**Rapport de projet M1**

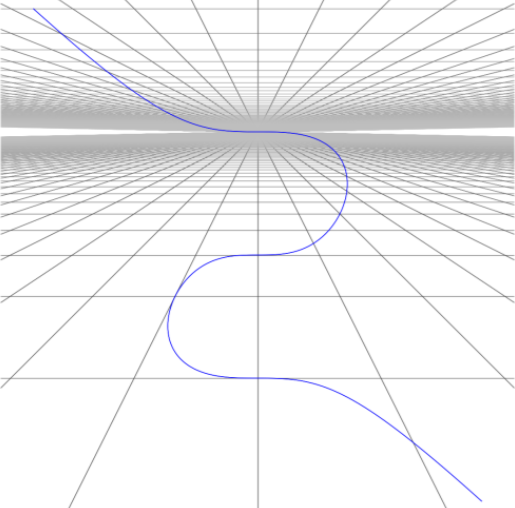
**Projet n°40 : Application pédagogique sur les courbes elliptiques**

Intervenant : Gabriel CHENEVERT.

Etudiants : François BERNARD

Simon CORDIER

Fabien DELATTRE



**Sommaire**

* Présentation de la mission
* Les courbes elliptiques et la cryptographie
* Interface graphique
* Calculs sur les courbes
* Difficultés rencontrées
* Conclusion

**Présentation de la mission**

Nous avions pour objectif de créer une application pédagogique illustrant les calculs sur les courbes elliptiques et leur utilité dans le monde de la cryptographie.

Cette application devait être pédagogique, c’est-à-dire pouvoir expliquer concrètement comment utiliser les courbes elliptiques et montrer la vitesse d’exécution des différentes opérations.

Ces courbes n’étant pas très intuitives, nous devions produire quelque chose de compréhensible pour des personnes dont ce n’est pas le domaine et faire en sorte qu’ils puissent les utiliser.

**Les courbes elliptiques et la cryptographie**

La cryptographie sur les courbes elliptiques (ECC : Elliptic Curve Cryptography en anglais) est apparue en 1985, proposée par Victor S. Miller et Neal Koblitz en 1987.

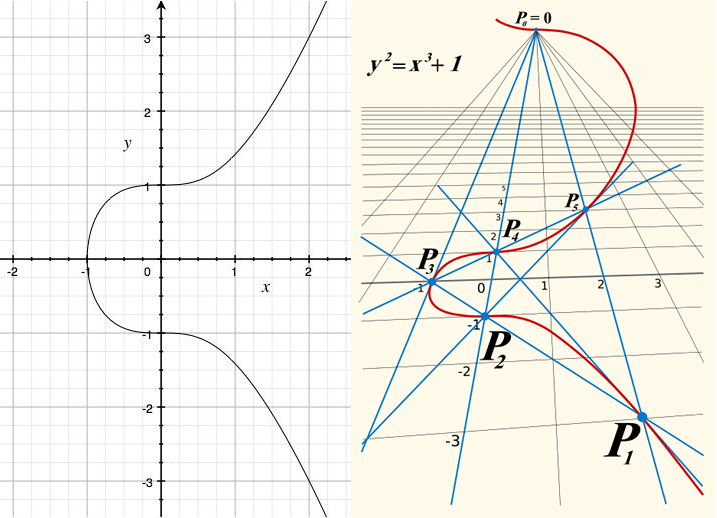
Elle a été proposée comme une amélioration du protocole d’échange de clés de Diffie-Hellmann, apportant une rapidité d’exécution 20% supérieure. Ces échanges sur canaux non sécurisés sont également appelés chiffrement asymétrique.

Pour différencier rapidement la nouvelle méthode, il a été vérifié qu’une clé de 200 bits pour des calculs sur les courbes elliptiques est plus sûre qu’une clé de 1024 bits pour le chiffrement RSA. La rapidité n’est donc pas comparable.

Bien que brevetées de nombreuses fois, la cryptographie sur ces courbes est une découverte récente, encore assez complexe mais de plus en plus utilisée. Résoudre son problème revient à résoudre le problème du logarithme discret, certes contournable, mais qui demande énormément de temps avec les technologies actuelles.

Une courbe elliptique possède une équation du type . Elle possède une formule d’addition de 2 points appartenant à la courbe, ainsi qu’un point neutre étant le point à l’infini noté O.

Pour deux points P et Q sur la courbe, si l’on trace la droite (PQ), on coupe la courbe en un troisième point. La symétrie de ce troisième point par rapport à l’axe des abscisses est le résultat P+Q. Dans le cas où P et Q ont la même abscisse, alors le troisième point est le point à l’infini (point O).



L’utilisation d’un modulo dans la formule d’une courbe elliptique () rend la résolution du problème bien plus difficile car il s’agit alors de résoudre le logarithme discret.

Certains algorithmes comme les attaques par force brute permette de résoudre ce problème mais lorsque que l’on prend de très grands entiers, la résolution peut alors prendre plusieurs années.

L’utilisation des courbes elliptiques s’effectue comme suit :

Alice et Bob choisissent tous les deux un entier suffisamment grand et qu’ils gardent **secrets**.

Ils se mettent d’accord et peuvent donner publiquement les coefficients a et b ainsi que le module p de la courbe. De même, ils prennent en commun un point de la courbe P, calculent chacun de leur côté le résultat de et et se les échangent.

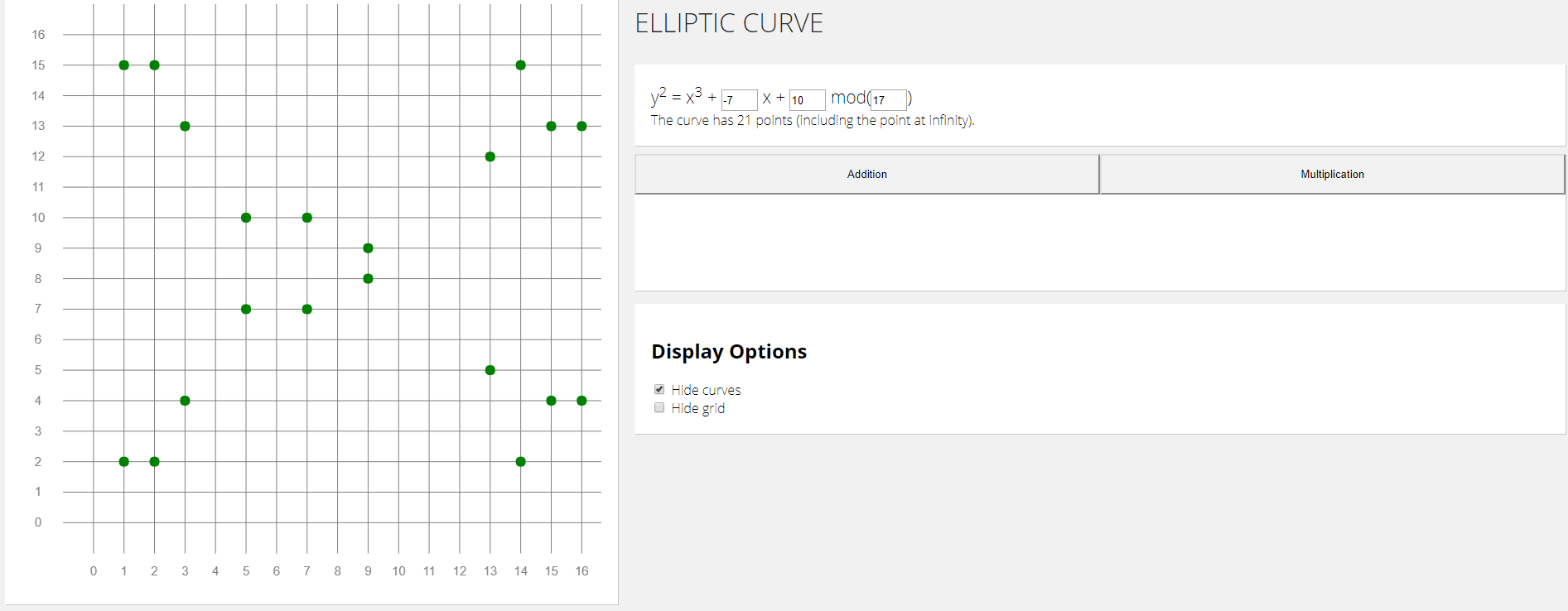
Une fois tout ceci réuni, si le but était un **échange de clés**, ils peuvent tous les deux calculer qui deviendra leur clé secrète commune.

Pour **envoyer un message chiffré**, celui qui veut le transmettre (disons Alice) choisit un entier n qu’il garde secret, puis envoie le résultat de n.P et à l’autre. La personne en face (Bob) peut alors calculer (avec n.P) et le soustraire à pour récupérer le message M caché.

Le seul moyen envisageable de trouver M est de trouver , ce qui demande un temps incalculable.

Si les entiers sont choisis secrètement et rigoureusement, cette méthode de chiffrement devient invulnérable.

**Interface graphique**



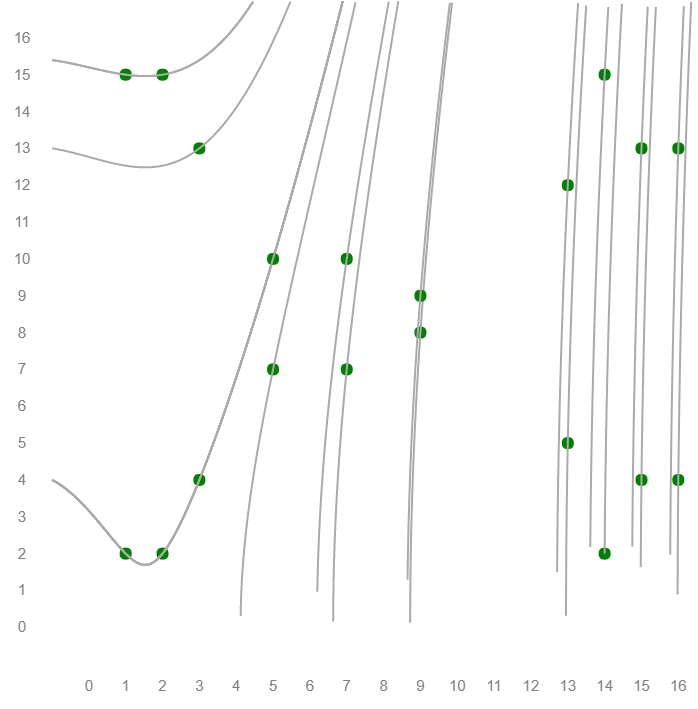
*INTERACTION AVEC LA PAGE HTML*

Entrer la fonction :



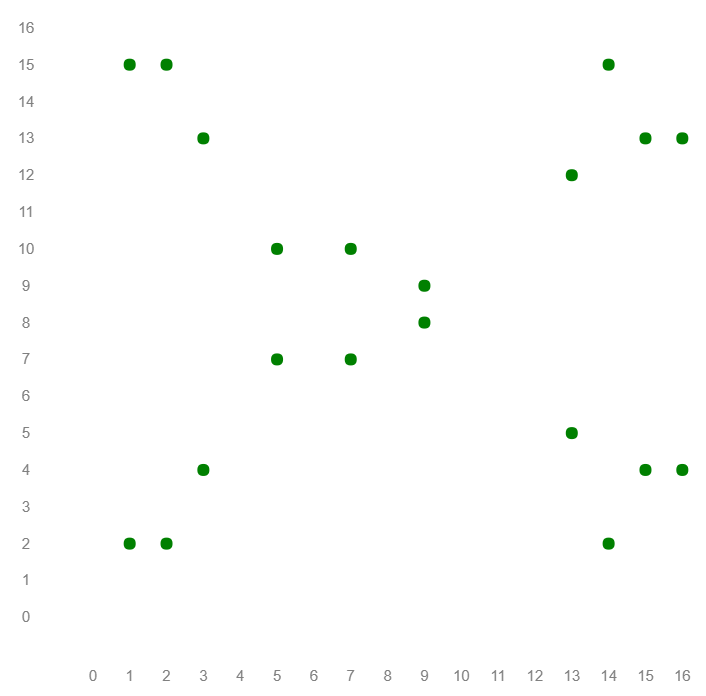
Vous pouvez choisir la valeur du coefficient de X (a), de la constante (b) et le module (m). (Exemple ici : a = -7 ; b=10 ; m=17)

Option d’affichage :



Cela permet :

-De cacher (ou non) la grille.



-De cacher (ou non) la courbe.

*LISTE DES FICHIERS EN LIEN AVEC L’INTERFACE GRAPHIQUE :*

Le fichier HTML : **modulo.html**

Le fichier CSS : **Style.css**

Les fichiers JS :

-**Canvas.js** : Contient la classe Canvas, il permet d’obtenir la grille.

-**Point.js** : Contient la classe Point, il permet de créer des points.

-**Menu.js** : Différencier le menu Addition et Multiplication.

-**Elliptic.js**:

constructor(a,b,m,canvas) : Créer une courbe elliptique ;

drawModulo() : Tracer la courbe elliptique ;

click(point) : Sélectionner un point sur la courbe elliptique  ;

drawLine() : Tracer la droite passant par les deux points sélectionnés ;

animationLine(that, start, end, progress) : complément de la fonction drawLine() , anime le tracé de la courbe.

**Calculs sur les courbes**

Nous allons maintenant voir ce qui se cache derrière cette interface graphique.

Voici les différentes fonctions de calcul que l’on peut trouver dans notre fichier Elliptic.js. Ces fonctions sont les fonctions utilisées dans la cryptographie sur courbes elliptiques.

calc(x) : Retourne le nombre de solution d’une courbe elliptique pour un x donné ;

sum(p1, p2) : Calcule la somme de deux points de la courbe elliptique et retourner les coordonnées du point résultant ;

mult(p, n) : Multiplie un point n fois en utilisant la fonction sum(p, p) ;

div(np, p) : Cherche n connaissant le produit n.P ainsi que le point P. (Calcul très long si les nombres sont grands)