Concours Alkindi 2016–2017
C'est de la crypto !

Plus de 47 000 élèves ont participé au premier tour de la 2e édition du Concours Alkindi. Pendant 45 minutes, ils ont été confronté à une série de sept petits défis interactifs sur ordinateur. Sous leur aspect très ludique, ces exercices font appel à des raisonnements logiques, mathématiques et informatiques poussés. Mais derrière ces exercices se cachent également un problème scientifique, une idée importante des sciences de l'information, une réalité historique et un problème concret. Ce document explicite le lien entre les exercices que les élèves ont résolu et la cryptographie.

1 Antidote

Lorsqu'on construit un système de chiffrement, même si le système théorique est sécurisé, il reste des attaques qu'on n'avait pas prévues. Par exemple, il est possible de retrouver le code d'accès de votre téléphone portable en observant les tâches de doigts sur l'écran. De même, dans certains ordinateurs, on peut retrouver des informations sur le mot de passe en mesurant le temps de calcul :

Imaginez le système suivant : le mot de passe est une suite de 4 chiffres. Lorsqu'on vous entrez une tentative de mot de passe, l'ordinateur compare votre premier chiffre avec le premier chiffre du secret. Cette opération prend une seconde. Si votre premier chiffre est incorrect, il s'arrête et répond "Non" sinon il passe au chiffre suivant : il compare le 2e chiffre avec le 2e chiffre du secret, cela prend encore une seconde. Si le 2e chiffre est incorrect, il répond "Non" sinon il continue. Il continue ainsi jusqu'au dernier chiffre. Si tout est juste, il vous laisse accéder.

Prenons un exemple :

— 3125 : l'ordinateur répond "Non" en 1 seconde. Le mot de passe ne commence donc pas par 3.

— 4951 : l'ordinateur répond "Non" en 2 secondes. Cela veut dire que le premier chiffre est correct. Le mot de passe commence par 4. Mais le 2e chiffre n'est pas 9. Il faut deviner la suite.

— 4203 : l'ordinateur répond "Non" en 3 secondes. Cela veut dire que les deux premiers chiffres sont corrects. Le mot de passe commence par 42. Le 3e chiffre n'est pas un 0.

— 4265 : l'ordinateur répond "Non" en 4 secondes. Donc les 3 premiers chiffres sont corrects. Reste à trouver le dernier.
— Finalement on arrive à trouver 4261 qui est le mot de passe secret.

Plus on essaie un mot de passe proche de la solution, plus l'ordinateur met du temps à répondre. On peut donc facilement retrouver le mot de passe avec quelques essais. On peut aussi faire des raisonnements similaires en écoutant le bruit de l'ordinateur ou en observant la consommation d'électricité. On appelle cela des attaques par canaux cachés.

Le principe général est donc le suivant : on peut observer plusieurs valeurs (temps, bruit, consommation, ...) ce qui nous indique si on est proche ou pas de la solution. Il faut les utiliser pour retrouver le secret. C'est exactement ce qu'on fait dans l'exercice !

2 Machines

Les ordinateurs stockent les données sous la forme de suites de 0 et de 1. Dans certains systèmes de chiffrement, les calculs sont effectués directement au niveau électronique, comme par exemple sur les cartes bancaires ou sur disques durs chiffrés. Si l'on veut casser ces protections, il faut essayer de comprendre comment fonctionne le circuit électronique. Pour cela, on peut seulement faire des mesures à l'entrée et à la sortie du circuit. On sait que ces circuits sont composés en assemblant une série de petites boîtes, appelées "portes logiques". Il faut donc retrouver quelles sont les portes logiques utilisées, et dans quel ordre, à partir des données partielles. On appelle cela de la rétro-ingénierie. Cela permet ensuite d'imiter le système, et donc de se faire passer pour quelqu'un d'autre.

3 Points et tirets

Le prédécesseur du téléphone a été le télégraphe qui était utilisé pour envoyer des phrases courtes à grande distance. Cet appareil ne permet de transmettre que deux choses : des bruits courts (appelés points) et des bruits longs (appelés traits). Il faut donc remplacer chaque phrase par une séquence de traits et de points. On pourrait choisir de représenter toutes les lettres par un code de même taille (par exemple six symboles). Mais ce n'est pas optimal : certaines lettres, comme "E", sont utilisées très souvent alors que d'autres très rarement. En assignant un code court aux lettres fréquentes on réduit la taille des messages même si cela nous oblige à assigner des codes plus longs aux lettres rares. Le code de Morse a été choisi parmi plusieurs tels codes parce qu'il minimisait la longueur des messages.

La version de l'exercice de quatre étoiles propose une amélioration : comment transmettre un tel code sans les espaces ? En effet les opérateurs du télégraphe, qui utilisait le code Morse, faisaient une petite pause après

chaque lettre pour marquer la fin et la personne qui recevait laissait un petit espace après chaque groupe de points et traits. Dans le numérique on utilise des codes similaires au code de Morse où chaque lettre est codée par des 0s et des 1s. Il n'y a donc plus la possibilité de laisser des espaces vides entre les codages des lettres d'un même mot. Les travaux de Huffman ont montré qu'on peut trouver des codes qui se déchiffrent bien. Il suffit qu'aucun mot ne soit préfixe d'un autre. Huffman explique même comment créer de tels codes.

4 Boules

Lors de la transmission de données, quelques erreurs de trahsmission peuvebt se glisser dans le messqge. Si des erreurs se glissent dans un texte français, vous pouvez toujours le comprendre. Vous venez bien de corriger la première phrase ! En revanche, si vous faites une erreur en notant un numéro de téléphone, vous allez tomber sur un mauvais destinataire, mais vous ne pourrez pas corriger et retrouver le bon numéro. Pourquoi ?

Dans le premier cas, vous savez que le mot *"trahsmission"* n'existe pas dans le dictionnaire. De plus, vous connaissez un mot très proche qui existe bien dans le dictionnaire, le mot *"transmission"*. Il est aisé de faire la correction. Dans un numéro de téléphone, on ne peut pas distinguer simplement un vrai numéro d'un numéro avec erreur.

Lorsque l'ordinateur transmet des données, il envoie une suite de 0 et de 1, appelés des bits. Comme pour le numéro de téléphone, on ne peut pas distinguer un vrai message d'un message erroné. Sauf si l'on ajoute des règles ! Une façon de détecter les erreurs est la suivante : à chaque fois que l'ordinateur de Bob doit envoyer un groupe de 3 bits à Alice, il ajoute un 4^{e} bit, de sorte que la somme des bits soit paire. Par exemple pour envoyer le message "010", Bob envoie "0101", et pour envoyer "110" il envoie "1100". Si jamais Alice reçoit "1011", elle sait qu'il y a une erreur car la somme des chiffres est impaire. Elle ne sait pas où est l'erreur, mais elle peut demander à Bob de renvoyer son message.

Ce que vient de faire Bob, c'est fixer une règle que doivent suivre les messages (comme les règles d'orthographe). Si l'on appelle a, b et c les trois bits du message, cela revient à dire qu'on va transformer abc en $abcd$ où $d = a + b + c$. C'est exactement la version facile de l'exercice. Alors, si un des chiffres du message est modifié, le destinataire pourra le détecter.

Et si on voulait pouvoir corriger l'erreur ? C'est possible. Par exemple, à la place d'envoyer "0", l'ordinateur de Bob peut envoyer "000". Et à la place d'envoyer "1", il peut envoyer "111". Pour envoyer le message "010", l'ordinateur de Bob va donc envoyer "000111000". Maintenant, si jamais Alice reçoit le message "010111000", elle sait qu'il est erroné car il ne suit pas le code (c'est comme si ce mot n'était pas dans le dictionnaire de mots

autorisés). Mais Alice sait qu'il y a un mot proche qui existe bien, le mot "000111000". Elle peut déduire que le message envoyé par Bob était "010". Alice a corrigé le message ! C'est possible car on a rajouté de l'information (on dit qu'il y a de la redondance) pour pouvoir corriger les erreurs : on parle de codes correcteurs.

L'inconvénient de ce code là est qu'il rallonge beaucoup la taille (on multiplie la taille des messages par trois pour corriger une erreur), cela ralentit la transmission. Il faut donc trouver des codes qui permettent de corriger beaucoup d'erreurs sans trop augmenter la taille du message. Par exemple, le code de Hamming transforme un mot de 4 bits en un mot de 7 bits, et permet de corriger une erreur. C'est plus efficace que l'exemple précédent : ici on n'a même pas doublé la longueur du message. On a rajouté trois contraintes. Cela correspond exactement au cas difficile de l'exercice.

Les codes correcteurs sont présents dans tous les systèmes de communication, que ce soit dans les téléphones portables pour passer des appels ou dans les ordinateurs pour stocker des données sur CD ou disque dur. Bien que la plupart des codes correcteurs ne soient pas secrets, on peut créer des codes secrets à l'aide des codes correcteurs.

5 Météo

Cet exercice étudie un mode de chiffrement très classique appelé chiffrement par substitution : chaque lettre de l'alphabet est remplacée par une autre. A priori, cela a l'air difficile à retrouver car il y a beaucoup de possibilités. Mais si l'on devine certains mots du message, la répétition des lettres permet de retrouver leur position, et alors on peut facilement décrypter le reste du message !

D'autres méthodes de chiffrement plus complexes sont vulnérables à des attaques similaires. Par exemple, lorsqu'Alan Turing a cryptanalysé la machine Enigma pendant la seconde guerre mondiale, il a utilisé la même idée. En particulier, il savait que tous les matins, les Allemands envoyaient le bulletin météo. Il suffisait donc de chercher la position du mot *METEO*. Ce mot était particulièrement pratique, car en allemand, ce mot s'écrit *WETTER*, donc il contient deux lettre en double. Dans la version difficile de l'exercice, on voit bien qu'il est plus facile de trouver la position des mots comme *TEXTE* où certaines lettres se répètent.

6 Aliceflix

Aujourd'hui, on récolte beaucoup de données personnelles, que l'on stocke dans des bases de données. Lorsqu'on rend publiques ces données, leur analyse permet de faire beaucoup de progrès. Par exemple en analysant le pourcentage de réussite d'un traitement médical partagées par un hôpital, les

médecins peuvent savoir si un traitement est efficace, et trouver de nouveaux remèdes. Mais il faut que ces données soient protégées : on ne doit pas pouvoir déterminer quelle personne est touché par quelle maladie.

La protection des données personnelles est exigée par la loi. On peut croire que la solution est de rendre publiques les données tout en les anonymisant. Malheureusement les chercheurs ont trouvé des failles dans beaucoup de techniques actuelles d'anonymisation.

En 2007, Netflix a organisé un concours pour que les informaticiens proposent des algorithmes pour proposer un film qui va vous plaire, en fonction des films que vous avez déjà vus. Pour pouvoir tester les algorithmes, Netflix a rendu publique une base de données de 500 000 utilisateurs, indiquant leurs préférences de films. Pour préserver l'anonymat, les noms étaient remplacés par des numéros. Une étude menée par deux chercheurs américains a montré qu'il était quand même possible d'identifier une grande partie des utilisateurs, avec l'aide d'autres données disponibles sur internet. On pouvait même en déduire des informations personnelles, par exemple les préférences politiques des utilisateurs. Depuis, ce type d'anonymisation n'est plus considéré comme sécurisé.

7 Vigenère

Cet exercice étudie le chiffrement de Vigenère. Cette méthode a été beaucoup utilisée entre le XVI^{e} et le XIX^{e} siècle. Ce n'est qu'en 1863 que le cryptologue allemand Kasiski a publié une méthode de cryptanalyse. Celle-ci s'appuie en partie sur la répétition de certains mots, qui aide à repérer la taille de la clé, comme dans la version difficile de l'exercice. Aujourd'hui, avec la puissance de calcul des ordinateurs, cette méthode de chiffrement n'est plus sécurisée.