Cours Turing
Semaine 10

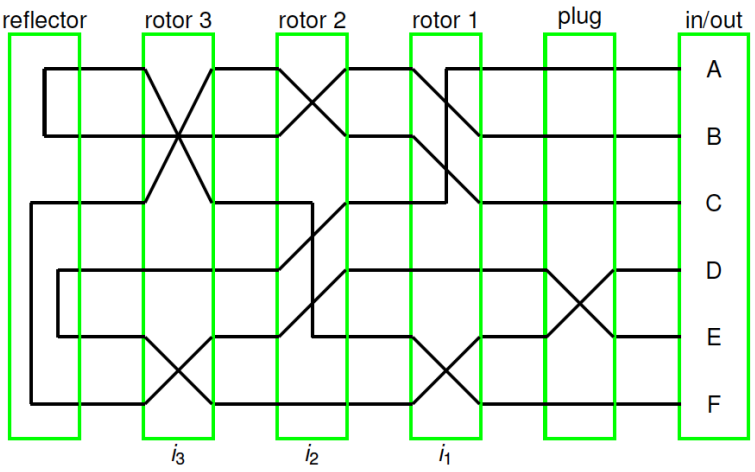
1 Enigma, Alan Turing et la naissance de l'informatique moderne

Le système cryptographique que nous allons passer en revue aujourd'hui est celui de la machine Enigma, inventée à la fin de la première guerre mondiale et utilisée intensivement par l'armée allemande lors de la seconde. Vous en voyez ci-dessous un modèle, exposé au Musée de la Science et de la Technologie de Milan :

Le principe de base de la machine est le suivant : celle-ci est composée (entre autres) de deux claviers, et chaque frappe d'une touche du clavier du bas déclenche l'allumage d'une autre touche du clavier du haut, par un système de connexions électriques que nous allons détailler. Par exemple, pour un réglage donné, en écrivant le mot BONJOUR sur le clavier du bas, on voit s'allumer la séquence de lettres AIOZAPN sur le second clavier.

Rien qu'avec cet exemple, on voit que le chiffrement à l'œuvre ici est plus qu'un simple chiffre par substitution monoalphabétique : la lettre O est une fois remplacée par la lettre I et une autre fois par la lettre A. De même, le A dans le mot chiffré remplace une fois la lettre B et une fois la lettre O. La force du système réside dans l'introduction de 3 rotors, chacun équipé de son propre système de connexions électriques, qui tournent au fur et à mesure que le message est écrit ; ainsi, *à chaque lettre frappée sur le clavier, un autre chiffrement par substitution est utilisé !*

Voyons plus en détail comment le système fonctionne sur le schéma ci-dessous, qui suppose pour simplifier que le clavier ne comporte que les 6 lettres A, B, C, D, E et F.



Lorsqu'une lettre est frappée sur le clavier à droite (disons la lettre A), un signal électrique est envoyé à travers les connexions successives des 3 rotors (laissons tomber pour l'instant l'étape "plug" ; nous y reviendrons), pour finir dans le réflecteur sur la gauche, qui renvoie à son tour le signal à travers les trois rotors, et finit par illuminer la lettre D. Comme déjà mentionné, avant que la prochaine lettre ne soit frappée, les rotors changent de position, ce qui change les connexions ci-dessus.

L'avantage d'introduire un réflecteur est de rendre le système symétrique : ainsi, pour déchiffrer le message reçu, il suffit de placer les rotors dans la même position initiale que lors du chiffrement, et d'écrire le message chiffré sur le clavier du bas pour voir s'allumer le message d'origine sur le clavier du haut. Vous pouvez essayer par vous-même sur le site web ci-dessous, qui offre une simulation du comportement de la machine :

`https://www.101computing.net/enigma-machine-emulator/`

Reste à expliquer un dernier détail, à savoir ce que signifie le "plug" ci-dessus. Ceci fait référence au *tableau de connexions* que vous voyez aussi au bas de l'image de la page précédente : ce tableau permettait de relier deux lettres entre elles par un fil électrique pour créer une permutation de celles-ci. Ainsi, dans le schéma ci-dessus, lorsque la lettre E est frappée au clavier, le

tableau de connexions effectue d'abord une permutation avec la lettre D, puis le signal suit les connexions pour finir par atteindre la lettre C. Sans cette permutation initiale, la lettre E serait chiffrée par la lettre A. En connectant de nombreuses paires de lettres ensemble (ce nombre de paires pouvait aller jusqu'à 10), le chiffrement gagne grandement en complexité, comme nous allons le voir. Bien sûr, il est alors nécessaire d'effectuer les mêmes connexions pour déchiffrer le message. Vous pouvez aussi ajouter de telles connexions sur la machine simulée.

Forces et faiblesses d'Enigma

Voyons tout d'abord quelles sont les forces de ce système. La principale force est le nombre de combinaisons offertes par un tel système !

1. Les rotors : dans la version de base, il existe 5 rotors différents, dont 3 sont choisis pour être placés dans la machine (l'ordre choisi importe), et chaque rotor a autant de positions possibles que de lettres dans l'alphabet, soit 26. Ceci donne lieu en tout à

$$5 \cdot 4 \cdot 3 \cdot 26 \cdot 26 \cdot 26 \simeq \text{un million de combinaisons possibles } (10^6)$$

2. Le tableau de connexions : le fait de pouvoir choisir 10 paires de lettres à permuter dans le tableau de connexions donne lieu quant à lui à un nombre encore plus grand de combinaisons, environ égal à $1,5 \cdot 10^{14}$.

Donc au total, le nombre de combinaisons de la machine est de l'ordre de $1,5 \cdot 10^{20}$; un chiffre absolument faramineux ! Le système de chiffrement semble juste parfait...

Oui, mais... Plusieurs défauts se sont aussi révélés au fil du temps (sans entrer dans trop de détails) :

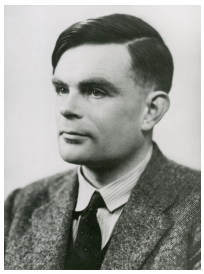
1. Le premier défaut a été de croire que le fonctionnement de la machine resterait caché des Alliés, ce qui ne fut pas le cas. Les Français, avec l'aide des Polonais, interceptent des plans de câblage de la machine dès 1933, et c'est le mathématicien polonais Marian Rejewski qui parvient le premier à reproduire le fonctionnement de la machine.

2. Pour chaque jour, il faut que les unités communiquant entre elles se mettent d'accord sur la position des rotors et du tableau de connexions à adopter : ces positions sont consignées dans de gros cahiers transportés dans chaque unité. Certains de ces cahiers furent aussi interceptés par les Alliés pendant la guerre.

3. Dans les messages échangés entre les unités, de nombreux mots reviennent fréquemment, comme des formules de salutation au début et à la fin des messages, ou des bulletins météo. L'utilisation de ces messages répétés a pu être utilisée pour le décryptage de la machine. De plus, la fatigue et les conditions difficiles d'utilisation ont mené à certaines erreurs de transmission, ce qui a laissé échapper de précieuses informations.

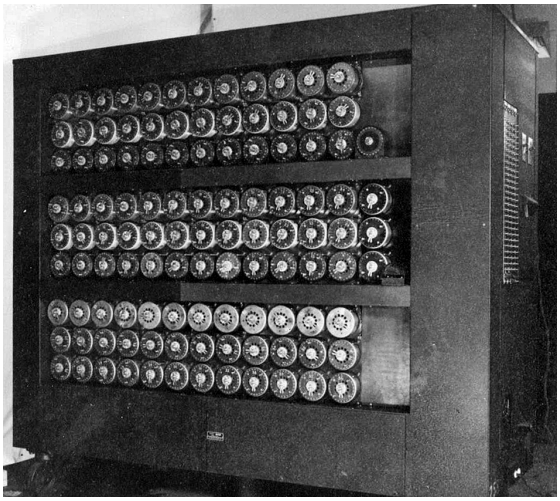
4. L'utilisation du réflecteur, si elle permet un déchiffrage aisé, a aussi pour défaut qu'une lettre n'est jamais chiffrée par elle-même, ce qui réduit (marginalement) le nombre de possibilités.

C'est en utilisant ces diverses faiblesses (entre autres) que l'unité travaillant en secret à *Bletchley Park* en Angleterre, emmenée par *Alan Turing*, a pu venir à bout du chiffrement d'Enigma.



La "bombe" d'Alan Turing

Encore une fois sans entrer dans trop de détails, mentionnons juste ici un élément clé du décryptage d'Enigma : Alan Turing et son équipe, au prix de nombreuses astuces, élaborent une méthode qui permet de faire abstraction du tableau de connexions pour le décryptage. Ainsi, le nombre de combinaisons possibles, au lieu d'être de l'ordre de 10^{20}, est ramené à un million, ce qui reste un très grand nombre de combinaisons, mais pas insurmontable... Pour essayer toutes les combinaisons possibles, Alan Turing construit une machine, qu'il appelle la *bombe*, qui permet de tester toutes ces combinaisons de manière systématique :



Sur la photo ci-dessus, vous voyez que la bombe est composée de plusieurs copies des 3 rotors d'Enigma. En un temps raisonnable, il est ainsi possible de parcourir toutes les combinaisons jusqu'à trouver la bonne.

Avec cette machine, le premier ordinateur est né : le reste appartient à l'histoire...

2 Le principe de Kerckhoff et la cryptographie moderne

Mentionnons finalement une leçon retenue de cet épisode : en 1883, *Auguste Kerckhoffs* avait édicté quelques principes que tout cryptosystème devrait vérifier pour être efficace et fiable :

Le premier principe disait qu'il ne fallait pas que la sécurité du système repose sur le *secret du système en lui-même*. Ce premier principe n'a pas été respecté par Enigma... avec le résultat que l'on sait. Dans la section suivante, nous vous proposons de découvrir un systÃšme de cryptographie moderne qui respecte ce premier principe !

3 Clé à usage unique (1917)

Ce système de chiffrement est aussi connu sur le nom de "masque jetable" ou "chiffre de Vernam" (du nom de son inventeur), ou encore "one-time pad" dans sa version anglaise. Pour le décrire, supposons que la clé secrète $K = (K_0, K_1, K_2, \ldots, K_{n-1})$ en possession d'Alice et Bob soit une séquence de n lettres. Pour chiffrer un message $M = (M_0, M_1, M_2, \ldots, M_{n-1})$ également composé de n lettres, Alice effectue l'opération suivante :

$$C = M \oplus K$$

ce qui est une notation abrégée pour $C_i = M_i \oplus K_i$, $i = 0, \ldots, n-1$, où $A \oplus B$ désigne ici l'*addition modulo 26* des lettres A et B, déjà vue la semaine dernière.

Alice envoie ensuite le message chiffré C à Bob. Celui-ci, également en possession de la clé K, effectue à son tour l'opération suivante sur le message reçu :

$$D = C \ominus K$$

où $A \ominus B$ désigne ici la *soustraction modulo 26*, opérée à nouveau lettre par lettre. Bob retrouve ainsi le message envoyé M, car

$$D = C \ominus K = (M \oplus K) \ominus K = M \oplus (K \ominus K) = M \oplus 0 = M$$

Sécurité de ce système

Alice et Bob ont ainsi trouvé une façon de communiquer secrètement. Pour autant, ce système est-il 100% sûr ? C'est ici qu'il importe de décrire plus précisément comment la clé K doit être générée. Pour que tout fonctionne bien, il faut trois ingrédients :

- chaque lettre K_i de la clé K doit être tirée *uniformément* au hasard parmi les 26 lettres de l'alphaber, c'est-à-dire que les probabilités que $K_i = A$, $K_i = B$, $K_i = C$, etc. valent toutes $\frac{1}{26}$, pour toute valeur de i allant de 0 à $n - 1$;

- les lettres $K_0, K_1, \ldots, K_{n-1}$ doivent être tirées *indépendamment les unes des autres*, comme des dés qu'on lancerait sur une grande table ;

- le tirage aléatoire de la clé K doit être aussi effectué *indépendamment du message M à envoyer*.

Pourquoi toutes ces conditions ? Pour la bonne raison suivante : si maintenant Eve intercepte le message chiffré C, que peut-elle en déduire sur le message d'origine M ?

Pour chaque lettre C_i, la probabilité que $C_i = A$ est la même que la probabilité que $M_i \oplus K_i = A$, ce qui revient à dire que $K_i = M_i$ (vu que la lettre A est représentée par le nombre 0). Or K_i est une lettre de l'alphabet tirée uniformément au hasard. Donc pour toute valeur de M_i, la probabilité que $K_i = M_i$ vaut $\frac{1}{26}$. Et donc la probabilité que $C_i = A$ vaut aussi $\frac{1}{26}$.

Notez ensuite qu'il en va de même pour la probabilité que $C_i =$ n'importe quelle autre lettre : toutes ces probabilités seront égales à $\frac{1}{26}$. Ainsi, pour Eve, qui ne connaît ni M, ni K, la séquence de lettres C apparaît donc comme une séquence de lettres complètement aléatoire ! Elle ne peut donc strictement rien déduire de cette séquence sur le message M d'origine : le système est sûr à 100%.

Quiz

- Pourquoi une attaque par force brute ne permettrait-elle pas de casser le système de clé à usage unique (en supposant donc qu'Eve dispose de la puissance de calcul nécessaire pour essayer toutes les clés possibles) ?

- (Question reliée) Vu que la clé K est tirée au hasard, il y a une probabilité (petite, certes, mais non-nulle) que toutes les lettres de K valent exactement A, donc 0. Dans ce cas, le message chiffré C est égal au message d'origine M : est-ce grave ?

Une clé malheureusement non réutilisable (d'où son nom...)

Ceci dit, le plus gros défaut de ce système est le suivant : supposons qu'Alice désire réutiliser la clé K pour envoyer deux messages M_1 et M_2, chacun de même longueur n que la clé K. Ainsi, Alice envoie les deux messages chiffrés C_1 et C_2 suivants :

$$C_1 = M_1 \oplus K \qquad\qquad C_2 = M_2 \oplus K$$

Individuellement, chaque message est bien protégé, comme nous venons de le voir. Mais si Eve intercepte les deux messages chiffrés C_1 et C_2, rien ne l'empêche d'effectuer une soustraction modulo 26 de ceux-ci, pour obtenir :

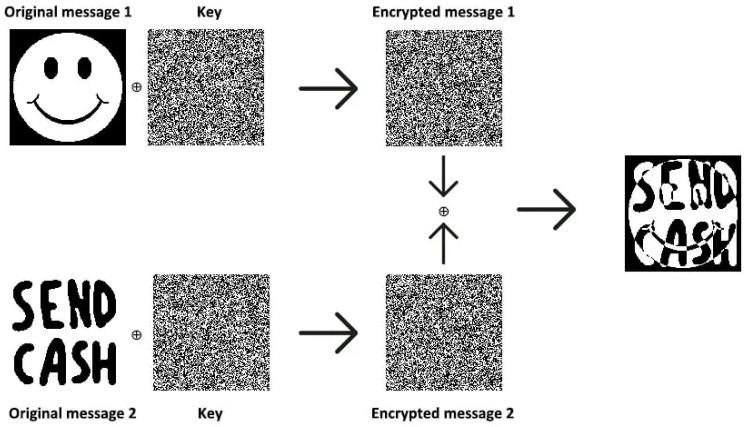
$$C_1 \ominus C_2 = (M_1 \oplus K) \ominus (M_2 \oplus K) = M_1 \oplus K \ominus M_2 \ominus K = (M_1 \ominus M_2) \oplus (K \ominus K)$$

donc, finalement :

$$C_1 \ominus C_2 = M_1 \ominus M_2 \oplus 0 = M_1 \ominus M_2$$

ce qui veut dire que la clé K a maintenant totalement disparu ! Certes, il peut être difficile pour Eve de déduire quelque chose à propos des messages M_1 et M_2 à partir du seul $M_1 \ominus M_2$, mais la sécurité du chiffrement n'est plus du tout garantie !

Le problème est bien illustré sur le schéma ci-dessous : si deux images noir/blanc sont chiffrées avec une même clé K (note : on utilise ici des bits 0/1 et des opérations modulo 2 plutôt que des lettres et des opérations modulo 26), alors une soustracton des deux images chiffrées donne l'image à droite :



Cette impossibilité de réutiliser de la clé est clairement un très grave défaut du système ! Si Alice désirait chiffrer un gros fichier de données avec ce système, elle devrait produire une clé de même taille, ce qui en soi est déjà coûteux (car produire de longues séquences de bits aléatoires ne s'improvise pas), mais il y a bien pire : il faudrait d'abord partager la clé secrètement avec Bob avant de communiquer. Or si une telle occasion de communiquer secrètement se présentait, pourquoi ne pas utiliser celle-ci pour communiquer directement le fichier ?

La semaine prochaine, nous verrons un autre système de chiffrement qui permet de réutiliser la clé K plusieurs fois.