**Chiffrement de Hill**

Diapo 1 :

Bonjour à tous, notre projet est le Chiffrement de Hill.

Diapo 2 :

D’abord, nous allons faire l’introduction de Chiffrement de Hill, son modèle utilisé, ses propriétés et ses fonctions.

Ensuite, nous allons vous montrer nos codes et nous allons vous donner l’explication pour nos codes aussi. Le Chiffrement de Hill est composé par 3 parties : Chiffrement, Déchiffrement et AttackHill.

Enfin, nous allons conclure notre projet, les difficulités que nous avons rencontrées et ce que nous avons appris.

**Diapo 3 :**

**Le chiffrement de Hill est un modèle simple d'extension du** [**chiffrement affine**](http://fr.wikipedia.org/wiki/Chiffre_affine) **à un** [**bloc**](http://fr.wikipedia.org/wiki/Chiffrement_par_bloc)**.**

**Ce système utilise les propriétés de l'**[**arithmétique modulaire**](http://fr.wikipedia.org/wiki/Arithm%C3%A9tique_modulaire) **et des** [**matrices**](http://fr.wikipedia.org/wiki/Matrice)**.**

**Il s'agit de chiffrer le message en substituant les lettres du message, non plus lettre à lettre, mais par groupe de lettres.**

**Diapo 4 :**

**Nous cherchons à chiffrer des message en utilisant une matrice dont le déterminant est premier avec 26, donc on prendra un exemple : la matriceA.**

**A=\begin{pmatrix} 3 & 5 \\ 6 & 17\end{pmatrix}**

**Son déterminant est 21. Comme 21 est premier avec 26, il possède un inverse dans \Z/26\Z.**

**Donc cette matrice est suffit pour faire le chiffrement, le déchiffrement et l’AttackHill.**

Diapo 5 :

Pour le Chiffrement :

D’abord, nous avons crée un tableau avec ces codes, nous avons mis les lettres dans le tableau selon leur rang.

Diapo 6 :

Nous avons calculer le déterminant de matrice, nous avons fait modulo pour le déterminant et nous avons calculé le pgcd de déterminant de matrice modulo 26 et 26.

Diapo 7 :

D’abord, nous avons vérifié que le pgcd de déterminant soit première avec le 26.

Nous avons remplacé chaque lettre par son rang à l'aide du tableau que nous avons construit (Les chiffres obtenues permettent de créer une matrice).

Ensuite,

blocksize permet de calculer le nombre de colonne de matrice

numblocks permet de calculer le nombre de blocks, c’est à dire le nombre de ligne de matrice

Nous avons construit la matrice pour cette message.

Nous avons fait modulo 26 pour le produit de la matrice fois la clé (matrice A)

ct permet de convertir la matrice en vecteur

Enfin, nous avons transformé les chiffres dans le vecteur en lettres

**Diapo 8 :**

**Pour tester, nous avons entré la matrice A et le message à coder, donc nous avons obtenu le résultat ABCD.**

**Diapo 9 :**

**Une information codé par une clé A (inversible) est décodé tout simplement en écrivant :**

**X = Y \* A^(-1)**

**Donc en maple, nous avons utilisé ce code :**

**le commande map permet d’appliquer une procédure à chaque opérande d'une expression**

**Diapo 10 :**

**Pour tester, nous avons entré la matrice A (clé) et le message à décoder, c’est le résultat que nous avons obtenu dans le code “Chiffrement”. Enfin, nous avons obtenu le message “ABCD”, donc nos code fonctionne correctement.**

Diapo 11 :

Pour AttackHill, nous avons utilisé le même principe comme le déchiffrement, matrice de message crypté = matrice de message non cryté fois la clé, nous avons obtenu deux équations, donc nous avons entré deux groupe de messges crypté et non crypté, nous pouvions obtenu quatre équations, il y a quatre unconnus, donc nous pouvons obtenir les résutats.

Diapo 12

Pour le tester, nous avons entré deux groupes de messages, nous avons obtenu le bon résultat.

Diapo 13

Les difficultés rencontré sont de se rappeler le cours sur les matrice, comment elle fonctionne et comment coder ces matrices sur maple.

Nous avons appris donc comment fonctionne le chiffrement de hill et le coder sur maple avec différentes fonctions.

nous sommes allé plus loin dans la programmation de maple avec des fonction que nous avons découvert tout au long du projet.