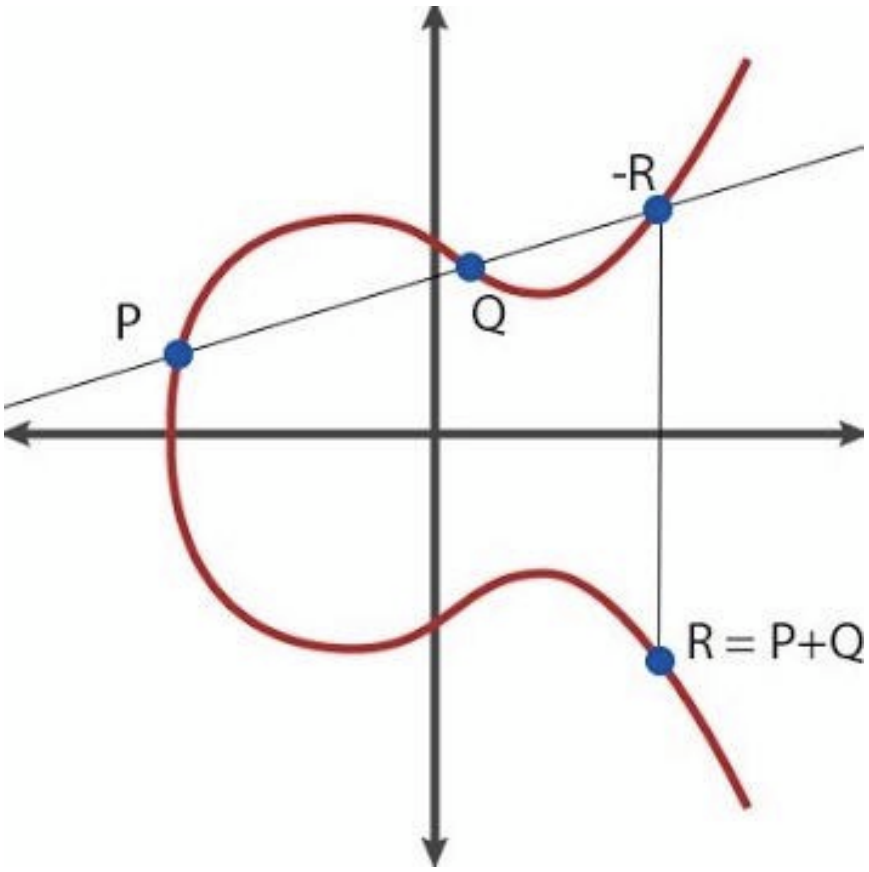
OSTROWSKI Camille
CUTTING Laurent

Cryptographie sur les courbes elliptiques

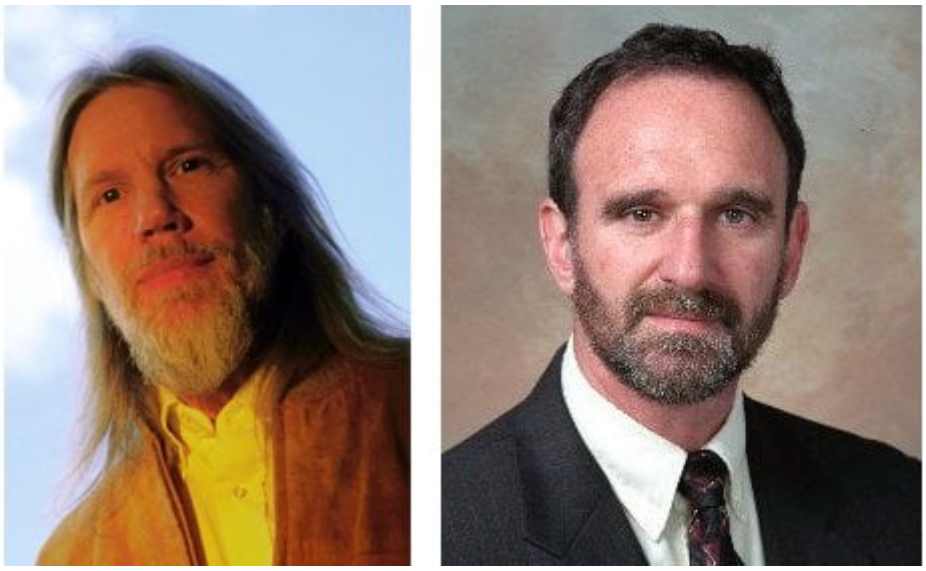
-

Elliptic Curves Cryptography



1. Introduction

Dans la fin des années 70, le monde de la cryptographie subit un énorme changement, tout d'abord grâce à l'invention du protocole d'échange public de clés Diffie-Hellman (1976). Ce protocole permet l'échange d'un secret permettant de communiquer de façon sécurisée et chiffrée, à distance, sans échange d'information préalable entre 2 agents, par exemple Alice et Bob, et ce, sans qu'un 3ème agent, Eve, ne puisse décrypter les messages, même en ayant accès à TOUTES les communications entre Alice et Bob.



Whitfield Diffie and Martin Hellman, inventeurs du protocole Diffie-Hellman

Ce protocole a été prouvé mathématique fiable contre les attaques passives (où l'attaquant peut écouter tous les échanges d'information) mais ce protocole est cependant vulnérable à un autre type d'attaque, les attaques de type homme du milieu (*Man In The Middle* ou *MITM*), en interceptant les messages entre Alice et Bob et en se faisant passer pour eux, Eve peut en effet faire croire à Alice et Bob qu'ils ont échangé un secret entre eux alors qu'en fait ils ont tous les deux échangé avec Eve.

Une parade pour pallier ce type d'attaque consiste à utiliser 'une fonction de cryptographie asymétrique pour signer les échanges entre Alice et Bob, et ainsi assurer l'identité de l'auteur des messages. C'est en 1977 que Rivest, Shamir et Adleman publieront leur algorithme de chiffrement, RSA, reposant sur une paire de clés, une clé privée et une clé publique, qui permet ainsi de signer les messages, ce qui aboutira à l'utilisation à grande échelle de Diffie-Hellman pour les communications sécurisées.

RSA possède cependant une faille qui rend de plus en plus facile les attaques par

“brute force” sur des clés de grandes tailles. Dans ce contexte, la cryptographie sur les courbes elliptiques vise à remplacer RSA par un autre système dont on ne connaît pour l’instant pas de faiblesse et proposant d’utiliser des clés de plus petites tailles en gardant le même niveau de sécurité.

2. Fonctions à trappe (*trapdoor functions*)

Tout d’abord, pour comprendre l'intérêt de la cryptographie sur les courbes elliptiques, il faut parler des fonctions à trappe (en anglais : *trapdoor functions*) :

Les algorithmes de cryptographie reposent sur le principe des fonctions à trappe, des fonctions faciles à calculer mais difficiles à inverser, c’est-à-dire retrouver la valeur d’une entrée en fonction d’une sortie, à moins de disposer d’un secret, aussi appelé trappe.

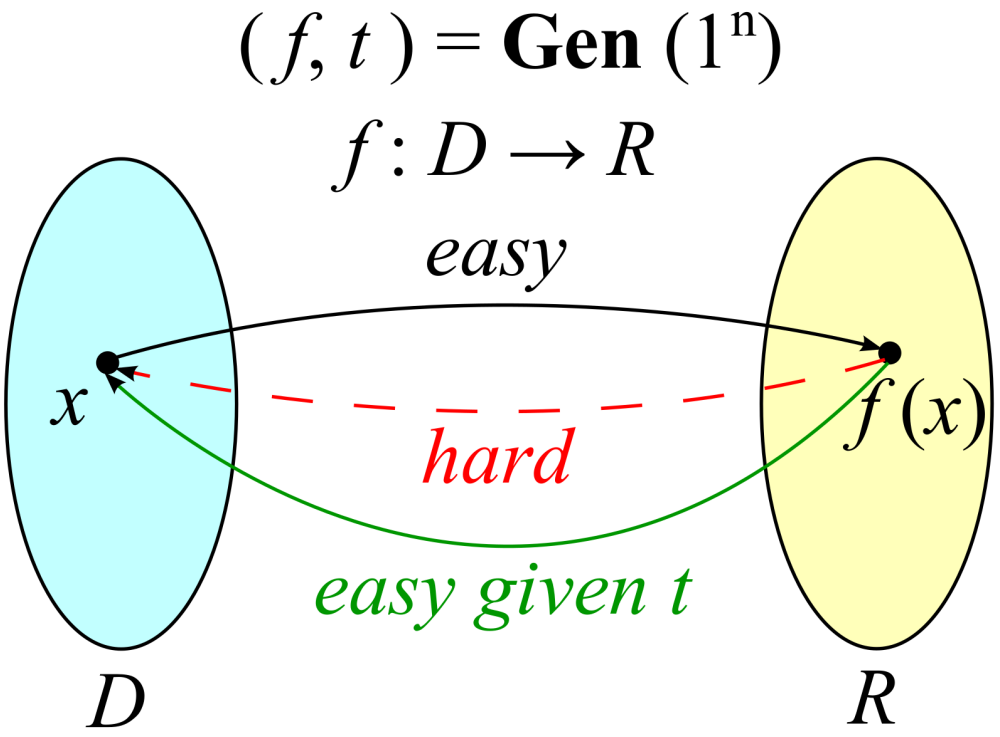


Schéma d’une fonction à trappe f, allant de D vers R, avec un secret t

En pratique, pour la cryptographie on recherche les propriétés suivantes pour les fonctions à trappe :

- La fonction doit être facile à générer, en connaissant le secret t, par un algorithme probabiliste en temps polynomial par rapport à la taille de t.
- L'application de la fonction doit être facile (en temps polynomial).
- Le calcul de l'inverse de la fonction, avec le secret, doit être facile (en temps polynomial).
- La méthode optimale pour inverser, sans le secret, doit être de complexité assez grande pour que les communications restent sécurisées en prenant en compte les capacités des ordinateurs dans les prochaines décennies. Dans le meilleur des cas, la fonction à trappe n'a pas de vulnérabilité et il n'existe pas de méthode plus optimale pour inverser sans le secret que la force brute.

En général, les fonctions utilisant des clés symétriques peuvent utiliser des clés quelconque, en revanche, les fonctions de cryptographie asymétriques forcent certaines propriétés sur les paires de clés, ce qui implique 2 résultats :

- Pour une taille de clé donnée, la proportion de clés asymétriques valides est moindre que pour des clés symétriques. L'espace de recherche pour retrouver le secret est donc plus restreint.
- Ces fonctions sont potentiellement vulnérables à certaines formes d'attaques qui se basent sur les propriétés mathématiques des clés, qui permettent de raccourcir le temps moyen de décryptage, parfois de façon drastique.

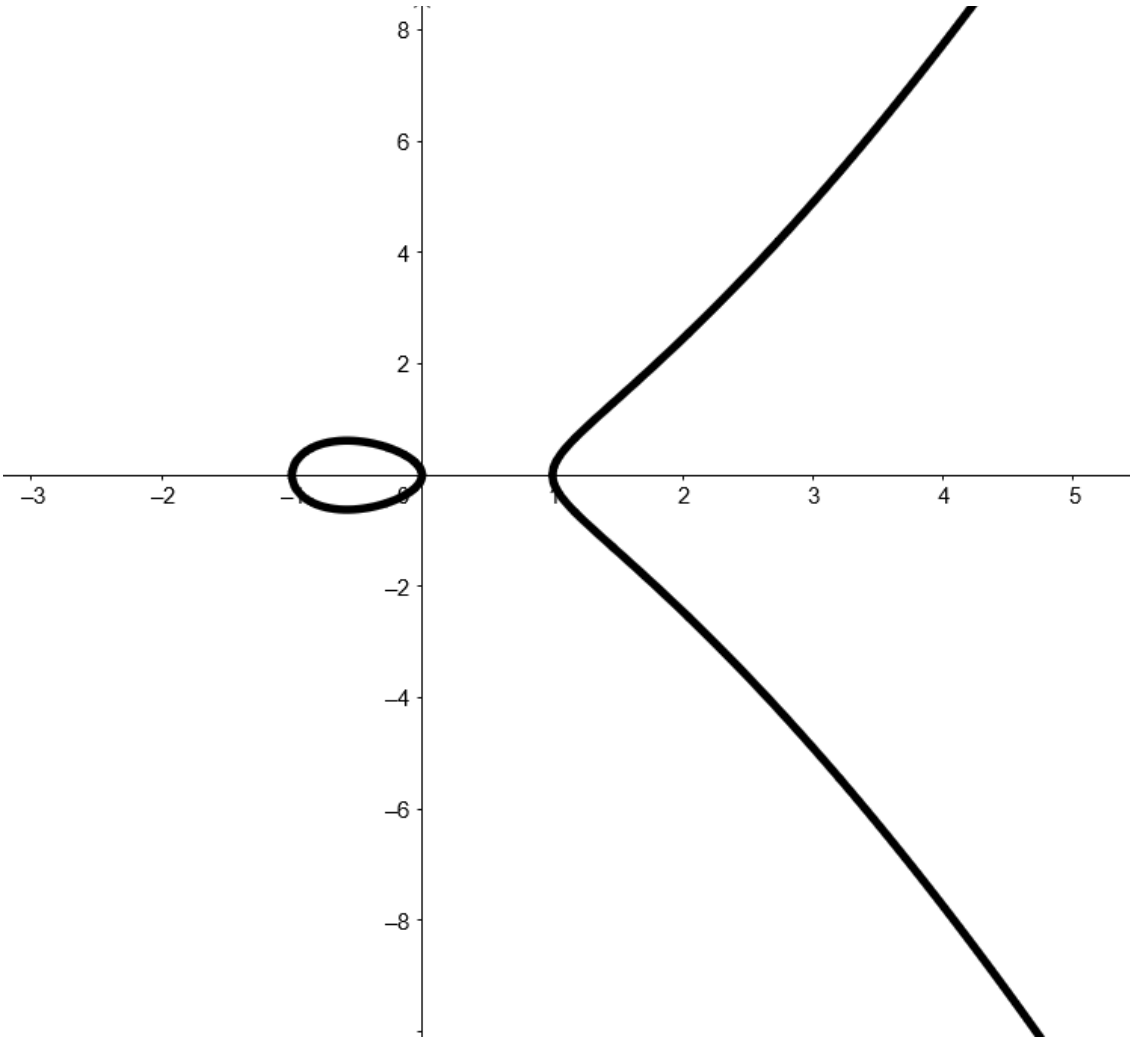
Ces propriétés impliquent que les fonctions avec des clés asymétriques ont besoin de secrets de plus grandes tailles, comparées aux fonctions symétriques, pour assurer une sécurité équivalente, et plus les failles permettent de raccourcir plus la taille doit augmenter également. Dans les contextes où les ressources de calcul sont limitées, une clé de grande taille est parfois indésirable, c'est pourquoi Ce désavantage est donc

3. Courbes elliptiques

Le principe de la CCE repose sur une famille de courbes algébriques particulière appelée les courbes elliptiques. Une courbe elliptique est paramétrée par deux coefficients a et b et est l'ensemble des points satisfaisant l'équation :

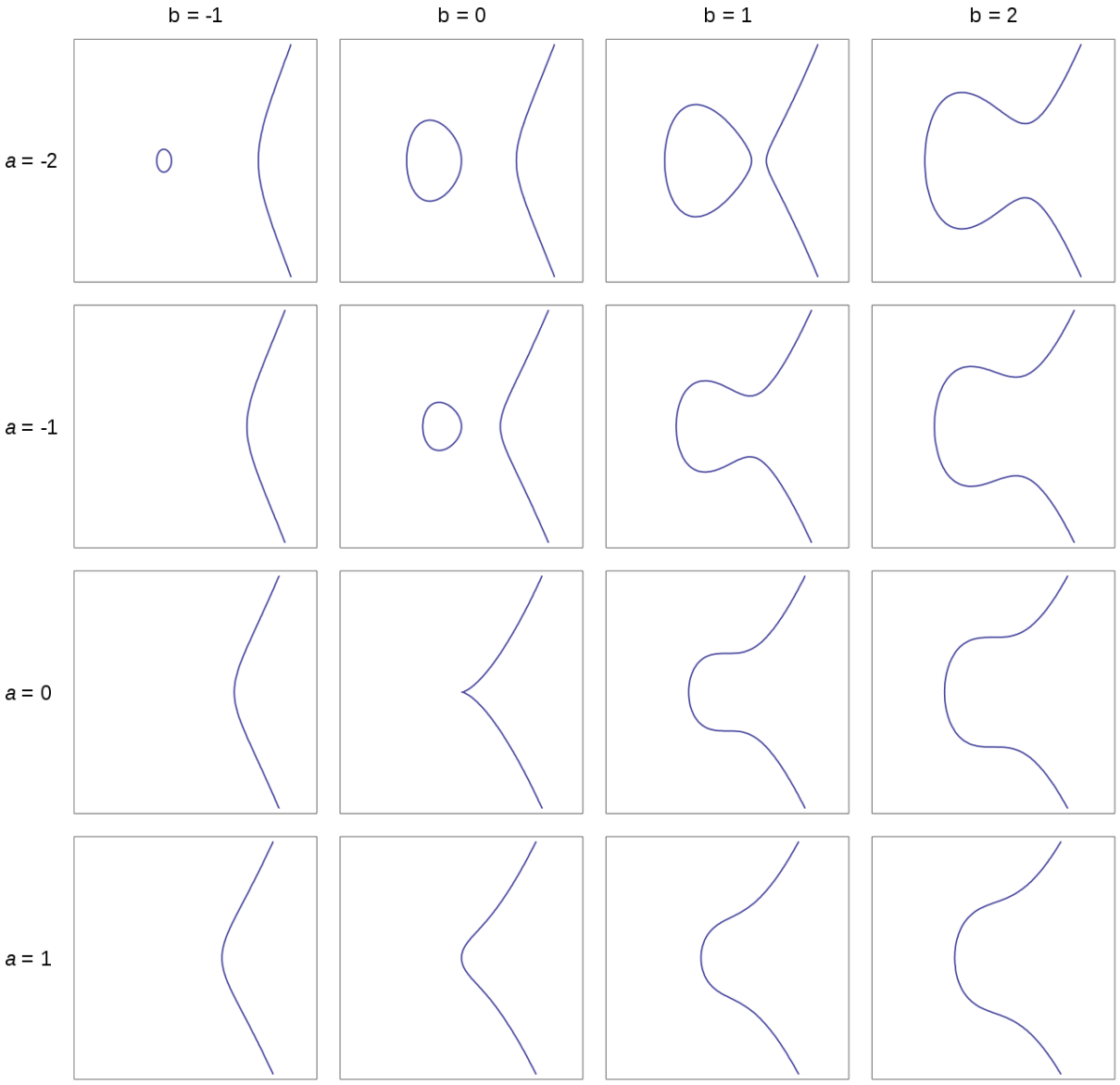
$$y^2 = x^3 + ax + b$$

On peut visuellement représenter ces courbes, par exemple en prenant x et y des points réels d'un plan, et en prenant les coefficients $a = -1$ et $b = 0$:



Exemple de courbe elliptique pour $a = -1$ et $b = 0$

Voici d'autres exemples de courbes dans la région $[-3; 3]^2$ pour différentes valeurs de a et b, on notera que le cas particulier $a = b = 0$ qui ne forme pas une courbe elliptique :



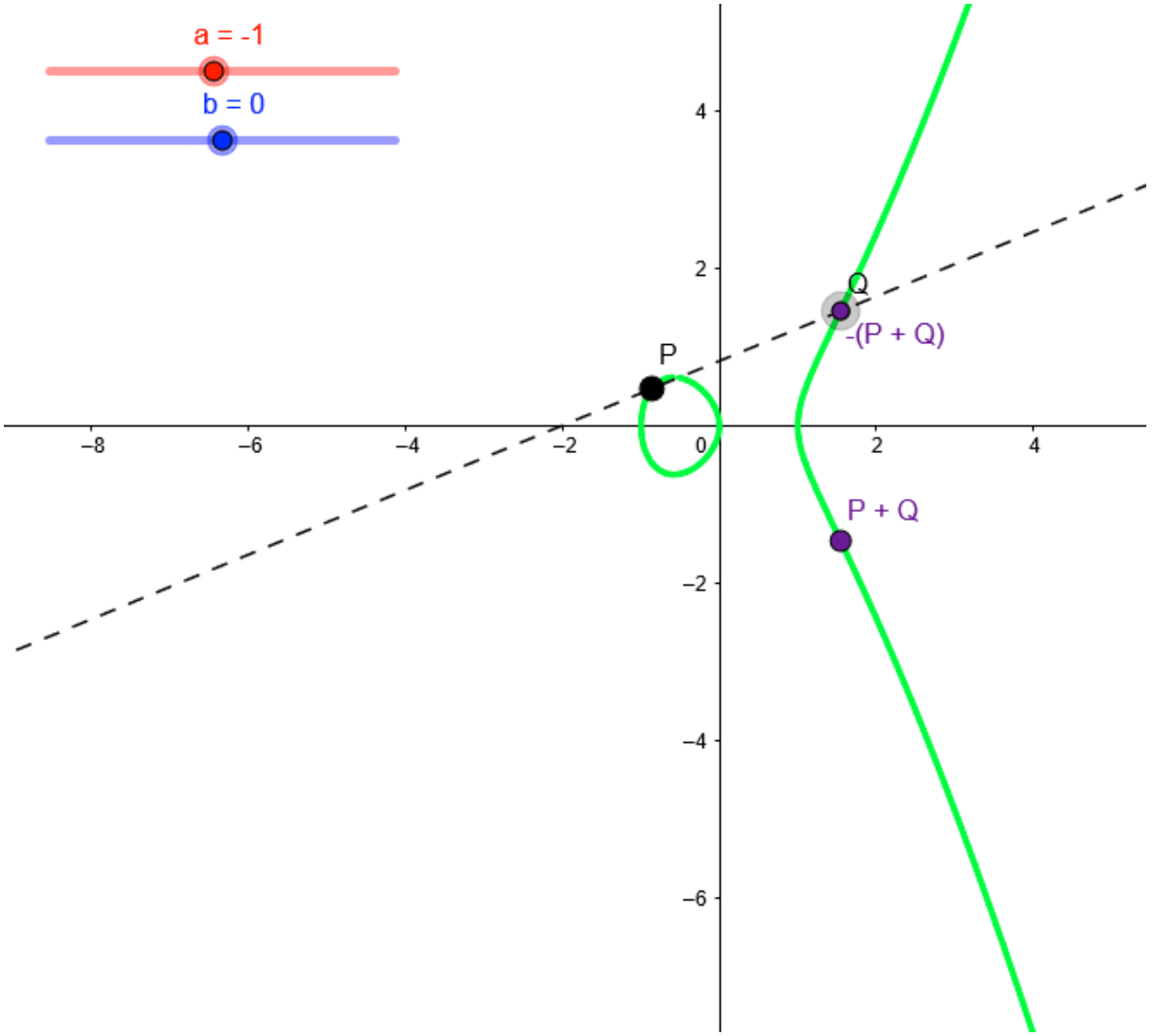
Différents exemples de courbes elliptiques pour différentes valeurs de a et b

Ces courbes disposent d'une opération **d'addition géométrique** qui est la clé de voûte de l'ECC.

L'addition géométrique de 2 points P et Q, communément notée $P + Q$ mais parfois $A . B$ ou $A \, dot \, B$ (produit scalaire) d'une courbe elliptique C se définit ainsi :

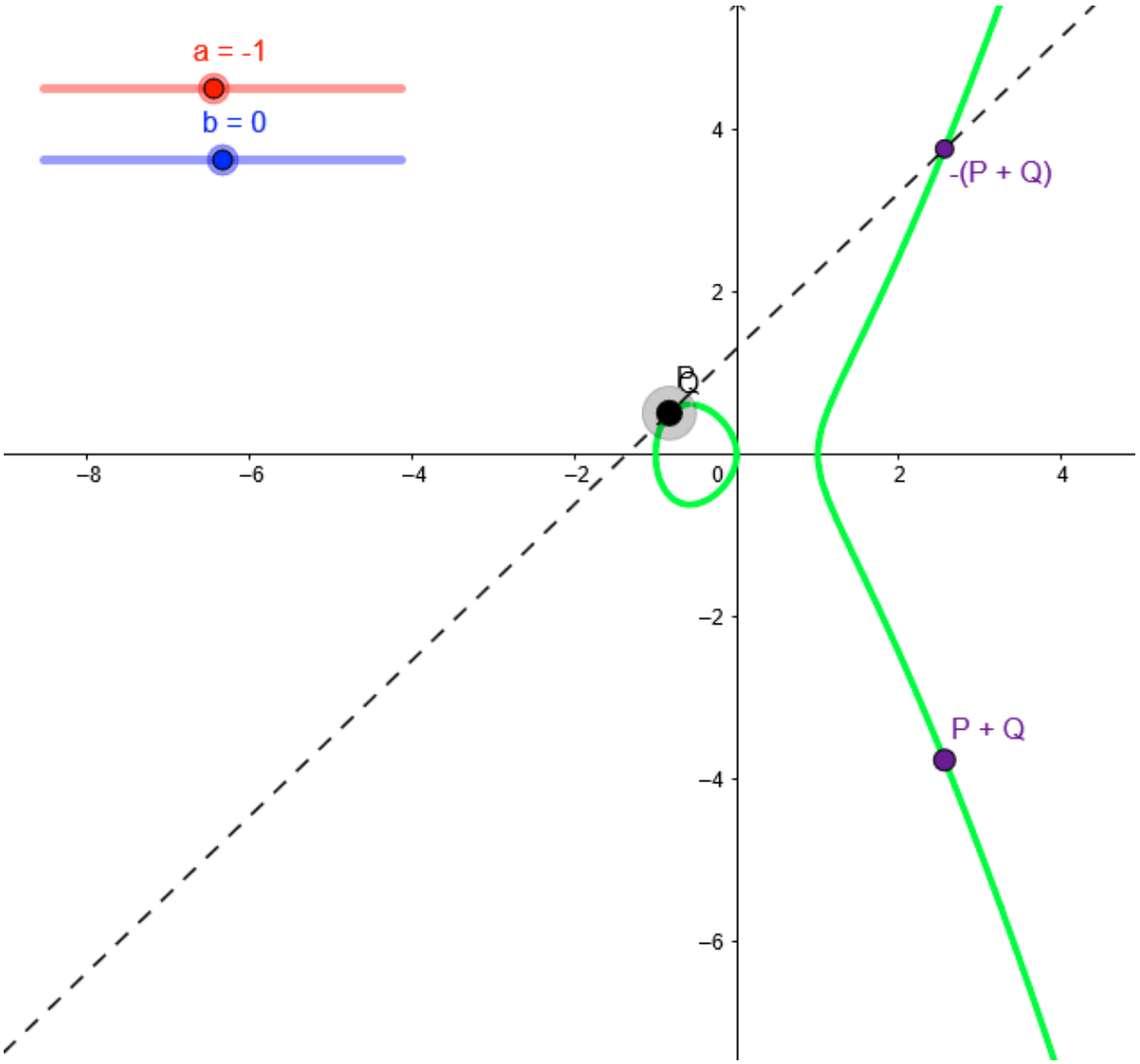
- On trace une droite passant par P et Q
- Cette droite intersecte la courbe C en un 3ème point, qu'on définit comme $-(P + Q)$. On peut prouver que la droite intersecte forcément la courbe en 3 points (sauf dans le cas où $P = -Q$, la droite étant verticale et ne coupant C qu'en 2 points) grâce au théorème de Bézout.
- On prend ainsi le symétrique du 3ème point par rapport à l'axe x pour obtenir $P + Q$

Par exemple :



Exemple d'addition géométrique de 2 points d'une courbe elliptique

Dans le cas particulier où $P = Q$, on prend la droite tangente à la courbe au point P, par exemple :



Exemple d'addition géométrique d'un point d'une courbe elliptique avec lui-même

Cette opération est également applicable dans le cas des courbes elliptiques sur des entiers, c'est-à-dire dans le cas où seul les coordonnées **entières** satisfaisant l'équation de la courbe font partie de C. Si P et Q sont entiers, alors $P + Q$ est également entier.

De plus, l'application d'une fonction de modulo sur les coordonnées conserve la propriété d'intersection de la droite entre P *et* Q et C en un troisième point. Visuellement, l'animation suivante montre assez bien ce qu'il se passe, quand la droite sort de la région du plan définie par la valeur du modulo, elle réapparaît de l'autre côté :

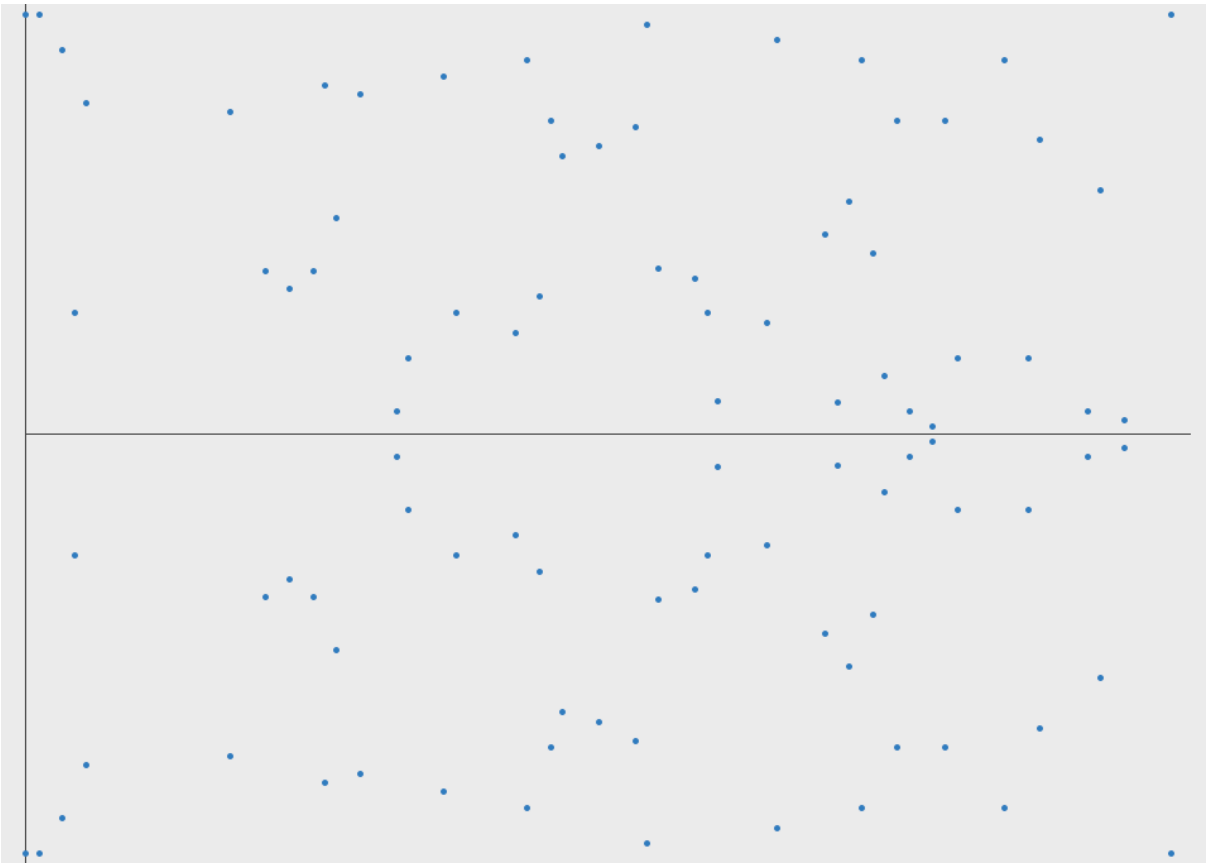


Illustration de l'application d'une fonction modulo sur le résultat de $A + B$ pour rester dans une région du plan proche de l'origine

Ces propriétés permettent de travailler avec des valeurs entières, ce qui permet de ne pas avoir d'erreurs sur les nombre flottant en Informatique, et qui sont relativement restreintes pour limiter la puissance de calcul nécessaire pour calculer les additions.

Avec l'opérateur d'addition, on peut également définir la multiplication d'un point P par un entier n comme ceci :

$$nP = \sum_{n} P$$

En pratique, cette opération est facile à appliquer, même pour n très grand, mais impossible à inverser, même en connaissant P. Dans le cadre de la cryptographie, cela nous permet de définir une fonction à trappe où n est le secret :

- nP est facile à calculer en utilisant l'exponentiation rapide (complexité temporelle en $log(n)$ fois le temps de faire une addition) en connaissant P le point de départ et n le secret.
- Retrouver n à partir de P et nP revient à essayer toutes les possibilités de P fois x, une à une, jusqu'à trouver la bonne valeur.

4. Cryptographie sur les courbes elliptiques

Grâce à notre fonction à trappe basée sur les courbes elliptiques, on peut maintenant définir un protocole de cryptographie publique permettant à Alice et Bob de communiquer de façon sécurisée et chiffrée :

- Alice et Bob se mettent d'accord pour générer une courbe C et un point P
- Alice envoie à Bob le résultat de n_{Alice} multiplié par P (où n_{Alice} est un entier connu uniquement d'Alice, c'est son secret).
- Bob envoie à Alice le résultat de n_{Bob} multiplié par P (n_{Bob} est le secret de Bob).
- Alice multiplie ensuite $n_{Bob}P$ par son propre secret pour obtenir $(n_{Alice} + n_{Bob})P$.
- Bob multiplie quant à lui $n_{Alice}P$ par son secret pour obtenir le même résultat qu'Alice.
- $(n_{Alice} + n_{Bob})P$ devient leur nouvelle clé de communication via un protocole utilisant une clé symétrique.

Un attaquant aurait alors accès à P, $n_{Bob}P$ et $n_{Alice}P$ mais ne pourra pas en déduire $(n_{Alice} + n_{Bob})P$, la seule façon pour lui de retrouver le résultat étant de bruteforcer le calcule d'un des 2 secrets

En pratique, la courbe C est définie par le protocole, et les 2 agents se mettent d'accord pour générer un point sur cette courbe.

5. Comparaison entre ECC et RSA

La difficulté à décrypter les messages chiffrés avec RSA se repose sur le problème de la décomposition en produit de facteurs premiers, un problème qu'on pensait, à l'époque où l'algorithme a été publié, aussi dur à résoudre que d'essayer des combinaisons d'entiers premiers en force brute, mais qui en réalité est vulnérable à au moins un type d'attaque.

Basée sur l'algorithme du **crible du corps de nombres généralisé** (*General number field sieve - GNFS* en anglais), cette attaque permet de réduire la complexité pour retrouver le secret d'un chiffrement RSA pour des clés de très grandes tailles (supérieure à 10^{100}) à, en notation L :

$$\exp\left(\left(\sqrt[3]{\frac{64}{9}}+o(1)\right)(\ln n)^{\frac{1}{3}}(\ln\ln n)^{\frac{2}{3}}\right)=L_n\left[\frac{1}{3},\sqrt[3]{\frac{64}{9}}\right]$$

Pour pallier cette faiblesse, il faut donc utiliser des clés de plus en plus grandes pour maintenir les propriétés cryptographiques de RSA face aux capacités des ordinateurs en constante évolution.

Par rapport à RSA, la cryptographie sur les courbes elliptiques repose sur une opération conçue pour ne pas être inversée facilement, ce qui permet à l'ECC d'avoir les propriétés suivantes :

- De plus petites tailles de clés (et donc de messages chiffrés et signatures) pour un niveau de sécurité donné par rapport à RSA, par exemple ci-dessous un tableau de comparaison des tailles de clés recommandées pour un niveau de sécurité donné :

Symmetric Key Size (bits)	RSA and Diffie-Hellman Key Size (bits)	Elliptic Curve Key Size (bits)
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	521

Table 1: NIST Recommended Key Sizes

Tableau de comparaison des tailles de clés recommandées en fonction du protocole, fourni par la NIST

- Une génération de clé plus rapide.
- Des calculs plus simples pour signer les messages, le chiffrement et le déchiffrement, ce qui permet de limiter le besoin de performances de calcul, ce qui est utile par exemple lors d'une utilisation dans un navigateur internet ou sur périphérique mobile.

On pourrait alors se demander pourquoi RSA reste prédominant dans les protocoles de cryptographie asymétrique, mais cela s'explique par plusieurs raisons :

- Tout d'abord, RSA est plus facile et rapide à implémenter, plus facile à comprendre

- Comme l'ECC a été inventée quelques années après (fin des années 80), RSA était alors déjà bien installé dans l'industrie, et les avantages de l'ECC n'étaient pas encore très prononcés à l'époque, les tailles de clés étant relativement limitées.
- Enfin, il y a une certaine méfiance vis-à-vis de l'ECC, car le protocole a été publié par la NIST (*National Institute of Science and Technology)* qui pourrait potentiellement avoir des liens avec la NSA. Il existe donc des suspicions d'une potentiellement *backdoor,* on a par exemple retrouvé des failles dans plusieurs protocoles utilisant des *magic numbers* fournis par des organisations externes comme la NSA, mais à ce jour, l'ECC semble dépourvue de faille intrinsèque et il existe plusieurs protocoles *opensource* fiables.

6. Conclusion

La cryptographie sur les courbes elliptiques est une technique de cryptographie développée dans la fin des années 80 qui permet de résoudre la plupart des problèmes des chiffrement basés sur RSA qui apparaissent ces dernières années. En particulier, l'ECC commence à se démocratiser dans la plupart des protocoles de communication chiffrées, elle est par exemple très utilisée dans le domaine de la crypto-monnaie et pourrait bien finir par totalement remplacer RSA.