Arithmétique modulaire, Logarithmes discrets,

Courbes elliptiques etc.

par

Jean-Luc Stehlé

Document complémentaire pour le cours de Mathématiques pour la Sécurité Informatique

L'arithmétique modulaire et plus particulièrement l'exponentiation modulaire et sa réciproque, le logarithme discret, sont à la base de nombreux systèmes cryptographiques, (entre autres RSA et Diffie-Hellman…). Après avoir rapidement rappelé les principes de l'arithmétique modulaire, on montre comment la théorie des courbes elliptiques permet de construire des structures algébriques ayant des propriétés similaires et utilisables en cryptographie.

1. L'arithmétique modulaire

Plusieurs systèmes cryptographiques classiques sont basés sur l'arithmétique modulo un nombre entier N, qu'on appelle aussi **arithmétique modulaire** , et dont nous rappelons ici les principales propriétés.

1.1. L'anneau des entiers modulo *N*

Deux nombres u_1 et u_2 dont la différence est un multiple entier de N sont appelés « congrus modulo N » et on écrit alors $u_1 \equiv u_2$ [N] . L'arithmétique modulo N consiste à faire comme si des nombres congrus modulo N étaient en fait égaux : lorsqu'on écrit le nombre entier u , on a en tête l'ensemble des entiers congrus à u modulo N. Un tel ensemble s'appelle une classe d'équivalence modulo N. On dira qu'on travaille sur des nombres définis modulo N.

L'ensemble des classes d'équivalence modulo N appelé l'ensemble quotient de $\mathbb{Z}$ par $N\mathbb{Z}$ et est noté $\mathbb{Z} / N\mathbb{Z}$, où le symbole $\mathbb{Z}$ désigne l'ensemble des entiers relatifs, et $N\mathbb{Z}$, l'ensemble des multiples de N (obtenu en multipliant par N chacun des éléments $\mathbb{Z}$).

Si dans une addition de nombres entiers $w_1 = u_1 + v_1$, on remplace les termes u_1 et v_1 par des termes u_2 et v_2 congrus aux premiers modulo N (vérifiant $u_2 \equiv u_1$ [N] et $v_2 \equiv v_1$ [N]) le nouveau résultat $w_2 = u_2 + v_2$ sera congru à l'ancien modulo N ($w_2 \equiv w_1$ [N]). Si on additionne un élément quelconque de la classe d'équivalence de u_1 et un élément quelconque de la classe d'équivalence de v_1 , on obtient toujours un élément de la classe d'équivalence de w_1. On peut ainsi définir une notion d'addition entre classes d'équivalences. On a une propriété analogue pour la multiplication.

Chaque classe d'équivalence modulo N contient un et un seul nombre compris entre 0 et N-1. Travailler en arithmétique modulo N revient à travailler sur les entiers compris entre 0 et N-1 et, lorsque le résultat d'un calcul arithmétique sort de ces limites, à remplacer ce résultat par le reste de sa division entière par N.

Les lois de l'addition et de la multiplication en arithmétique modulaire (dans $\mathbb{Z} / N\mathbb{Z}$) sont, pour la plupart, similaires aux lois de l'arithmétique classique dans $\mathbb{Z}$. On a une structure d'anneau commutatif sur $\mathbb{Z} / N\mathbb{Z}$ est appelé l'anneau quotient de l'anneau des entiers relatifs $\mathbb{Z}$ par l'idéal $N\mathbb{Z}$ formé des multiples de N.

Cependant, contrairement à $\mathbb{Z}$ l'anneau $\mathbb{Z} / N\mathbb{Z}$ n'est pas intègre : le produit de deux nombres non nuls peut être nul. Prenons l'exemple de N=12. On a $6 \times 4 \equiv 0$ [12] (car 24 est congru à 0 modulo 12).

1.2. Le groupe multiplicatif modulo N

On dira qu'un élément u de $\mathbb{Z} / N\mathbb{Z}$ est inversible modulo N si et seulement s'il existe v tel que $u.v \equiv 1$ [N]. Ce sera le cas si et seulement si u est non nul et premier à N (n'a pas de diviseur commun avec N). Le produit de deux éléments inversibles est lui-même inversible et l'ensemble des éléments inversibles modulo forme un groupe multiplicatif noté U_N .

Le nombre d'éléments de U_N (égal au nombre d'entiers compris entre 1 et N-1 et premiers à N) est noté $\Phi(N)$, et est appelé indicateur d'Euler de N.

En particulier, lorsque N est premier, on a $\Phi(N) = N–1$ et tout élément non nul de $\mathbb{Z} / N\mathbb{Z}$ est inversible. L'ensemble $\mathbb{Z} / N\mathbb{Z}$ est alors un corps.

Le calcul explicite de l'inverse d'un nombre modulo N se fait relativement facilement grâce à un algorithme connu sous le nom d'**algorithme d'Euclide généralisé**.

1.3. L'exponentiation modulaire

L'exponentiation modulaire est l'opération d'élévation à une puissance modulo N. Étant donné un nombre b dans $\mathbb{Z} / N\mathbb{Z}$ (donc défini modulo N), appelé la base, et un nombre entier e appelé l'exposant, on définit le nombre $a = b^e[N]$ (b puissance e modulo N) comme le résultat de la multiplication modulo N de e nombres égaux à b .

La base b et le résultat a sont définis modulo N. Cela signifie que si on remplace la base b par un nombre qui lui est congru modulo N, le nouveau résultat sera congru à l'ancien modulo N. Par contre, il en va différemment de l'exposant e.

Par exemple lorsque N est un nombre premier (n'ayant pas d'autre diviseur entier que lui-même et l'unité), ou, plus généralement, lorsque N n'a pas de facteur carré autre que 1, on a la relation $b^d[N] = b^e[N]$ dès que $d \equiv e\ [\Phi(N)]$.

Dans le cas général, cette relation ne reste vraie que si la base b est inversible modulo N. Vérifions le pour N=12.
On a $\Phi[12]= 4$, car il y a 4 entiers inférieurs à 12 et premiers à 12 : l'ensemble des éléments inversibles modulo 12 est $U_{12} = \{ 1, 5 ,7 ,12 \}$

Choisissons par exemple b = 2 (non inversible modulo 12 car non premier à 12). La suite des puissances successives de 2 est la suite 2, 4, 8, 4, 8, 4, 8, 4, 8, … (en effet, on a $8 \times 2 = 16 \equiv 4\ [12]$, donc $2^1 \equiv 2\ [12]$ et aucune autre puissance de 2 n'est congrue à 2^1 modulo 12.).

Si par contre on choisit b=5 (premier à 12 donc inversible modulo 12), la suite des puissances successibles de 5 est 5, 1, 5, 1, 5, 1,… (en effet on a $5 \times 5 = 25 \equiv 1\ [12]$) et $5^d = 5^e\ [12]$ dès que $d \equiv e\ [4]$ (deux termes distants de 4 dans la suite précédente sont toujours égaux)

Lorsque N n'a pas de facteur carré, cela signifie qu'il est égal au produit d'un ou plusieurs nombres premiers tous différents : $N = p_1\ p_2\ \ldots\ p_r$. Il est facile de voir que, dans ce cas, son indicateur d'Euler est donné par la formule $\Phi(N) = (p_1-1)(p_2-1)\ldots(p_r-1)$

Temps de calcul

Pour fixer les idées, dans les applications cryptographiques les valeurs de N habituellement utilisées sont souvent de très grands nombres (par exemple de l'ordre d'une ou quelques centaines de chiffres décimaux), et les calculs en arithmétique modulo de tels nombres ne sont possibles que sur ordinateur.

Lorsque l'exposant e devient grand le calcul direct de $b^e[N]$ nécessite e–1 multiplications modulaires, ce qui revient rapidement prohibitif. Néanmoins il existe un algorithme efficace permettant d'effectuer ce calcul en un temps approximativement proportionnel au nombre de bits nécessaires pour écrire l'exposant e sous forme binaire.

Cet algorithme consiste tout simplement à calculer les carrés successifs de b modulo N ($b^2[N]$, $b^4[N]$, $b^8[N]$, $b^{16}[N]$, $b^{32}[N]$, $b^{64}[N]$, et ainsi de suite, puis à ne retenir dans cette liste que ceux qui correspondent à des bits à 1 dans l'écriture binaire de e.

Si e est à 512 bits, il faudra en moyenne 768 multiplications modulo N pour calculer $b^e[N]$, et non pas de l'ordre de 2^{512} multiplications

Notons néanmoins que si les temps nécessaires à une addition ou une multiplication élémentaires se comptent en fractions de microsecondes, les exponentiations modulaires en grands nombres sont bien plus lentes à calculer. Les temps se mesurent alors plutôt en centièmes ou dixièmes de secondes (pour des nombres à une centaine de chiffres décimaux).

En première approximation, lorsque l'exposant peut prendre une valeur quelconque (modulo $\Phi(N)$), on peut estimer que, sur un processeur donné, les temps de calculs d'une exponentielle modulaire croissent comme le cube du nombre de bits de N. Retenons que le passage d'une arithmétique modulaire à 512 bits à une arithmétique modulaire à 1024 bits multiplie en gros les temps de calculs de l'exponentielle modulaire par 8

1.4. Le problème du logarithme discret

L'exponentiation modulaire décrite ci-dessus permet, connaissant la base b et l'exposant e , de calculer facilement $a = b^e[N]$. Le problème réciproque, consistant à retrouver l'exposant e connaissant a et b , est en général beaucoup plus complexe. Lorsque N devient grand, les temps de calcul se comptent rapidement en millions d'années non plus en centièmes de secondes comme dans le cas de l'exponentiation modulaire.

Ce problème est appelé le problème du logarithme discret, par analogie avec le logarithme classique. Si on était en arithmétique classique, $a = b^e$ est équivalent à $e = \text{Log } a / \text{Log } b$ et le calcul de e est donc immédiat. Malheureusement (et heureusement pour les cryptographes) il n'existe pas d'équivalent du logarithme en arithmétique modulaire.

On se placera toujours dans le cas où b est inversible modulo N (donc premier à N). Considérons alors la suite des puissances successives de b (en partant de $b^0=1$). On démontre qu'il existe un entier q positif tel que $b^q \equiv 1[N]$ (en on choisira pour q le premier entier à vérifier cette propriété). Ce nombre q sera appelé **ordre de b** . On montre facilement qu'il divise $\Phi(N)$. La suite des puissances successives de b est donc périodique de période q, et peut s'écrire $1, b, b^2, b^3, \ldots, b^{q-1}, b^q \equiv 1, \ldots$

Si on remarque que $b^u \times b^v \equiv b^{u+v}$ [N], et que $b^{u+\lambda q} \equiv b^u$ [N] (pour un entier λ quelconque) , on met en évidence un isomorphisme entre le groupe additif $\mathbb{Z} / q\mathbb{Z}$ (ensemble des entiers modulo q, ordre de b) et le groupe multiplicatif des puissances successives de b. Mais si on souhaite faire des calculs explicites, cet isomorphisme apparaît comme une fonction à sens unique : connaissant e il est facile de calculer b^e , mais dans l'autre sens, connaissant b^e il est impossible en pratique (sauf cas particulier) de retrouver e.

Dans certains cas (et en particulier lorsque N est premier), on peut choisir b tel que la suite de ses puissances successives contienne tous les éléments du groupe multiplicatif U_N . L'ordre de b est alors égal à Φ(N) et on dira que **b est un générateur de U_N**.

2. Systèmes cryptographiques basés sur l'exponentiation modulaire

2.1. Principe de base

On utilise la propriété fondamentale citée plus haut : lorsque N n'a pas de facteur carré autre que l'unité (et en particulier lorsque N est lui-même premier) on a la relation $b^d[N] = b^e[N]$ dès que $d \equiv e\ [\Phi(N)]$.

Étant donnés deux entiers c et d inverses l'un de l'autre modulo Φ(N) (donc vérifiant $c.d \equiv 1\ [\Phi(N)]$), on a alors, pour tout entier les égalités

$$(M^c)^d [N] \quad \equiv \quad (M^d)^c [N]$$

Étant donné un message représenté par un nombre compris entre 0 et N–1 (ce qui est possible si par exemple le message est une suite de bits ayant moins de bits que N , sinon, on découpera le message en blocs plus petits) la fonction de codage est tout simplement l'élévation à la puissance c modulo N et le décodage est l'élévation à la puissance d modulo N ou réciproquement.

Pour mettre en place ce système de cryptage, il suffit donc que deux interlocuteurs se mettent d'accord d'une part sur la valeur de N modulo laquelle on travaille et sur les valeurs des exposants c et de d .

Le nombre N doit être sans facteur carré. Soit il est premier, soit il s'écrit sous la forme $N = p_1 p_2 \ldots p_r$ où les p_i sont r nombres premiers distincts entre eux (le premier cas correspondant à r=1). Si N est premier, on a Φ(N) = N-1 . Dans le cas général, $\Phi(N) = (p_1-1)(p_2-1)\ldots(p_r-1)$. Ce nombre N peut, sans inconvénient, être publié.

Les exposants c et d sont liés par la relation $c.d \equiv 1\ [\Phi(N)]$. Pour les déterminer, il suffit de choisir c quelconque premier à Φ(N) donc inversible modulo Φ(N). Connaissant c, il est alors facile de calculer son inverse d modulo Φ(N), en utilisant par exemple l'algorithme d'Euclide généralisé. Pour cela, il faut bien entendu, connaître Φ(N). Par contre les exposants c et d restent confidentiels.

2.2. Le système à clé publique RSA

C'est une application immédiate du système de cryptage décrit ci-dessus. Il est basé sur la remarque suivante : si un nombre N est le produit de deux très grands nombres premiers distincts p et q (une centaine de chiffres décimaux), il est impossible de retrouver p et q connaissant N (du moins il n'existe à ce jour actuellement aucune méthode de calcul permettant de les retrouver en un temps raisonnable).

Si on ne connaît pas p et q, il est impossible de impossible de retrouver $\Phi(N) = (p-1)(q-1)$.

Un utilisateur va donc se générer confidentiellement deux très grands nombres premiers p et q, calculer et publier leur produit N. Il va par ailleurs générer un nombre c premier à $\Phi(N)$ et le publier. Mais bien sûr, il conserve confidentiel $\Phi(N)=(p-1)(q-1)$. Il pourra donc de son côté calculer d, inverse de c modulo $\Phi(N)$, mais sera seul a pouvoir les calculer. L'exposant c est utilisé comme clé publique et l'exposant d comme clé secrète.

Temps de calcul

Compte tenu des temps de calcul d'une exponentiation modulaire pour les grandes valeurs de N utilisées en RSA, on ne peut d'espérer des vitesses de chiffrement se mesurant en kilo bits par seconde alors que les communications se font à des vitesses se mesurant en dizaines ou centaines de mégabits pas seconde. A moins d'utiliser des dizaines de milliers de processeurs de cryptage fonctionnant en parallèle, il est donc hors de question d'utiliser RSA pour chiffrer des données « à la volée ».

En pratique un utilisera plutôt RSA pour des applications comme l'authentification des interlocuteurs, de la signature électronique, de la non-répudiation,…

3. Systèmes basés sur le logarithme discret

3.1. Le protocole d'échange de clés de Diffie-Hellman

Ce protocole est classiquement utilisé par deux interlocuteurs qui communiquent sur un réseau public et veulent se mettre d'accord sur une clé de cryptage confidentielle. On travaille en arithmétique modulo N où N est un très grand nombre premier (par exemple de l'ordre d'une ou quelques centaines de chiffres décimaux), choisi une fois pour toutes et publié. On choisit par ailleurs un nombre g générateur du groupe multiplicatif U_N .

Lorsque deux utilisateurs (appelons les Alice et Bob) désirent communiquer entre eux (après s'être mutuellement authentifiés), chacun va, de son côté, confidentiellement, choisir un nombre aléatoire a pour Alice et b pour Bob. Le protocole est alors le suivant

1. Alice calcule $g^a[N]$ et le communique à Bob (sur le réseau public).
2. Bob calcule $g^b[N]$ et le communique à Alice (sur le réseau public).
3. Chacun des deux interlocuteurs peut maintenant, de son côté, calculer $g^{ab}[N] = g^{ba}[N]$ qui va servir de clé de chiffrement entre eux

Un pirate espionnant le réseau public (et connaissant N et g, verra passer $g^a[N]$ et $g^b[N]$, mais n'a aucun moyen de calculer $g^{ab}[N]$). Pour ce faire, il faudrait qu'il puisse résoudre le problème du logarithme discret.

Il est clair que si on résout le problème du logarithme discret, on sait casser le protocole de Diffie-Hellman. La réciproque revient à dire que si on sait casser Diffie-Hellman, on sait résoudre le logarithme discret. Cette réciproque est conjecturée mais n'a pas encore été prouvée.

3.2. Le protocole à clé publique El-Gamal

Conceptuellement, ce protocole est similaire à celui de Diffie-Hellman. Les nombres N et g sont choisis et publiés une fois pour toutes. L'utilisateur Alice choisira confidentiellement un nombre a qui est sa clé secrète, et publiera $g^a[N]$ qui est sa clé publique.

N'importe quel utilisateur, par exemple Bob, peut envoyer un message confidentiel à Alice. Pour ce faire, il génère un nombre aléatoire confidentiel b (qui jouera le rôle de clé secrète provisoire pour lui) et calcule d'une part une clé publique provisoire $g^b[N]$ et d'autre part, à partir de la clé publique de Alice le nombre $g^{ab}[N]$ qui servira de masque pour crypter son message.

Si M est le message initial clair que Bob doit transmettre à Alice, (suite de bits de longueur inférieure à celle de N), Bob envoie à Alice d'une part le nombre $g^b[N]$ et d'autre part l'expression $M \oplus g^{ab}[N]$ (où $\oplus$ désigne l'opération XOR (OU Exclusif) bit à bit).

Le destinataire peut recalculer le masque $g^{ab}[N]$ à partir de sa clé secrète et de l'information que lui a envoyée Bob, et retrouver ainsi le message M.

La sécurité de ce protocole est la même que celle de Diffie-Hellman : un pirate pourra connaître $g^a[N]$ et $g^b[N]$, mais n'a aucun moyen de calculer $g^{ab}[N]$, donc il ne pourra pas lire le message.

Il faut bien sûr changer de b à chaque message. Cela double le nombre total de bits à transmettre, et c'est là l'un des inconvénients de ce protocole. L'autre est bien sûr celui des temps de calcul. On doit

calculer pour chaque bloc une exponentielle modulaire, donc les temps de calcul sont du même ordre que ceux de RSA et il est hors de question d'utiliser ce type d'algorithme pour du chiffrement « à la volée ».

3.3. Authentification de l'utilisateur

Dans la mesure où le protocole de El-Gamal permet à tout le monde d'envoyer un message que seul Alice pourra lire, Alice pourra facilement s'authentifier en prouvant qu'elle a su lire le message. Il suffit de lui envoyer un message aléatoire codé en El-Gamal et lui demander de renvoyer la version décodée.

Le protocole peut d'ailleurs se simplifier.

Rappelons qu'Alice s'est choisi un nombre a qui est sa clé secrète, et a publié $g^a[N]$ qui est sa clé publique. Un interlocuteur (par exemple Bob) voulant authentifier Alice tirera un nombre aléatoire b qu'il garde secret, calculera $g^b[N]$ et l'enverra à Alice en lui demandant de renvoyer par retour $g^{ab}[N]$. Cela prouvera alors que Alice connaît la clé secrète a . La vérification est possible parce que Bob peut recalculer $g^{ab}[N]$ à partir de la clé publique d'Alice et du secret b.

3.4. Protocole DSA de signature électronique

L'idée est toujours la même : Alice possède un secret a qu'elle est seul à connaître et a publié $g^a[N]$. En fait, le protocole est un peu plus subtil, mais il reste basé sur des exponentiations modulaire et sa sécurité est celle du protocole de Diffie-Hellman.

4. Considérations générales sur l'utilisation de l'arithmétique modulaire en cryptographie.

Tous les codes utilisés et évoqués ci-dessus reposent sur l'exponentiation modulaire. Compte tenu des temps de calcul, il ne sont pas adapté à du cryptage à la volée et il vaut mieux les réserver aux protocoles d'authentification, aux signature électronique, aux échanges de clé. Le corps du message sera

crypté pas des algorithmes plus rapides mus en œuvre une fois que les interlocuteurs se seront mutuellement authentifiés et se seront mis d'accord sur une clé de cryptage.

Pour le chiffrement à la volée l'algorithme standard recommandé à partir du printemps 2001 est l'algorithme AES (Advanced Encryption Standard). L'arithmétique utilisée est un peu plus complexe que l'arithmétique modulaire classique. Par exemple, les opérations réalisées au niveau de l'octet le sont dans l'anneau des polynômes sur $\mathbb{Z} / 2\mathbb{Z}$ modulo un polynôme de degré 8 spécifié une fois pour toutes.

4.1. Sécurité des protocoles basés sur le logarithme discret

Rappelons que le problème du logarithme discret consiste à retrouver x connaissant g et $g^x[N]$. Si on savait retrouver x en un temps raisonnable, les systèmes basés sur le logarithme discret sont bons à mettre à la poubelle.

La complexité du problème est entre autres liée à l'ordre de g (c'est-à-dire à la première valeur positive de q pour laquelle on a $g^q \equiv 1$). Lorsque N et g sont correctement choisis, q est de l'ordre de grandeur de N. La recherche de x par essais et erreurs nécessite un temps de calcul proportionnel à N. Malheureusement, il existe des méthodes plus rapides pour résoudre le logarithme discret. Une méthode prometteuse est un crible (« number field sieve ») proposé par Lenstra et Gordon en 1993. Les temps de calcul sont de l'ordre de $\exp(O(\sqrt[3]{\ln q \quad \ln\ln^2 q}\ ))$, à comparer à un temps de l'ordre de $q \approx \exp(\ln\ q)$ par la méthode directe par essais et erreurs. Pour des nombres de l'ordre de cent chiffres, on devrait arriver à des temps de calcul de l'ordre de l'année ou guère plus. Pour des nombres à 512 ou 1024 bits, il semble que le problème du logarithme discret soit encore largement hors de portée des ordinateurs actuels.

Cela dit, ce qui est impossible aujourd'hui risque d'être possible demain avec les progrès de la technologie. Néanmoins, il n'y a pas lieu de s'inquiéter. Imaginons (ce qui est fort vraisemblable) que d'ici deux ans les puissance des processeurs soient multipliés par 10, tant en vitesse de calcul qu'en espace mémoire. Nous pourrons alors, sans être pénalisés, doubler le nombre de bits de N (ce qui multiplie par 8 les temps de calculs des exponentiations modulaires et par 2 la mémoire nécessaire). Par contre le temps nécessaire à résoudre le logarithme discret va être multiplié par un facteur colossalement plus élevé, rendant nos protocoles bien plus surs qu'actuellement.

Le risque majeur pourrait provenir d'un mathématicien génial qui trouverait le moyen de résoudre le logarithme discret en un temps de l'ordre de ln(q) (donc proportionnel au nombre de bits de N). Cela paraît improbable mais non impossible.

5. Utilisation de courbes elliptiques en lieu et place de l'arithmétique modulaire

L'idée d'utiliser la théorie des courbes elliptiques en cryptographie remonte aux années 1985 et a été proposée indépendamment par N. Koblitz et par V. Miller.

Son intérêt est de permettre la construction de structures algébriques dans lesquelles on a des notions analogues à celles de l'exponentielle modulaire et du logarithme discret, et dans lesquelles le logarithme discret serait encore plus difficile à résoudre qu'en arithmétique modulaire classique. Pour le moment, il ne semble pas qu'il existe, sur ces structures, d'algorithmes de calcul du logarithme discret significativement plus rapides que $\exp(O(\ln q))$. Il semblerait donc qu'on puisse obtenir une sécurité comparable en travaillant sur moins de bits qu'en arithmétique modulaire classique.

Néanmoins, on est à la merci des progrès des mathématiques. On pourrait d'ailleurs faire la remarque que le problème du logarithme discret en arithmétique modulaire classique est connu depuis longtemps et a été beaucoup étudié, surtout depuis une vingtaine d'années, date des brevets de Diffie-Hellman. S'il y avait vraiment un algorithme génial, on l'aurait vraisemblablement déjà trouvé.

Des structures algébriques complexes comme celles construites sur les courbes elliptiques ont été bien moins étudiées par le passé. Toutefois, depuis leur apparition relativement récente en cryptographie, de nombreux spécialistes se sont penchés sur le sujet et il n'est pas exclu qu'on découvre prochainement des algorithmes intéressants pour résoudre l'équivalent « courbes elliptiques » du problème du logarithme discret. Il faut bien sur se méfier de ce type de raisonnement hâtif, et il est difficile, dans l'état actuel de nos connaissances, de faire le choix entre l'arithmétique modulaire classique ou les courbes elliptiques.

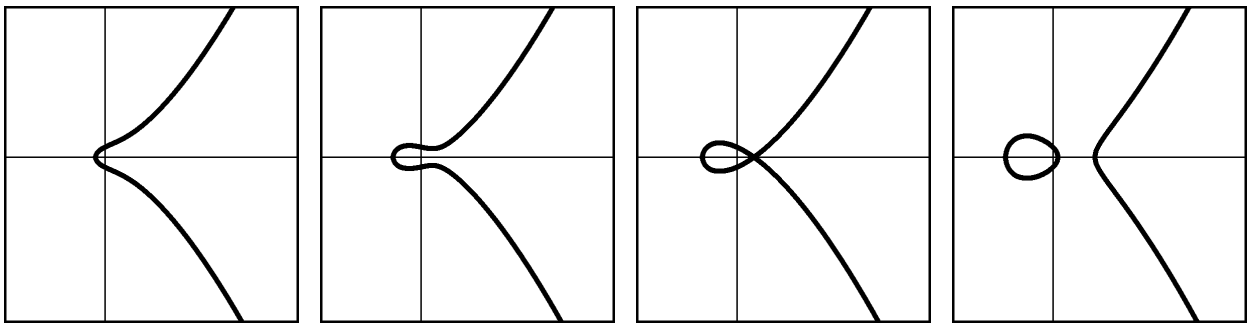
5.1. Présentation « naïve » des courbes elliptiques sur $\mathbb{R}$

Les fonctions et les intégrales elliptiques ont été introduites au XIX$^{\text{ième}}$ siècle, en liaison avec des problèmes liés au calcul des longueurs des arcs d'ellipses. Elles jouent un rôle qui généralise à celui des fonctions circulaires (sinus, cosinus,…) et ont donné naissance à la très riche théorie des courbes elliptiques. Notons que la forme de ces courbes ne ressemble nullement à une ellipse, pas plus qu'une sinusoïde à un cercle.

Définition des courbes elliptiques réelles

Lorsqu'on travaille dans le corps $\mathbb{R}$ des nombres réels, une courbe elliptique peut toujours, après changement de variables, se ramener à une équation de la forme $y^2 = x^3 + p\,x + q$. Selon la valeur des paramètres p et q, le graphe de la courbe peut prendre plusieurs formes. Quelques exemples sont donnés dans les figures ci-dessous.

Figure 1 : Exemple de courbes elliptiques sur le corps des réels



Ces courbes sont symétriques par rapport à un axe horizontal (car y n'intervient que par son carré y^2) et chacune d'elles a la caractéristique commune à toutes les courbes algébrique du troisième degré : si une droite coupe la courbe en deux points, elle la recoupe en un troisième point. Pour que ce soit vrai dans tous les cas, on considère qu'une tangente à la courbe la coupe en deux points confondus et on rajoute à la courbe un point à l'infini noté Ω par lequel passera toute verticale (qui recoupe la courbe en deux points symétriques).

Dans toute la suite, on éliminera les courbes elliptiques ayant un point double comme la troisième de celles présentées en exemple.

Sur l'ensemble des points de la courbe, on peut définir une loi de composition (que nous noterons par le symbole « ° » pour éviter de la confondre avec une addition ou une multiplication classique.

Étant donnés deux points A et B situés sur la courbe elliptique, on peut construire leur composé A°B par la construction géométrique suivante : on construit la droite joignant A et B (ou, s'ils sont confondus en un même point, la tangente à la courbe en ce point). Cette droite recoupe toujours la courbe en un troisième point. Le résultat A°B est le symétrique de ce troisième point.

La figure 2 ci-après illustre cette construction. Soit S1 le résultat de l'opération A°B . On construit tout d'abord S1' , situé sur la courbe et aligné avec A et B, puis S1, symétrique du précédent..

Notons que si A et B sont symétriques, la droite qui les joint est verticale et elle recoupe la courbe au point à l'infini Ω . Ce point à l'infini est considéré comme son propre symétrique.

La loi ° ainsi définie est une loi de groupe abélien :

1. La loi est associative, c'est-à-dire étant donnés trois points A, B et C on a toujours la relation (A°B)°C = A°(B°C). C'est relativement long à démontrer, mais on peut facilement le vérifier géométriquement sur la figure 2 : on construit tout d'abord A°B = S1 puis S1°C = T par la méthode décrite précédemment ; d'un autre côté, on construit A°C = S2, puis S2°B et on vérifie qu'on retombe bien sur le point T.

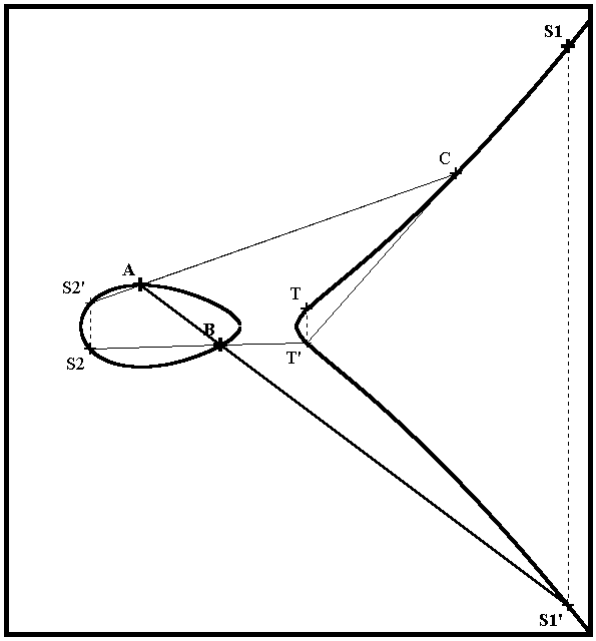
Les autres propriétés d'une loi de groupe sont très faciles à démontrer.

2. La loi « ° » est commutative (A ° B = B ° A) ce qui est géométriquement évident .

3. L'élément neutre est Ω (A ° Ω = Ω ° A = A , ce qui est presque aussi évident sur la figure).

4. Le symétrique A' du point A est son opposé (A ° A' = A'°A = Ω), ce qui est résulte immédiatement de la définition de la loi de composition « ° »)

Figure 2 : Loi de groupe sur les points d'une courbe elliptique réelle



L'exponentiation sur une courbe elliptique

Ayant défini cette loi de groupe, on peut construire l'équivalent de l'exponentiation. Etant donné un point G sur la courbe elliptique, et un entier positif n , on définit le point $G^{\circ n}$ comme étant la composée (par l'opération « ° ») de n points égaux à G. Lorsque l'exposant n est très grand, il est inutile de faire n opérations « ° » successives. On peut appliquer un algorithme similaire à celui décrit en arithmétique modulaire et consistant calculer les carrés successifs de G (pour la loi « ° »)

Le principe de l'utilisation des courbes elliptiques en cryptographie.

Connaissant la loi de groupe sur une courbe elliptique, telle que décrite précédemment, on peut transposer les protocoles utilisés en arithmétique modulaire. Il faut faire l'hypothèse que, si le calcul d'une exponentiation peut se faire rapidement, le problème inverse est considérablement plus difficile. C'est l'équivalent en courbes elliptiques du problème du logarithme discret de l'arithmétique modulaire. Connaissant G et $G^{\circ n}$, pour n assez grand, il est impossible de retrouver en un temps raisonnable la valeur de l'exposant n .

Les protocoles mis en œuvre sont calqués sur les protocoles correspondants en l'arithmétique modulaire. Détaillons par exemple Diffie Hellman.

Au départ les interlocuteurs se mettent d'accord sur la courbe elliptique à utiliser et sur un point G de cette courbe. Cette information est publique. Puis les interlocuteurs Alice et Bob choisissent chacun de leur côté un nombre aléatoire secret a et b. Alice calcule $G^{\circ a}$ et l'envoie à Bob. De son côté, Bob calcule $G^{\circ b}$ et l'envoie à Alice. Les deux peuvent maintenant calculer et utiliser comme clé commune les coordonnées du point $G^{\circ ab}$ (car dans la structure de groupe abélien liés à la courbe elliptique, on a les relations $(G^{\circ a})^{\circ b} = (G^{\circ b})^{\circ a} = G^{\circ ab}$). Mais le pirate connaissant G, $G^{\circ a}$, $G^{\circ b}$ n'a aucun moyen de calculer cette clé.

Le choix de G

L'un des problèmes qui se pose est celui du choix initial d'un point G, les calculs de type Diffie-Hellman se passant sur le groupe des puissances successives de G. Ce groupe est-il fini ? Cela dépend de la courbe et du point G choisi. Il peut arriver qu'en listant les puissances successives de G, on retrouve des points déjà rencontrés. On démontre alors qu'il existe un entier r tel que $G^{\circ r} = \Omega$ (élément neutre de la loi « ° »). Le plus petit r pour lequel cela se produit est appelé ordre de r. Dans certains cas le groupe des puissances successives de G peut être infini.

Pour assurer la sécurité du schéma de Diffie Hellman, il faut que l'ordre de G soit grand . Il vaudrait mieux que ce soit un nombre à plusieurs centaines de bits. S'il vaut autour de 2^{30} ou 2^{40} , le protocole de Diffie Hellman n'offre plus aucune garantie de sécurité.

La difficulté des calculs précis sur des nombres réel

Une des grandes difficultés du travail sur des nombres réels ou complexes est le problème des erreurs d'arrondi. Les protocoles d'authentification ou d'échanges de clés basés sur le schéma de Diffie Hellman supposent que les deux interlocuteurs fassent chacun de leur côté un calcul différent comme par exemple $(G^{\circ a})^{\circ b}$ ou $(G^{\circ b})^{\circ a}$ dont la théorie prouve qu'ils doivent donner le même résultat.

Or lorsqu'on travaille sur des nombres réels ou complexes, en général on a des erreurs d'arrondi liés à la précision avec laquelle on représente ces nombres. Supposons qu'un nombre soit représenté avec une précision relative ε (ordre de grandeur de l'erreur d'arrondi) ; une fois ce nombre élevé à la puissance a, l'ordre de grandeur de l'erreur d'arrondi est de a ε (les résultats sont sensiblement similaires, qu'on travaille sur des nombres réels ou sur les coordonnées des points d'une courbe elliptique munie de l'opération « ° »). Lorsque les exposants a et b sont des grands nombres (512 bits par exemple), on voit qu'il faut stocker les nombres réels sur plus de 1024 bits pour avoir une précision suffisante. De plus une erreur infinitésimale sur les coordonnées de G peut transformer un point d'ordre fini de la courbe elliptique en un point d'ordre infini.

L'utilisation de points à coordonnées rationnelles

Une façon de tourner la difficulté des erreurs d'arrondis sur les nombres réels est de se limiter à des points rationnels, c'est-à-dire donc les coordonnées sont rationnelles (représentées par une fraction dont le numérateur et le dénominateur sont des nombres entiers). Lorsque les coefficients p et q de la courbe elliptique sont rationnels, la composée de deux points rationnels est encore un point rationnel, comme nous le verrons au paragraphe suivant. On peut ainsi faire des calculs exacts.

Le problème de la précision reste cependant posé. Très rapidement les fractions qu'il faudra utiliser auront des numérateurs et dénominateurs tellement grands qu'ils seront impossibles à manier sur ordinateurs. En effet, le nombre de chiffre nécessaires pour représenter ces entiers dépassera très rapidement le nombre d'atomes de l'univers, et a plus forte raison l'espace mémoire de l'ensemble des ordinateurs de l'univers.

Un autre problème est un problème purement mathématique : y a-t-il assez de points rationnels sur une courbe elliptique pour permettre notre application. Le problème des points rationnels sur une courbe algébrique est un problème mathématique extrêmement passionnant. Citons au passage le cas de la courbe d'équation $y^n = 1 - x^n$, pour n>2. Cette courbe n'a pas de points rationnels autres que ceux des axes (x=0 et y=0). Ce résultat, conjecturé en 1637 par Fermat, n'a été démontré en 1995 par Wiles. Notons que Fermat le présentait sous la forme $a^n + b^n = c^n$ avec a, b et c entiers, ce qui est clairement équivalent (en posant x=a/c et y=b/c).

Utilisation de courbes elliptiques implicites

Nous avons annoncé que si deux points d'une courbe elliptique étaient rationnels, il en est de même de leur composée (au sens de la loi « ° »). En fait, on a des résultats bien plus forts.

1. Si on se donne deux points rationnels $M_1=(x_1,y_1)$ et $M_2=(x_2,y_2)$ distincts et non symétriques (c'est-à-dire vérifiant $x_1 \neq x_2$), il existe une et une seule courbe elliptique d'équation de la forme $y^2 = x^3 + p\,x + q$ passant par ces deux points, et les coefficients p et q sont eux-mêmes rationnels.

2. Les coordonnées (x_3,y_3) du point composé $M_3 = M_1 \circ M_2$ peuvent se calculer directement à partir des coordonnées de M_1 et M_2 sans avoir à calculer l'équation de la courbe. La méthode de calcul est la suivante :

   Calcul de la pente de la droite $M_1 \circ M_2$: $\lambda = \dfrac{y_2 - y_1}{x_2 - x_1}$

   Calcul des coordonnées de M_3 : $x_3 = \lambda^2 - x_1 - x_2$

   $y_3 = -y_1 + \lambda\,(x_1 - x_3)$

3. Si se donne un point rationnel $M_1=(x_1,y_1)$ et un nombre rationnel λ , il existe une et une seule courbe elliptique d'équation de la forme $y^2 = x^3 + p\,x + q$ passant par M_1 et dont la pente en M_1 est égale à λ .Les coefficients p et q sont eux-mêmes rationnels. Il est possible de calculer la composée $M_1 \circ M_1$ sans calculer l'équation de la courbe, par des formules similaires aux formules précédentes (en faisant $M_1=M_2$)

Ces formules montrent que l'on reste toujours dans le monde des nombres rationnels

5.2. Les courbes elliptiques complexes

Plaçons nous maintenant dans le corps $\mathbb{C}$ des nombres complexes. L'équation est la même que précédemment : $y^2 = x^3 + p\,x + q$. Elle est du troisième degré en x et du second degré en y. Pour toute valeur de y, il y a 3 valeurs de x qui satisfont l'équation (en tenant éventuellement compte de racines multiples). Toute droite horizontale coupe donc la courbe en 3 points. Toute droite verticale coupe la courbe en deux points symétriques sur la courbe. Si on rajoute à la courbe le point à l'infini Ω , on considère que ce point est sur toutes les verticales, ce qui donne un troisième point d'intersection. Plus généralement, on démontre que toute droite coupe la courbe en 3 points (en comptant pour deux un point où la droite est tangente à la courbe, et en tenant comte du point Ω commun à toutes les verticales).

Lorsque les coefficients p et q sont réels (c'est-à-dire à partie imaginaire nulle) , on démontre que si deux des trois points d'intersection sont réels (ont des coordonnées réelles) , il en est de même du troisième.

Cercles et tores, Fonctions périodiques et doublement périodiques

L'anneau $\mathbb{Z}$ des entiers relatifs est un sous-anneau du corps $\mathbb{R}$ des nombres. Comme en arithmétique modulaire, on peut définir un ensemble quotient $\mathbb{R}/\mathbb{Z}$ en considérant que deux nombres sont égaux dès que leur différence est un nombre entier. On a une bijection évidente entre $\mathbb{R}/\mathbb{Z}$ et l'ensemble des points d'un cercle. Plaçons nous par exemple dans un repère orthonormé et considérons le cercle centré à l'origine et de rayon 1. On peut alors associer au nombre réel z (défini à l'addition près d'un nombre entier) le point d'angle polaire $2\pi z$ sur le cercle (angle qui est défini à un multiple de 2π près) c'est-à-dire le point de coordonnées ($\cos 2\pi z$, $\sin 2\pi z$).

Travailler dans $\mathbb{R}/\mathbb{Z}$ revient en fait à ne conserver que la partie fractionnaire d'un nombre, (en pratique, à ne travailler qu'avec des nombres z vérifiant $0 \le z < 1$). La partie fractionnaire de la somme de deux nombres est égale à la somme de leur partie fractionnaire, donc on peut additionner des classes d'équivalence. On dit que l'addition passe au quotient et définit une structure de groupe additif sur $\mathbb{R}/\mathbb{Z}$. Cependant la multiplication ne passe pas au quotient, contrairement à ce qui se passait pour l'arithmétique modulaire.

Rappelons qu'une fonction f de variable réelle est dite périodique de période ω si on a, pour tout nombre réel z, la relation $f(z+\omega) = f(z)$. On a alors, pour tout entier k la relation $f(z+k\omega) = f(z)$. Pour simplifier, on supposera que ω vaut 1. Comme une fonction périodique de période 1 prend la même valeur sur tous les nombres ayant même partie fractionnaire, on peut, à toute fonction périodique de période 1, associer une fonction définie sur $\mathbb{R}/\mathbb{Z}$ et réciproquement.

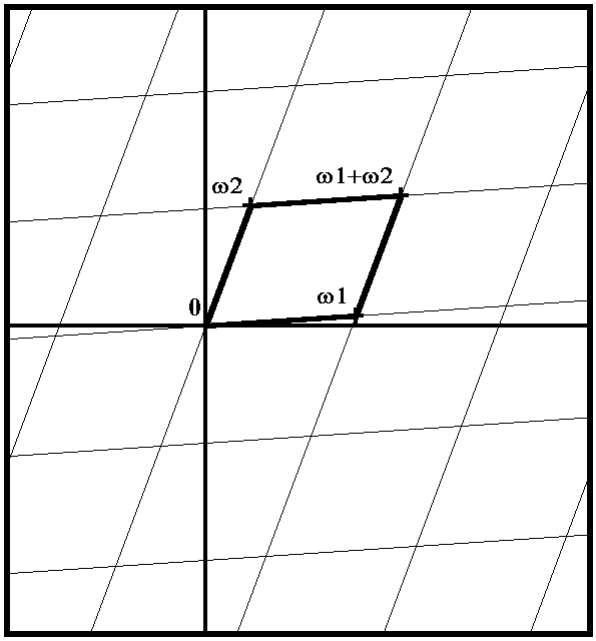
Lorsqu'on travaille avec des variables complexes, ces notions se généralisent, en remplaçant « périodique » par « doublement périodique », « cercle » par « courbe elliptique », « fonction cosinus » par « fonction $\wp$ de Weierstrass ».

On se donne au départ deux nombres complexes non nuls ω_1 et ω_2 dont le quotient a une partie imaginaire non nulle, ce qui revient à dire que les points ω_1 et ω_2 du plan complexe ne sont pas alignés avec l'origine. On dit qu'une fonction f de variable complexe est doublement périodique de périodes ω_1 et ω_2 si on a, pour tout nombre complexe z, les relations $f(z+\omega_1) = f(z)$ et $f(z+\omega_1) = f(z)$. On a alors, pour tout couple d'entiers k_1 et k_2 la relation $f(z+k_1\omega_1+k_2\omega_2) = f(z)$.

Considérons l'ensemble de leurs combinaisons linéaires entières des périodes ω_1 et ω_2 , c'est-à-dire l'ensemble Λ formé des nombres complexes de la forme $m\,\omega_1 + n\,\omega_2$ avec m et n entiers. Cet ensemble est noté $\Lambda = \mathbb{Z}\,\omega_1 + \mathbb{Z}\,\omega_2$. Dire qu'une fonction f est doublement périodique de périodes ω_1 et ω_2 revient à dire que $f(z+\lambda)=f(z)$ dès que $\lambda\in\Lambda$. Cet ensemble forme un réseau parallélogrammique dans le plan complexe, comme on le voit sur la figure 3.

Comme précédemment, on peut définir un ensemble quotient $\mathbb{C}/\Lambda$ en considérant que deux nombres sont égaux dès que leur différence est dans Λ. Travailler dans $\mathbb{C}/\Lambda$ revient à considérer comme égaux deux nombres complexes dont la différence est dans Λ, et, en pratique, à ne travailler qu'avec des nombres z situés dans le parallélogramme engendré par l'origine et les points ω_1 et ω_2 du plan complexe. Dès qu'on sort de ce parallélogramme, on ajoute ou soustrait un nombre adéquat de fois les périodes ω_1 et ω_2 de façon à se retrouver dans ce parallélogramme. L'addition passe au quotient et permet de définir une structure de groupe additif sur $\mathbb{C}/\Lambda$. Cependant, comme dans le cas de $\mathbb{R}/\mathbb{Z}$, la multiplication ne passe pas au quotient.

Figure 3 : Réseau parallélogrammique associé à une fonction doublement périodique



Le groupe $\mathbb{C}/\Lambda$ a topologiquement la forme d'un tore. Pour le voir, il suffit de considérer le parallélogramme formé des points 0 , ω_1 , ω_2 , $\omega_1+\omega_2$ du plan complexe (voir figure 3) et d'imaginer qu'on le replie de façon à recoller le segment $0\ \omega_1$ avec le segment $\omega_2\ \omega_1+\omega_2$ et le segment $0\ \omega_2$ avec le segment $\omega_1\ \omega_1+\omega_2$. Le tore apparaît comme l'équivalent du cercle, mais à deux dimensions. Un cercle peut être obtenu par un segment de droite qu'on replie de façon à faire coïncider ses extrémités. Un tore est obtenu en repliant un parallélogramme de façon à faire coïncider ses côtés parallèles.

A toute fonction doublement périodique de périodes ω_1 et ω_2 , on peut associer une fonction sur le tore $\mathbb{C}/\Lambda$ et réciproquement.

Fonctions doublement périodiques et courbes elliptiques complexe

Donnons nous un réseau $\Lambda = \mathbb{Z}\,\omega_1 + \mathbb{Z}\,\omega_2$. A ce réseau est associée une fonction doublement périodique, appelée fonction $\wp$ de Weierstrass, du nom du mathématicien du XIX^{ième} siècle qui l'a introduite. Elle n'est pas définie pour les points du réseau. Pour $z \notin \Lambda$, elle est définie comme somme de la série

$$\wp(z) = \frac{1}{z^2} + \sum\left[\frac{1}{(z-m\,\omega_1-n\,\omega_2)^2} + \frac{1}{(m\,\omega_1+n\,\omega_2)^2} \right]$$

La somme est étendue à toutes les valeurs du couple (m,n), à l'exception de (0,0).

Il est facile de voir que cette série converge pour toute valeur de z, et que la somme est doublement périodique de périodes ω_1 et ω_2 .

On montre que la fonction $\wp(z)$ et sa dérivée $\wp'(z)$ sont reliés par une équation de la forme

$$\wp'(z)^2 = \wp(z)^3 + p\,\wp(z) + q$$

où les nombres p et q peuvent être calculés à partir des périodes ω_1 et ω_2 .

Pour $z \notin \Lambda$, le point de coordonnées $(\wp(z), \wp'(z))$ se trouve donc sur la courbe elliptique complexe d'équation $y^2 = x^3 + p\,x + q$. Lorsque z se rapproche d'un point du réseau Λ, les nombres $\wp(z)$ et $\wp'(z)$ tendent vers l'infini, et le point correspondant sur la courbe elliptique tend vers Ω, point à l'infini de la courbe.

Les fonctions $\wp$ et $\wp'$ sont doublement périodiques de réseau de périodes Λ, elles passent au quotient, et on peut les définir sur le tore $\mathbb{C}/\Lambda$ (à l'exception du point 0 du tore correspondant aux points du réseau Λ sur lesquels ces fonctions ne sont pas définie). A chacun des points du tore (c'est-à-dire des classes d'équivalence modulo Λ) correspond un point de la courbe elliptique complexe. En associant au point 0 du tore le point à l'infini Ω de la courbe elliptique, on obtient une fonction définie sur le tore tout entier.

On démontre que la fonction ainsi définie est une bijection : à tout point du tore correspond un point unique de la courbe elliptique (complétée par son point à l'infini) et réciproquement. On a une propriété bien plus forte : étant donnés deux nombres complexes c et b (définis modulo le tore Λ) et leur somme c=a+b (définie elle aussi modulo Λ), si l'on considère leurs images A,B et C sur la courbe elliptique, on a la relation C = A°B pour la loi de groupe sur les points de la courbe elliptique, telle que nous l'avions définie précédemment. On dit qu'on a un isomorphisme de groupes. (Remarquons au passage que l'image de 0 (élément neutre de l'addition des nombres complexes) est le point à l'infini Ω de la courbe elliptique, (élément neutre de la loi de groupe ° définie sur la courbe)

Réciproquement, supposons donnée une courbe elliptique complexe d'équation $y^2 = x^3 + p\,x + q$ sans point double (pour ce faire, les paramètres p et q doivent vérifier la relation $4\,p^3 + 27\,q^2 \neq 0$). On peut alors trouver un couple de nombres complexes ω_1 et ω_2 tels que la fonction $\wp$ de Weierstrass associée au réseau $\Lambda = \mathbb{Z}\,\omega_1 + \mathbb{Z}\,\omega_2$ permette de construire une bijection entre les points de la courbe elliptique et le tore $\mathbb{C}/\Lambda$

La fonction $\wp$ joue pour les courbes elliptiques le même rôle que la fonction cosinus pour le cercle.

A un nombre réel z défini modulo $\mathbb{Z}$ on associe le point de coordonnées (cos(x), sin(x)) situé sur un cercle, réalisant ainsi une bijection entre les points du cercle et l'ensemble quotient $\mathbb{R}/\mathbb{Z}$. (Notons qu'au signe près, la fonction sinus est la dérivée de la fonction cosinus).

A un nombre complexe z défini modulo Λ on associe le point de coordonnées $(\wp(z), \wp'(z))$ situé sur une courbe elliptique complexe, réalisant ainsi une bijection entre les points de cette courbe elliptique l'ensemble quotient $\mathbb{C}/\Lambda$.

Comment casser le logarithme discret sur une courbe elliptique

Comme nous l'avons vu, toute la sécurité d'un cryptosystème de type Diffie-Hellman est basé sur l'impossibilité de résoudre en un temps raisonnable le problème du logarithme discret, c'est-à-dire de retrouver l'exposant n connaissant G et $G^{\circ n}$.

Si on travaille sur les courbes elliptiques complexes on semble avoir une piste pour résoudre ce problème en utilisant l'isomorphisme de groupes décrit précédemment entre le tore $\mathbb{C}/\Lambda$ et les points de la courbe elliptique. Au point G on peut associer un nombre complexe z, construit de façon que $\wp(z)$ et $\wp'(z)$ soient les coordonnées de G ($\wp$ est la fonction de Weierstrass associée au réseau Λ).

Le point $G^{\circ a}$ est simplement le point associé au nombre az. Le passage de G à $G^{\circ a}$ est une simple multiplication d'un nombre complexe par un nombre entier a. Le passage réciproque de $G^{\circ a}$ à G est une simple division par l'entier a. Mais les choses ne sont pas si simples. Comme nous l'avons vu, les calculs doivent être faits avec une très grande précision. Or les fonctions $\wp$ et $\wp'$ sont des fonctions transcendantes, c'est-à-dire des fonctions qui ne peuvent pas s'exprimer à l'aide d'opérations élémentaires (additions, multiplications, divisions, exponentiations…). Pour les calculer on a recours à des sommes de séries infinies ou à des intégrales. On peut certes, faire sur ordinateur les calculs avec la précision qu'on souhaite, mais si cette précision doit être de l'ordre du millier de bits, les temps de calcul deviennent rapidement prohibitifs.

Le problème posé est plus compliqué qu'une simple multiplication ou division dans $\mathbb{C}$. En effet, il s'agit, connaissant G et $G^{\circ a}$, de retrouver a. Soient z et w les représentants de ces nombres dans $\mathbb{C}$ .On veut trouver un entier a tel qu'on ait $w = a\,z$. Ce n'est pas si simple qu'il n'y paraît car les nombres w et z sont en fait défini à l'addition près d'un élément quelconque du réseau Λ c'est-à-dire d'un nombre $m\,\omega_1 + n\,\omega_2$. avec m et n entiers. Il y a fort peu de chances qu'on tombe du premier coup sur deux nombres w et z tels que le rapport w/z soit entier. Il faut tâtonner en essayant diverses valeurs du couple (m,n) de façon à rendre entier le rapport $(w + m\,\omega_1 + n\,\omega_2)/z$, , et ce sera aussi long que de rechercher la valeur de a, par tâtonnement en calculant directement sur la courbe elliptique les puissances successives de G.

Les courbes elliptiques sur des corps finis

En pratique, on travaille sur des courbes elliptiques sur des corps finis. On dit qu'on a une structure de corps sur un ensemble si on a défini sur cet ensemble une loi d'addition et une loi de multiplication ayant des propriétés (commutativité, associativité,…) similaires à celles de l'arithmétique classique.

Sur un ensemble fini à q éléments, on peut définir une structure de corps si et seulement si q est un nombre premier ou une puissance d'un nombre premier ($q = p^f$ avec p premier, f entier). Et on démontre en outre que ces structures sont toutes isomorphes entre elles. On appellera $\mathbb{F}_q$ un ensemble à q éléments muni d'une telle structure. Lorsque q est le nombre premier p, cette structure de corps n'est autre que celle de l'arithmétique modulaire dans $\mathbb{Z}/p\mathbb{Z}$ vue précédemment.

Sur $\mathbb{F}_q$, si on additionne p termes égaux à 1 (élément neutre de la multiplication), on retrouve, comme dans $\mathbb{Z}/p\mathbb{Z}$, le nombre 0 (élément neutre de l'addition) : 1+1+1+1+…+1 = 0. Et p est le plus petit nombres pour lequel cela se produit. On dit que le corps est de caractéristique p.

Si la caractéristique p est différente de 2 ou de 3, les propriétés des courbes elliptiques sont sensiblement les mêmes sur $\mathbb{F}_q$ que sur $\mathbb{R}$. On appellera courbe elliptique l'ensemble des couples (x, y) de $\mathbb{F}_q$ vérifiant l'équation $y^2 = x^3 + p\,x + q$, (où p et q sont des éléments de $\mathbb{F}_q$), auxquels on ajoute un point à l'infini Ω.

On définit le composé de deux points de la courbe elliptique de la même façon que précédemment, de même que la notion d'exponentiation, et le problème du logarithme discret s'exprime dans les mêmes termes. Les protocoles cryptographiques de type Diffie-Hellman prennent alors la même forme que précédemment.

Si on travaille sur un corps $\mathbb{F}_q$ avec q très grand, le temps de calcul pour retrouver a connaissant G et $G^{\circ a}$ deviennent très importants ce qui permet de baser un cryptosystème très robuste sur ce modèle.

Les meilleurs algorithmes connus pour résoudre le problème du logarithme discret semblent être en un temps polynomial en q, c'est-à-dire de l'ordre de $q \approx \exp(\ln q)$ ou de $\exp(O(\ln q))$, et il semble qu'on ne connaisse actuellement de technique « rapide » telle celles qui, en arithmétique modulaire classique ramènent ce temps à quelque chose de l'ordre de $\exp(O(\sqrt[3]{\ln q \ \ln\ln^2 q}))$.

Mais les choses évoluent très vite en ce domaine, et les systèmes cryptographiques sont à la merci d'un mathématicien génial qui trouvera un algorithme plus rapide.