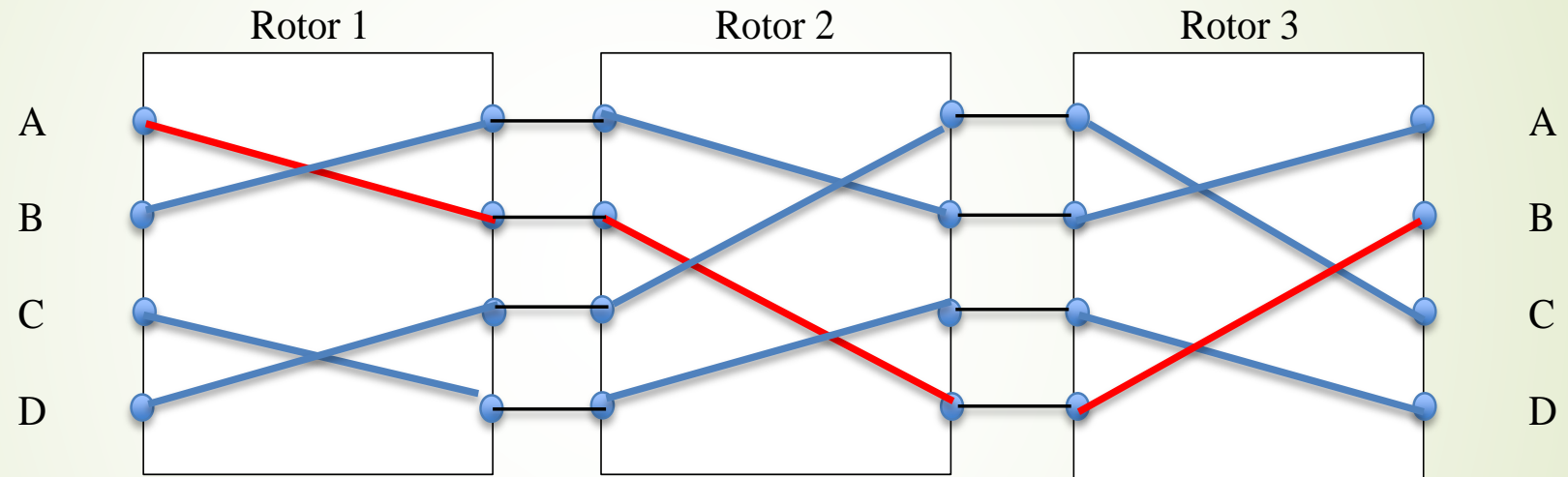
Nombre de possibilités:

$$5 * 4 * 3 = 60$$

$$26^3 = 17\ 576$$

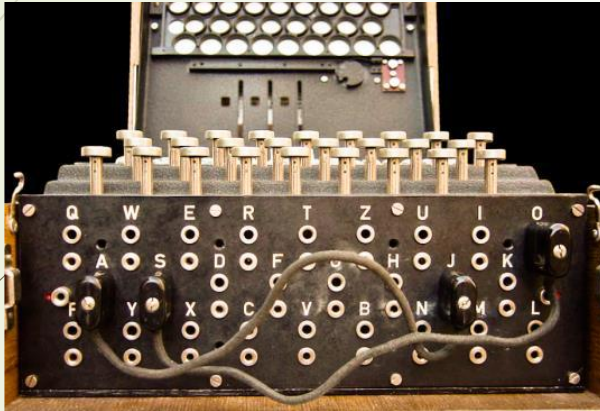
$$\text{Soit : } 60 * 17\ 576 = 1\ 054\ 560$$

Fonctionnement Rotors



- « A » cryptée en B
- Les rotors tournent: sorties/entrées décalées

Le Plugboard



- Tableau de connexion situé devant la machine
- Permet de permuter deux lettres entre elles
- Il offre le plus de possibilité de cryptage

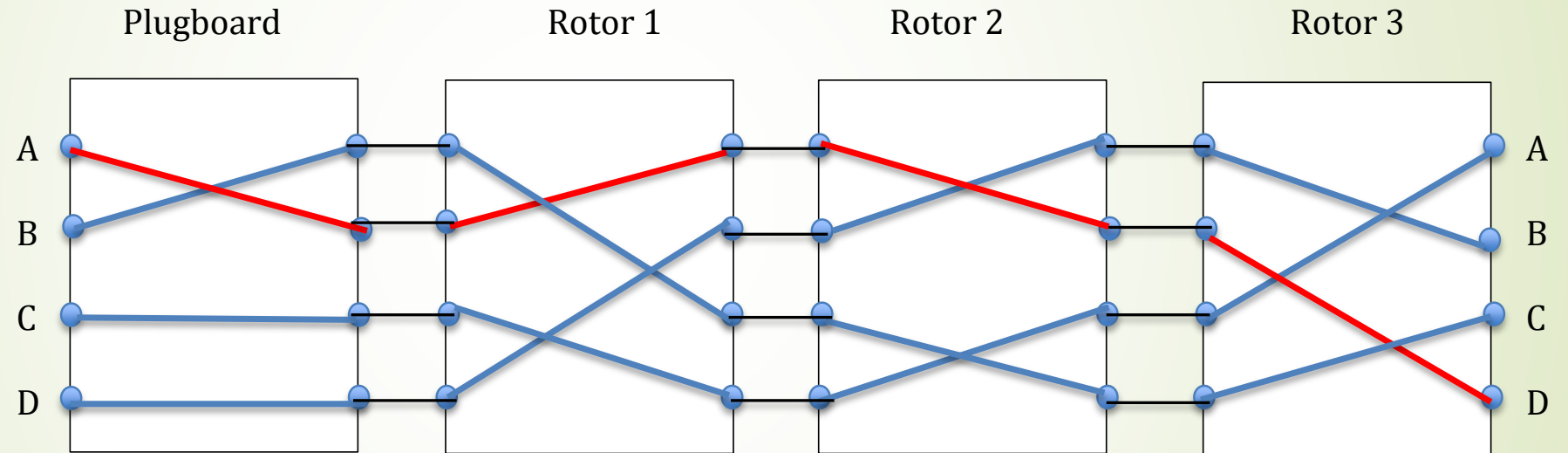
Nombre de possibilités du
plugboard:

$$\frac{26!}{6!10!2^{10}} = 150\,738\,274\,937\,250$$

Nombre de possibilités totale:

$$150\,738\,274\,937\,250 * 1\,054\,560 \\ = 158\,962\,555\,217\,826\,360\,000$$

Fonctionnement Plugboard



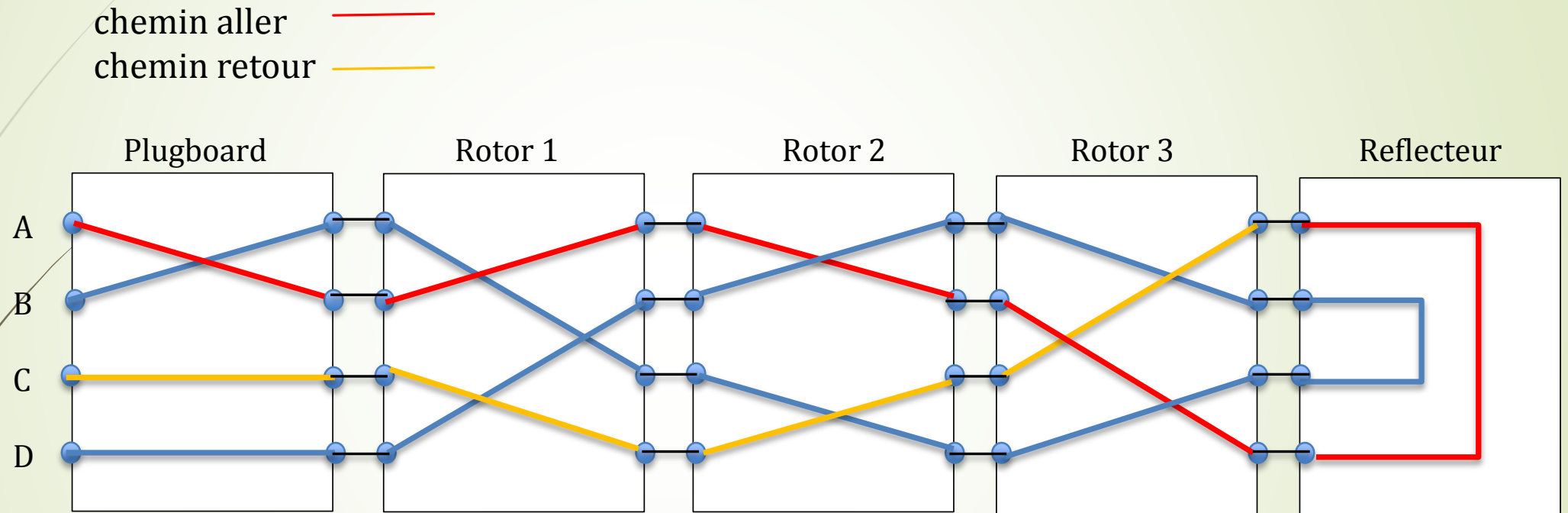
- Le plugboard échange les lettres « A » et « B »

Le réflecteur



- Rend le cryptage réversible
- Plus besoin d'avoir une machine pour crypter et une autre pour décrypter
- « A » cryptée en « B » alors « B » sera décryptée en « A »
- Empêche alors toute lettre d'être cryptée par elle-même
- Ne rajoute pas de possibilités de cryptage

Fonctionnement Réflecteur



- Le réflecteur fait une ultime permutation
- « A » cryptée en « C »
- « C » décryptée en « A »

Organisation du projet

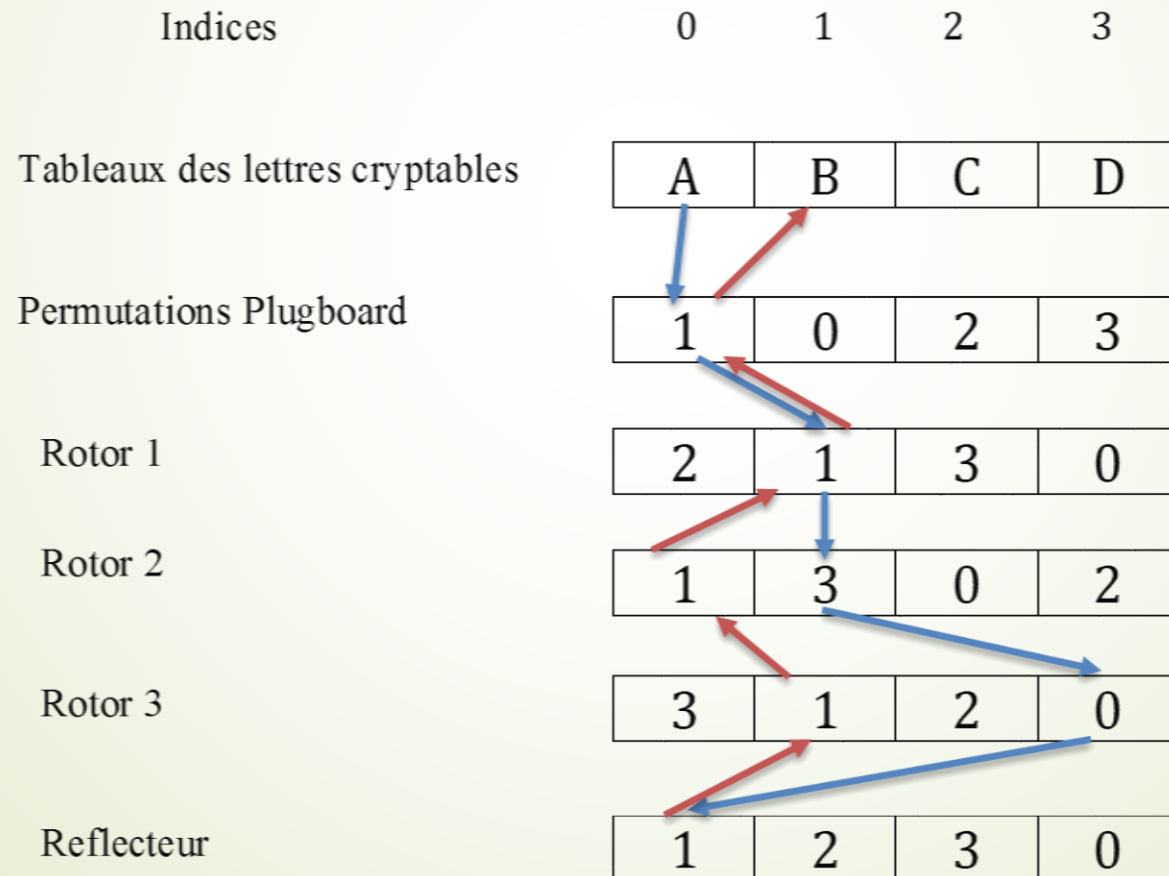
- Compréhension du sujet et du fonctionnement d'Enigma
- Mise en accord sur les différents outils (Java, MVC)
- Création du diagramme de Gantt



- Conception UML
- Répartition des tâches:
 - Model : cœur de l'application
 - View : interface utilisateur
 - Controller : verification des données saisies

Logique de fonctionnement

- Rotors = tableaux d'entrées/sorties





Difficultés rencontrées et solutions retenues

Difficultés	Solutions
Complexité algorithmique: Temps de calcul	Création de tableaux « miroirs »
Interface graphique: Non responsive (adaptative)	Création interface v2.0 avec Java FX et Scene Builder
Méthode « Decrypter » Décryptage long et coûteux	Création d'une classe à part. - méthode de décryptage naïve - Méthode basée sur indice de coïncidence

Dictionnaire



- Dictionnaire : fichier texte de 600 mots les plus courant
- Crypte chaque mot 46^3
- Recherche dans la chaîne cryptée une correspondance
- **Avantages:** traite aussi bien un texte long que court
- **Limites:** très coûteux en temps pour trouver la position des rotors.
Au plus: $46^3 * \text{nombre de mots possibles}$

Indice de coïncidence

Formule mathématique:

$$IC = \sum_{i=1}^{26} \frac{ni(ni-1)}{N(N-1)}$$

- Calcul la probabilité d'apparition de chaque lettre (alphabet de 26 lettres « a » à « z »)
 - Détermine le langage utilisé grâce aux indices de référence (0,072 pour le français)
 - Détermine si c'est une substitution poly-alphabétique ou mono-alphabétique
-
- Décryptage du texte 46^3 fois
 - Calcul de l'indice de coïncidence pour chaque décryptage
 - On garde le texte décrypté où l'indice est le plus proche de 0,072
-
- **Avantages:** très efficace et calcul au plus 46^3 pour trouver la position des rotors. Possibilité de trouver les branchements dans le plugboard de manière moins coûteuse.
 - **Limites:** Indice pas ou peu fiable sur les textes courts.



Bilan

- Nouvelles connaissances en cryptographie
 - Fort intérêt pour la cryptanalyse
 - Application de nos connaissances en informatique et en gestion de projet
 - Envie de continuer le projet
- 