**DigiCode**

**Pascal Lafourcade et Malika More**

L’objectif de cette activité est de faire comprendre le concept d’attaque par canal auxiliaire (*side channel,* en anglais). L’activité met en situation les élèves qui doivent trouver le code d’entrée d’une porte qui s’ouvre si le code secret à 4 chiffres est correctement saisi sur un digicode. Ce digicode qui possède dix touches portant les chiffres de 0 à 9 et deux voyants lumineux : un rouge et un vert. Quand un chiffre correct est saisi, le voyant vert s'allume, mais sitôt qu'un chiffre incorrect est tapé, le voyant vert s'éteint, le rouge s'allume et il faut recommencer à taper le code depuis le début.

Par exemple, si le code est 1234 et qu'Alice tape par inadvertance 124, les couleurs des voyants sont les suivantes :

* 1 => le voyant vert s'allume
* 2 => le voyant vert reste allumé
* 4 => le voyant vert s'éteint, le rouge s'allume et le code est annulé

Il est alors demandé aux élèves de calculer le nombre maximal d'essais qu’il faut effectuer pour ouvrir la porte ?

Pour bien faire comprendre le fonctionnement du digicode, il est possible de mettre les étudiants en situation en leur faisant jouer un jeu de rôle. Un élève choisit un code à 4 chiffres qu’il mémorise. Il joue alors le rôle du digicode et un autre élève essaie donc de le deviner en indiquant les touche qu’il saisit sur le digicode. En effet, à chaque saisie d’un chiffre sur le clavier l’élève qui joue le rôle du digicode indique seulement la couleur de la lumière.

Les élèves trouvent rapidement le digicode. Ils s’aperçoivent que le nombre d’essais est loin d’être 10 000, ce qui correspond aux nombres possibles de combinaisons pour un digicode de 4 chiffres de 0 à 9. Une fois que les élèves ont compris le fonctionnement des lumières vertes et rouges, il envisage une solution plus efficace. Ils s’aperçoivent que pour rechercher le premier chiffre du code, il suffit de tous les essayer un par un, jusqu'à ce que la lumière verte s'allume et ainsi être sûr de l'avoir trouvé. Cela prend au pire 10 essais. Connaissant le premier chiffre, il est alors possible de rechercher le deuxième en tapant deux chiffres à la fois jusqu'à ce que la lumière verte s'allume. En suivant cette stratégie, il suffit de 10 autres essais au maximum. Ainsi, au total, en utilisant les lumières, il faut faire 10 + 10 + 10 + 10 = 40 essais dans le pire des cas pour ouvrir la porte.

Les élèves arrivent souvent à trouver cette stratégie et ont ainsi compris le principe des attaques par canaux auxiliaires. En effet les informations des couleurs font fuir des informations qui permettent de retrouver en quelques essais le code secret au lieu des 10 000 essais par brute force.

Il est possible de faire encore un peu mieux ! Pour cela il faut juste 9 essais au maximum pour connaître le premier chiffre. En effet, si la lumière est encore rouge au neuvième essai, alors il ne reste plus qu'une solution possible et il est inutile de faire le dernier essai. Le même raisonnement s'applique pour le deuxième et le troisième chiffres. En revanche, pour le dernier chiffre, si 9 essais au maximum suffisent toujours pour le découvrir, il faut bien ouvrir la porte, et donc taper le code une fois qu'il est découvert.

Cette activité permet aux élèves d’avoir des interactions et de comprendre le principe des attaques par canaux cachés. Ces attaques existent vraiment sur les systèmes informatiques. Par exemple, le code suivant teste si un code PIN stocké dans le tableau pinPresente est égal au tableau pinCarte qui contient le code PIN secret de l’utilisateur.

for ( i = 0 ; i <= 7; i++)

 if ( pinCarte[i] != pinPresente[i] )

 return false;

return true ;

En effet, ce programme se comporte exactement comme le digicode avec les lumières de couleur rouge et verte. Pour ce programme c’est le temps d’exécution qui fait office de canal auxiliaire. Les tableaux sont de taille 8 et pour vérifier la validité du code PIN il faut que tous les chiffres soient valides et donc effectuer les 8 tours de la boucle for. Si un chiffre est différent alors le programme va quitter la boucle plus tôt. Ainsi en mesurant le temps d’exécution du programme il est possible de déduire quel est le premier chiffre en au maximum 8 essais et donc en 8\*256 essais il est possible de trouver tous les chiffres.

Pour éviter cela une possibilité est d’utiliser le programme suivant :

boolean test = true ;

for ( i = 0 ; i <= 7; i++)

 test = test && ( pinCarte[i] == pinPresente[i]);

return test ;

Dans ce programme, il faut effectuer tous les tours de la boucle pour obtenir le résultat, ainsi il n’y a aucune différence quel que soit la valeur du code PIN saisi.

Les attaques par canaux auxiliaires ou canaux cachés peuvent être fait en mesurant le temps d’exécution d’un programme, mais aussi en mesurant la consommation électrique ou encore les émissions électromagnétiques. D’autres attaques existent en observant par exemple les défauts des mémoires caches. Ces attaques ont été introduites par Paul Kocher dans les années 1990 dans son papier originel, où il a montré qu’il était possible de récupérer les récupérer les clés secrètes des chiffrements DES et RSA. Pour le chiffrement RSA, l’utilisation de l’exponentielle rapide permet de chiffrer et déchiffrer plus vite, mais permet une attaque par mesure de temps d’exécution pour en déduire la clé.