

# Cryptographie sur les courbes elliptiques

S. Duquesne

Ecole Jeunes Chercheurs  
5 avril 2005

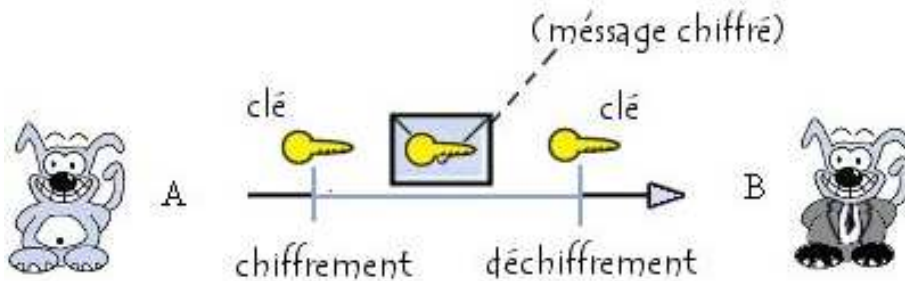
## 1 Introduction

### 1.1 Qu'est ce que la cryptographie ?

Donnons tout d'abord la définition trouvée dans un dictionnaire usuel:

**cryptographie** n. f. Ensemble des principes, méthodes et techniques dont l'application assure le chiffrement et le déchiffrement des données, afin d'en préserver la confidentialité et l'authenticité.

Le principe de base de la cryptographie est donc de pouvoir communiquer secrètement et peut être schématisé par le dessin suivant



L'homme a depuis longtemps eu de tels besoins et les premiers protocoles de chiffrement (c'est à dire les méthodes de chiffrement et de déchiffrement) ont été introduits par ceux qui en avaient le plus besoin et qui en avaient le plus les moyens : les militaires (ce qui est d'ailleurs toujours le cas). On parle souvent historiquement du chiffrement de César (décalage de lettres) mais ce n'est pas le plus ancien. Les grecs utilisaient par exemple des bandes de papier enroulées autour d'un bâton



### Exemples de chiffrement de César

Décalage de 3 lettres vers la droite:

A T T A Q U E Z L E P A L A I S  
 D W W D T X H C O H S D O D L V

Décalage d'une lettre vers la gauche (clin d'oeil cinématographique):

I B M  
 H A L

Notons que ce type de chiffrement est très facilement attaquable par une analyse de fréquence (la lettre apparaissant le plus souvent en français est le e, il suffit donc de chercher la lettre apparaissant le plus souvent dans le message pour "casser" la méthode.

Pendant longtemps des variantes du chiffrement de César ont été utilisées. On peut par exemple décider de décaler la première lettre de 3 lettres vers la droite, la deuxième d'une lettre vers la gauche, puis à nouveau la troisième de 3 lettres vers la droite et ainsi de suite. Le dernier exemple d'importance est la machine Enigma utilisée par l'Allemagne durant la seconde guerre mondiale



Ces méthodes ont toutes un point commun : la méthode de chiffrement elle-même est secrète.

**Inconvénients:**

- Elle n'est pas secrète pour tous ceux qui l'utilisent. Par exemple, si le chiffrement de César est connu de tous les généraux, César ne peut pas communiquer avec un de ses généraux sans que les autres ne le sache.
- Tout s'effondre le jour où elle est découverte. Ainsi les Allemands comp-taient en partie sur le fait que les alliés n'avaient pas accès à Enigma. La prise d'une machine Enigma (cf U-571, encore une référence cinemato-graphique même si celle là n'est pas du même niveau que la précédente) a été déterminante dans la victoire des alliés. L'autre chose sur laquelle ils comptaient était que le décryptage était infaisable à la main ce qui était vrai sauf qu'à ce moment un certain Turing a inventé le premier ordinateur.

En fait l'un des principes de la cryptographie moderne (principes de Kerchoffs au 19<sup>ème</sup> siècle) est le suivant:

**la sécurité d'un système de chiffrement ne doit résider que dans la clé et non dans le procédé de chiffrement.**

Cela a conduit à une certaine démocratisation du secret: la méthode de chiffrement peut maintenant être connue de tous et donc utilisée par tous.

## 1.2 La cryptographie pour faire quoi ?

La cryptographie moderne permet entre autres d'assurer

- la confidentialité (communications secrètes),
- l'authenticité (accès par mot de passe),
- la non-répudiation (signature),
- l'intégrité du message transmis

La cryptographie ne sert bien sûr pas qu'à ça mais ce sont les applications princi-pales. De nombreux protocoles existent et permettent de résoudre des problèmes dont vous ne soupçonnez, et ne soupçonnerez, jamais l'existence.

## 1.3 Où trouve t'on de la cryptographie ?

Sans même forcement le savoir, la cryptographie est très présente dans la vie de tous les jours et tend à le devenir de plus en plus. Voici des domaines bien connus où elle est utilisée massivement

- Armée, Gouvernements qui sont bien sûr les mieux équipés
- Système bancaire. La machine Enigma a été initialement développée pour les banques et il y en a bien sûr dans les cartes bleues.

- Internet (achats, identification). En effet, les communications sur internet circulent dans des infrastructures où on ne peut pas garantir la fiabilité et la confidentialité, il faut donc les sécuriser.

Remarque importante:

Si, comme dans la plupart des cas, le site est sécurisé (un petit cadenas apparaît en bas ou en haut de votre navigateur) vous courrez infiniment moins de risque en effectuant vos achats sur internet qu'en mettant le pied dehors (et je ne parle pas de retirer de l'argent à un guichet automatique). Il faut bien se dire que si quelqu'un était capable de casser le protocole cryptographique qui protège votre numéro de carte bancaire il aura bien mieux à faire (niveau rentabilité) que de s'en servir pour vous attaquer ...

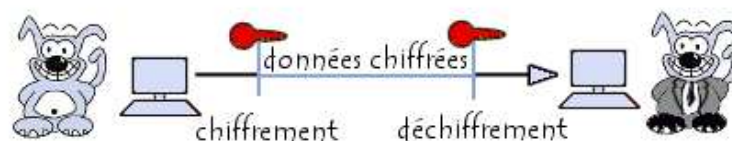
- Téléphones portables
- TV payante
- Cartes d'identités électroniques
- Vote électronique

Au même titre que les protocoles précédemment évoqués, cette liste n'est pas du tout exhaustive. On peut par exemple trouver des protocoles cryptographiques très complexes dans un hôpital pour les dossiers médicaux. En effet ceux ci doivent être tenus secret. Enfin pas secret pour tout le monde sinon vous ne serez jamais soigné et surtout ils ne sont pas secret au même niveau pour tous les personnels de l'hôpital. Quand on voit le nombre de personnes avec leurs compétences différentes (et donc un niveau d'accessibilité au dossier différent) qui gravitent autour d'un patient et le nombre de patients qu'il peut y avoir dans un hôpital, on se dit, comme probablement la plupart des hôpitaux que c'est quand même plus simple quand le dossier n'est pas secret et que la femme de ménage sait que votre arrière grand mère n'était pas mariée quand elle a eu votre grand père. Heureusement que les américains sont là pour faire des procès aux hôpitaux et donc donner du travail aux cryptographes.

Je crois que ceci est un très bon exemple d'endroit où vous n'auriez jamais pensé trouver de la cryptographie évoluée, de problème cryptographique que vous n'auriez jamais imaginé et de l'importance de la cryptographie dans les décennies à venir.

## 2 Cryptographie symétrique/asymétrique

Il existe essentiellement deux types de cryptographie. Le premier type est le plus intuitif et le plus naturel. On l'appelle la cryptographie symétrique ou à clé privée. Le principe est de dire que les correspondants A et B ont tous les deux la même clé qui sert à la fois à chiffrer et à déchiffrer le message.



Pour donner un équivalent pratique, A et B vont ensemble au supermarché acheter un coffre fort et gardent chacun un double de la clé. Lorsqu'il veulent communiquer, ils mettent un message dans le coffre fort et l'envoient à l'autre par la poste qui peut l'ouvrir et lire le message grâce a sa clé.

Les systèmes les plus connus sont le DES, le triple-DES, l'AES (remplaçant du DES), et le masque jetable.

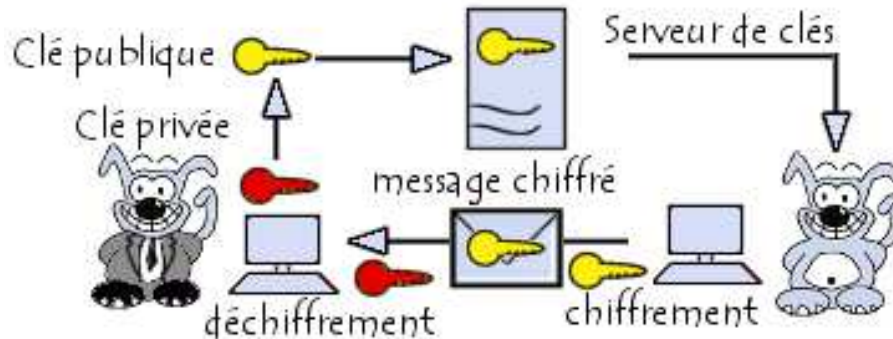
Les trois premiers sont essentiellement basés sur un mélange des bits du message avec les bits de la clé (c'est grossier mais c'est l'idée). Le dernier c'est essentiellement le chiffrement de César avec une clé de la même taille que le message et qu'on jette après utilisation : on additionne bêtement le message avec la clé et une fois que c'est fait on jette la clé, il faut donc beaucoup de clés et surtout elles doivent être de la même taille que le texte de départ. C'était la méthode utilisée pour le téléphone rouge.

Notons qu'il est prouvé que c'est le seul système qui soit totalement sûr. Tous les autres systèmes sont basés sur des difficultés calculatoires insurmontables avec les moyens actuels.

La cryptographie symétrique présente deux inconvénients majeurs. Le premier est qu'il faut avoir une clé pour chaque correspondant. En effet si A communique avec B et C avec la même clé, C peut intercepter un message que A envoie à B. Ça peut vite devenir fort peu pratique. Mais le plus gros inconvénient est qu'il faut s'échanger une clé (il faut que A et B aille au supermarché ensemble). Or le but étant de s'envoyer des messages chiffrés, il est bien sûr ridicule de s'envoyer la clé par courrier ou tout autre moyen non sécurisé. Pour le téléphone rouge c'est la valise diplomatique qui était utilisée. Mais le problème reste entier pour tous ceux qui ne disposent pas de tels moyens et qui ne veulent pas communiquer qu'avec leur voisin de pallier (et encore en supposant que le gérant du supermarché, la caissière ou votre concierge n'a pas subrepticement fait un double de la clé).

Pour résoudre ce problème on utilise donc ce qu'on appelle la cryptographie a clé publique (ou asymétrique). Le principe est cette fois que la clé de chiffrement est différente de la clé de déchiffrement. Ainsi A publie sa clé de chiffrement (clé publique) de telle sorte que B puisse l'utiliser pour chiffrer le message qu'il veut envoyer à A. A peut alors déchiffrer le message avec sa clé de déchiffrement qu'il est le seul à connaître (contrairement à la cryptographie symétrique où B sait aussi déchiffrer). De cette manière, A ne parle jamais à personne de sa clé de déchiffrement et aucune fuite n'est possible (sauf si elle est écrite sur un post-it

dans son portefeuille ou tatoué sur l'oreille de son chien).



Pour reprendre l'image des coffres forts, A va dévaliser le supermarché de coffres forts et pour simplifier le problème on suppose qu'ils s'ouvrent tous avec la même clé (si c'est un code par exemple avec les grosses molettes qui tournent comme dans les films). A ouvre ensuite tous ses coffres et les dépose sur le trottoir devant le bureau de poste local (en espérant que l'Europe n'a pas libéralisé la poste et que le bureau n'a pas été fermé ...). Si B veut envoyer un message à A, il lui suffit de se rendre à la poste, de prendre un coffre fort sur le trottoir, de mettre son message dedans, de refermer le coffre et de l'envoyer à A. A peut alors ouvrir le coffre quand il le reçoit 3 mois plus tard par UPS (qui entre temps a fait un procès à la poste parce qu'elle avait le monopole du transport de coffre fort et que c'est anti-concurrentiel parce que le fait que les coffres soient déposés devant la poste incite les gens à utiliser la poste).

Ce principe a été introduit par Diffie et Hellman en 1976 et utilise une fonction à sens unique (clé publique) avec trappe (clé privée). Une fonction à sens unique est une fonction facile à calculer dans un sens (chiffrement rapide) et extrêmement difficile à calculer dans l'autre sens (déchiffrement) sauf si on connaît la clé (trappe).

L'exemple le plus connu d'un tel système est le système RSA du nom de ses inventeurs (Rivest, Shamir et Adleman). Il est basé sur le fait qu'il est facile de multiplier deux nombres premiers entre eux mais difficile de les retrouver étant donné le produit. La clé publique est dans ce cas essentiellement le produit et la clé privée les deux nombres premiers. On peut trouver de nombreuses références sur RSA étant donné qu'il est extrêmement utilisé et qu'il figure au programme du Bac S option mathématiques. Le but de ce cours est en fait de s'intéresser à un autre problème difficile à résoudre, le problème du logarithme discret, en particulier sur les courbes elliptiques.

La cryptographie asymétrique permet donc de résoudre les problèmes de la cryptographie symétrique (il n'y a pas d'échange de clé et A n'a qu'une seule clé publique que tous ses correspondants peuvent utiliser). Malheureusement,

elle présente un inconvénient à son tour, elle est 100 à 1000 fois plus lente que la cryptographie symétrique ce qui la rend particulièrement inapte à chiffrer des gros messages (photos, vidéo (TV payante), son (téléphone sécurisé) ...)

En pratique les deux types sont mélangés : On utilise dans un premier temps la cryptographie asymétrique pour s'échanger une clé privée commune puis on utilise cette clé privée commune pour échanger des messages chiffrés à l'aide de la cryptographie symétrique. C'est par exemple le cas de PGP pour le mail.

## 3 Le logarithme discret

### 3.1 Définition

Nous nous intéressons donc maintenant à un autre problème sur lequel on va pouvoir baser des protocoles à clé publique.

Soit  $G$  un groupe (noté additivement) cyclique fini d'ordre  $N$  engendré par un élément  $P$ .

Soit  $Q$  un élément de  $G$ . Comme  $G$  est un groupe cyclique engendré par  $P$ , il existe un unique entier  $n$  compris entre 1 et  $N$  tel que  $Q = nP$ . Cet entier  $n$  est appelé le logarithme discret de  $Q$  en base  $P$  et nous le noterons  $\log_P(Q)$ .

Usuellement ce problème est plutôt présenté pour un groupe noté multiplicativement ce qui donne :

Soit  $G$  un groupe (noté multiplicativement) cyclique fini d'ordre  $N$  engendré par un élément  $g$ .

Soit  $h$  un élément de  $G$ . Comme  $G$  est un groupe cyclique engendré par  $g$ , il existe un unique entier  $n$  compris entre 1 et  $N$  tel que  $h = g^n$ . Cet entier  $n$  est appelé le logarithme discret de  $h$  en base  $g$  et nous le noterons  $\log_g(h)$ . L'avantage de cette définition est qu'on comprend mieux d'où vient le nom de logarithme discret puisque si  $h$  et  $g$  sont des réels (ce qui n'est bien sûr pas le cas puisque  $\mathbb{R}$  est loin d'être fini), on a bien  $n = \frac{\log(h)}{\log(g)}$  c'est à dire le logarithme de  $h$  en base  $g$ . Cependant le groupe que nous utiliserons par la suite est usuellement noté additivement et nous garderons donc les notations initiales. Notez bien que  $nP$  signifie  $P + P + \dots + P$   $n$  fois où  $+$  est la loi du groupe  $G$ .

Schoup a démontré qu'un algorithme générique pour résoudre le problème du logarithme discret nécessitait au moins  $O(\sqrt{N})$  opérations de base sur le groupe. On entend par algorithme générique un algorithme utilisant uniquement la structure de groupe, autrement dit qui s'applique à un groupe quelconque. Des algorithmes plus efficaces peuvent exister dès lors qu'on a choisi un groupe spécifique avec des propriétés qui lui sont propres. Des algorithmes génériques permettent effectivement d'atteindre cette borne (rho pollard, pas de bébés-pas de géants).

Il peut donc être intéressant de baser des protocoles cryptographiques sur ce problème puisqu'à priori les attaques possibles seraient en la racine de la taille du groupe considéré c'est à dire exponentielle alors que les attaques connues pour RSA sont sous-exponentielles. Reste à trouver des bons candidats pour ces groupes pour lesquels il n'existe pas d'autres attaques que les attaques génériques (en tous les cas pas plus efficaces).

Un candidat souvent mentionné est le groupe multiplicatif d'un corps fini ( $G = \mathbb{F}_q^*$ ). Malheureusement, il existe une attaque sous exponentielle pour ces groupes et sa complexité est la même que les attaques de RSA et il ne présente donc pas un grand intérêt.

Il est donc nécessaire de trouver un groupe sur lequel le problème du logarithme discret n'a pas de solution meilleure que les algorithmes génériques. Dans l'état actuel des connaissances, les courbes elliptiques (et hyperelliptiques semblent procurer de tels groupes, mais avant de s'y intéresser, nous allons donner un exemple de protocole cryptographique basé sur le logarithme discret.

### 3.2 Utilisation en cryptographie

Nous avons vu précédemment que le problème majeur de la cryptographie symétrique était l'échange de clé entre A et B et que la cryptographie asymétrique permettait de résoudre ce problème. Il est donc naturel de présenter le protocole d'échange de clés de Diffie-Hellman.

- A choisit une clé secrète  $a \in [2, \dots, N - 1]$  et calcule  $aP$ . De même pour B,
- A envoie à B  $aP$  et B envoie à A  $bP$ ,
- A calcule  $a(bP)$  et B calcule  $b(aP)$ ,
- A et B ont une clé privée commune :  $abP$ .

Il existe bien sûr une multitude de protocoles basés sur le logarithme discret qui permettent de répondre à une multitude de problèmes concrets. Il en existent même qui permettent de répondre à des situations auxquelles RSA n'a pas apporté de solution.

## 4 Qu'est ce qu'une courbe elliptique ?

Les courbes elliptiques sont des objets mathématiques complexes et nous ne donnons donc ici qu'un aperçu de ce qu'elles sont et une infinitésimale partie de leurs propriétés. Il est toutefois amusant de noter qu'elles ont été introduites sans la moindre arrière pensée cryptographique pour résoudre des problèmes de mathématiques fondamentales voire même inutiles. Elles sont l'élément fondamental et essentiel de la récente résolution d'un des plus célèbres problèmes mathématiques : le grand théorème de Fermat (qui affirme que l'équation  $x^n +$

$y^n = z^n$  n'a aucune solution entière dès lors que  $n$  est supérieur à 3) lui aussi d'une totale inutilité. Créées et utilisées pendant 300 ans dans la plus totale inutilité des mathématiques fondamentales, elles sont aujourd'hui au centre de la cryptographie moderne et ont déjà rapporté bien plus d'argent qu'elles n'en ont coûté en maigres financements pour la recherche fondamentale. Maintenant que j'ai défendu mon steak haché au cas ou un futur ministre lirait ces lignes, définissons ces fameuses courbes elliptiques. Ici, nous les verrons comme:

Ensemble des couples  $(X, Y)$  (ou points) vérifiant

$$Y^2 + h(X)Y = F(X)$$

ou dans la plupart des cas

$$Y^2 = X^3 + aX + b$$

On choisit bien sûr un corps de base  $\mathbf{K}$ . Pour nous,  $\mathbf{K}$  sera un corps fini  $\mathbb{F}_{p^r}$  avec  $p$  premier et de manière générale, on choisira soit  $p = 2$  soit  $r = 1$ .

On dispose sur de tels objets d'une structure de groupe.

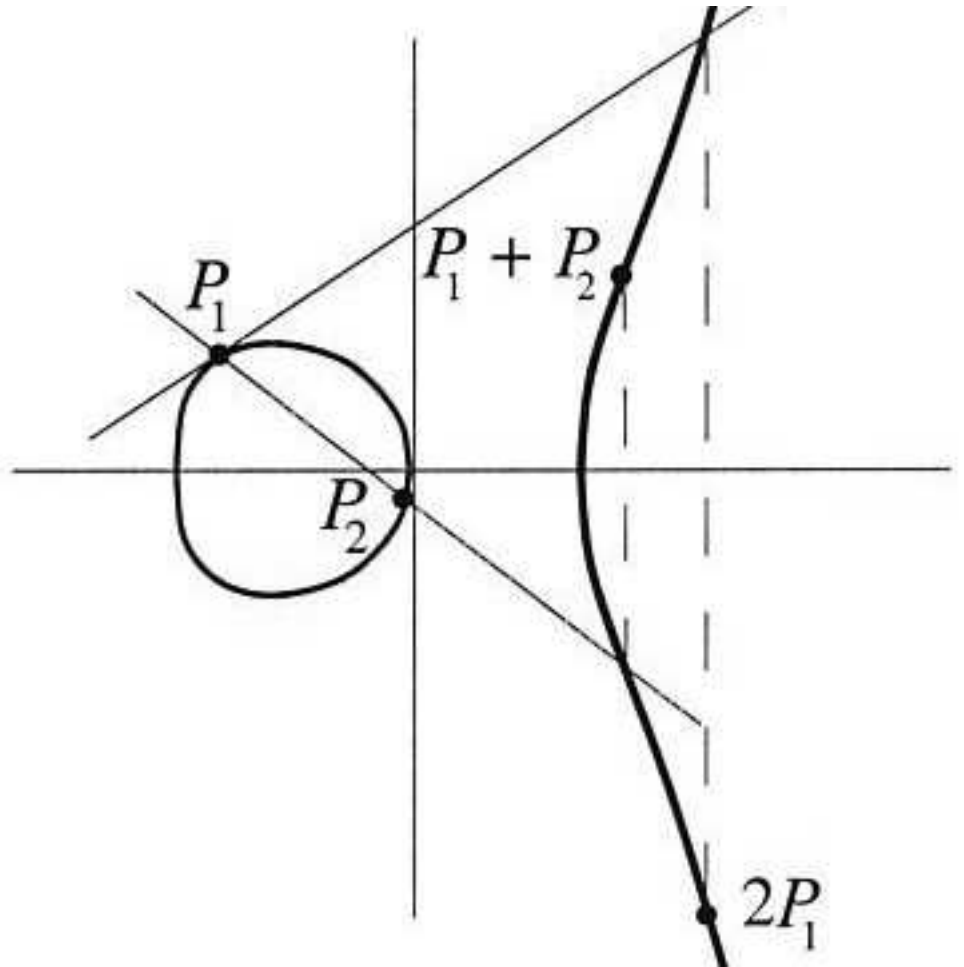
## 5 Loi de groupe sur une courbe elliptique

Usuellement la loi de groupe est définie géométriquement quand le corps de base  $\mathbf{K}$  est le corps des réels. Même si ce n'est pas la situation dans laquelle nous sommes placés, c'est la description la plus facile à comprendre. On peut donc additionner des points sur une telle courbe en utilisant la règle suivante :

$$P_1 + P_2 + P_3 = \mathcal{O} \iff P_1, P_2, P_3 \text{ sont alignés.}$$

où  $\mathcal{O}$  est le point à l'infini sur la courbe et l'élément neutre du groupe.

Ainsi pour additionner deux points  $P_1$  et  $P_2$  de la courbe elliptique, on trace la droite qui passe par ces deux points. Il est relativement facile de prouver que cette droite recoupe la courbe en un troisième point  $P_3$  (car le polynôme  $F$  est de degré 3). D'après la règle  $P_1 + P_2 = -P_3$  et il ne reste plus qu'à prendre l'opposé de  $P_3$  pour obtenir  $P_1 + P_2$ . Notons que dans la plupart des cas, l'opposé d'un point est simplement son symétrique par rapport à l'axe des abscisses. Enfin dans le cas où  $P_1 = P_2$  la droite qui passe par  $P_1$  et  $P_2$  est bien sûr la tangente à  $P_1$ .



Pour les calculs, il suffit alors de calculer les équations de cette droite et l'intersection avec la courbe en fonction des coordonnées des points  $P_1 = (x_1, y_1)$  et  $P_2 = (x_2, y_2)$  pour obtenir les coordonnées  $(x, y)$  de leur somme. Cela donne les formules suivantes :

$$\begin{aligned} \text{on calcule } \lambda &= \frac{y_1 - y_2}{x_1 - x_2} && \text{si } P_1 \neq P_2, \\ &= \frac{3x_1^2 + a}{2y_1} && \text{si } P_1 = P_2, \\ \text{et alors } x &= -x_1 - x_2 + \lambda^2, \\ y &= -y_1 + \lambda(x_1 - x). \end{aligned}$$

Ceci fait, ces formules sont bien sûr applicables à d'autres corps que le corps des réels et en particulier aux corps finis qui nous intéressent.

De part cette structure de groupe, on peut donc utiliser les courbes elliptiques comme base pour les protocoles basés sur le logarithme discret. Il faut cependant prendre quelques précautions de sécurité car certaines attaques existent tout de même.

## 6 Attaques

Notons tout d'abord que, du fait du théorème des restes chinois, la sécurité d'un groupe va être mesuré non pas par sa taille  $N$  mais par la taille du plus grand nombre premier  $\ell$  divisant  $N$ . Il est donc nécessaire dans un premier temps d'être capable de mesurer la taille du groupe, autrement dit combien de points y a-t'il sur une courbe elliptique donnée. Le théorème de Hasse permet d'en avoir une bonne approximation. En effet, il affirme que le nombre de points d'une courbe elliptique définie sur  $\mathbb{F}_q$  est compris entre  $q - 2\sqrt{q}$  et  $q + 2\sqrt{q}$ . Cela ne suffit cependant pas pour déterminer la taille du plus grand nombre premier divisant  $N$ . De nombreux algorithmes sont largement documentés dans la littérature et permettent de répondre assez rapidement à la question (SEA, Satoh, AGM, ...) mais restent mathématiquement difficilement accessibles. On supposera désormais dans la suite, pour plus de simplicité, que  $N$  est premier (pour trouver une courbe avec une telle propriété, il suffit d'en choisir une au hasard, de calculer le nombre de ses points et d'en choisir une autre si ce nombre n'est pas premier).

Il faut bien sûr se prémunir en premier lieu contre les attaques génériques qui sont incontournables. Nous avons vu que leur complexité était en  $O(\sqrt{N})$ . Ainsi, si l'on considère que  $2^{85}$  sont infaisables (c'est ce qui est communément admis de nos jours), on devra travailler sur un groupe possédant environ  $2^{170}$  éléments pour se protéger des attaques génériques. Dans le cadre des courbes elliptiques, il suffira pour cela de construire une courbe elliptique sur un corps à environ  $2^{170}$  éléments (le théorème de Hasse nous assurant que la courbe elle-même aura à peu près autant d'éléments). On dit dans ce cas qu'on travaille en 170 bits (ou ECC 170 pour Elliptic Curve Cryptography). A titre de comparaison, pour le même niveau de sécurité, il faut travailler en 1024 bits avec RSA ou le logarithme discret sur  $\mathbb{F}_q^*$ .

Idéalement, le travail devrait être terminé puisque les courbes elliptiques ont été choisies pour faire de la cryptographie car elles étaient optimales pour une utilisation du logarithme discret. Autrement dit, on ne connaît pas d'autres attaques que les attaques génériques. Ceci dit le travail acharné des chercheurs a tout de même permis d'exhiber quelques attaques plus efficaces que les attaques génériques mais nous allons voir qu'elles ne concernent qu'un petit nombre de courbes elliptiques et qu'elles sont facilement évitables.

La première de ces attaques est communément appelée MOV (pour Menezes,

Okamoto et Vanstone) ou Frey-Ruck du nom de leurs inventeurs. Dans les deux cas le principe est le même, à savoir utiliser une fonction (couplage de Weil pour MOV et couplage de Tate pour Frey-Ruck) pour transférer le problème du logarithme discret sur une courbe elliptique définie sur un corps fini  $\mathbb{F}_q$  à un problème de logarithme discret sur une extension  $\mathbb{F}_{q^k}^*$  de degré  $k$  de  $\mathbb{F}_q$ . Autrement dit, grâce à cette fonction on peut ramener un problème de logarithme discret en 170 bits à un problème de logarithme discret sur un corps fini en  $170 * k$  bits. Or nous avons déjà vu qu'il existe des attaques sous exponentielles pour résoudre ce genre de problème. Cette attaque est attrayante mais ne fonctionne que pour des petites valeurs de  $k$ . En effet, dans l'hypothèse par exemple d'une sécurité requise en 85 bits, dès que  $k > 6$  le problème du logarithme discret est plus difficile à résoudre sur un corps fini en  $170 * k$  bits que sur la courbe elliptique de 170 bits initiale. En fait, on peut démontrer que  $k$  est le plus petit entier tel que  $q^k - 1$  soit divisible par l'ordre de la courbe elliptique ce qui fournit un test très simple pour vérifier si une courbe elliptique donnée est sensible à cette attaque. En pratique,  $k$  est toujours très grand et l'attaque est donc inefficace sauf pour une certaine famille de courbes elliptiques, les courbes supersingulières (c'est à dire telles que  $\#E(\mathbb{F}_q) = q + 1$ ) qui sont très rares et bien connues. En fait cette attaque a eu plus d'applications positives que d'applications négatives. En effet ce transfert de logarithmes discret peut être utilisé pour construire de nouveaux protocoles impossibles à construire avec les autres outils de la cryptographie. C'est le cas par exemple de la cryptographie basée sur l'identité.

La plus sérieuse attaque est appelée la descente de Weil. Elle s'applique quand le corps de base est une extension d'un corps plus petit (par exemple  $155 = 31 * 5$  donc  $\mathbb{F}_{2^{155}}$  est une extension de degré 31 de  $\mathbb{F}_{2^5}$  ou une extension de degré 5 de  $\mathbb{F}_2^{31}$ ). On effectue dans cette situation une restriction aux scalaires qui consiste à écrire les composantes de l'équation de la courbe elliptique par rapport à cette extension. Pour clarifier, nous donnons un exemple de restriction aux scalaires qui ne correspond pas du tout à la cryptographie mais qui a l'avantage d'être très parlant.

**Exemple** Considérons sur  $\mathbb{C}$  la courbe d'équation

$$z^2 + 2z + 1 = 0.$$

On sait que  $\mathbb{C}$  est une extension de degré 2 de  $\mathbb{R}$ . On écrit donc  $z = a + ib$  et l'équation devient alors

$$a^2 - b^2 + 2abi + 2a + 2ib + 1 = 0.$$

En décomposant suivant les composantes, on obtient donc le système

$$a^2 - b^2 + 2a + 1 = 0 \text{ et } 2ab + 2b = 0.$$

On peut faire exactement la même chose sur les corps finis et c'est ce qui s'appelle la restriction aux scalaires. Ainsi l'équation d'une courbe elliptique sur  $\mathbb{F}_{2^{155}}$  peut être vue comme un système de 5 équations définies sur  $\mathbb{F}_{2^{31}}$ . Dans certains cas, ce système représente la jacobienne d'une courbe hyperelliptique de genre 5 (ou 31 si on regarde l'autre extension). On peut ainsi ramener le problème du logarithme discret sur une courbe elliptique définie sur  $\mathbb{F}_{2^{155}}$  à un problème de logarithme discret sur une jacobienne de courbe hyperelliptique de genre 5 (ou 31) définie sur  $\mathbb{F}_2^{31}$  (ou  $\mathbb{F}_2^5$ ). Or il se trouve que dès que le genre est supérieur ou égal à 4, il existe un algorithme (calcul d'indice) meilleur que les algorithmes génériques pour résoudre ce problème (cf le cours de Andreas Enge).

D'une façon générale, pour échapper à cette attaque, il faut choisir soit  $\mathbb{F}_p$  soit  $\mathbb{F}_{2^p}$  avec  $p$  premier comme corps de base ce qui est très peu contraignant. Si on tient vraiment à une extension composée (type  $\mathbb{F}_2^{155}$ ) il faut faire un peu plus attention à ce qu'aucun nombre premier de Mersenne n'apparaisse (c'est à dire de la forme  $2^p - 1$ . Encore des mathématiques inutiles qui finalement servent longtemps après (Mersenne est mort en 1648). Heureusement que les gens qui le finançaient n'attendaient pas des dividendes à la fin de chaque année) comme c'est le cas de 155 ( $31 = 2^5 - 1$  est un nombre premier de Mersenne). Des variantes de cette attaque existent et exigent qu'on exclu certains degrés d'extension. Attention, toutefois, même pour ces degrés d'extension, l'attaque ne fonctionne que sur une infime partie des courbes elliptiques définies sur ces corps.

Finalement, on peut dire qu'à quelques précautions près (le plus souvent par acquis de conscience car la probabilité de tomber sur une courbe attaquable est extrêmement faible), les courbes elliptiques restent, à ce jour, un excellent exemple de groupe sur lequel les meilleures attaques sont les attaques génériques (et donc exponentielles). Bien sûr cela n'est pas prouvé et les courbes elliptiques, comme les autres systèmes cryptographiques, ne sont pas à l'abri d'une nouvelle attaque.

## 7 ECC vs RSA

Pour conclure, nous donnons une comparaison de la cryptographie basée sur les courbes elliptiques avec RSA. RSA est actuellement le système de cryptographie à clé publique le plus utilisé. Néanmoins, nous avons déjà vu que les courbes elliptiques présentaient un certain nombre d'avantages et nous allons les expliciter plus précisément ici.

Tous ces avantages sont basés sur la différence de complexité pour résoudre les problèmes mathématiques sous-jacents. D'un côté la factorisation (et donc RSA) peut être effectuée à l'aide d'algorithmes sous-exponentiels tandis que de l'autre seuls des algorithmes exponentiels permettent de résoudre le problème du logarithme discret sur une courbe elliptique. Cela a pour première conséquence

que les clés sont plus courtes avec ECC qu'avec RSA. Typiquement, actuellement une clé inviolable doit avoir 1024 bits pour RSA contre 170 pour ECC. Mais le phénomène le plus remarquable est que cette différence de taille de clé va aller en s'amplifiant avec le temps du fait de la différence de complexité. Par exemple, pour une sécurité requise de 115 bits (c'est à dire qu'on considère que le système est sûr si il faut plus de  $2^{115}$  opérations pour le casser), les clés ECC seront de 230 bits tandis que les clés RSA seront de 2048 bits. On passe (entre 85 et 115 bits de sécurité) d'un rapport 6 entre les tailles de clé à un rapport 9. Cette différence de taille de clé induit de nombreux avantages (qui vont donc aller en s'amplifiant quand on augmente le niveau de sécurité requis). En particulier, les calculs sont à priori plus rapide et requièrent moins de mémoire.

Ainsi les premières applications des courbes elliptiques ont été pour les environnements restreints (PDA, téléphones mobiles, cartes à puces,...) qui ont peu de puissance de calcul (par exemple, traiter de 1024 bits avec un processeur 8 bits demande une multiprécision lourde) et surtout peu de mémoire disponible. En fait on s'est depuis rendu compte que même sur des machines puissantes, ECC était intéressante. Ainsi un serveur SSL peut il traiter de 20 à 30 % de requêtes supplémentaires avec un ECC 170 qu'avec RSA 1024 et 200 % avec ECC 230 plutôt que RSA 1024. Et encore, l'implémentation de RSA dans SSL est certainement beaucoup mieux optimisé que l'adaptation à ECC qui en a été faite pour ce test.

Plus précisément on ne peut pas vraiment dire que ECC soit toujours plus efficace que RSA. En effet, dans la pratique courante, on observe que ECC est beaucoup plus rapide que RSA pour déchiffrer ou signer un message (facteur 4 à 20) c'est à dire pour les opérations privées alors que pour chiffrer ou vérifier une signature (opérations publiques), RSA est beaucoup plus rapide. En fait ECC met à peu près le même temps pour tout alors que RSA est beaucoup plus déséquilibré.

Le choix peut donc dépendre de l'utilisation qui doit être faite (puissance de calcul des acteurs, exigence en terme d'efficacité pour les différentes opérations). Ainsi, si le chiffrement (opération publique) est effectué par une carte à puce et le déchiffrement par un gros serveur, RSA sera très bien adapté. Il en sera de même si le temps de déchiffrement n'a pas d'importance mais que le chiffrement doit être le plus rapide possible (en fait dès qu'il y a un déséquilibre entre l'opération publique et l'opération privée).

Cela est du au fait qu'avec RSA les gens choisissent en général  $e = 3$  comme exposant public (bien que ce soit reconnu que c'est une faille majeure de RSA) et il n'y a donc essentiellement rien à faire pour les opérations publiques. Par contre les opérations privées sont beaucoup plus longues car elles utilisent cette fois comme exposant l'inverse de  $e$  modulo  $(p-1)(q-1)$  qui lui pèse bien ses 1024 bits. En choisissant un autre exposant public que 3 (65535 par exemple) RSA est encore plus rapide que ECC pour les opérations publiques (et toujours aussi lent pour les opérations privées bien sur) mais dans des proportions bien

plus raisonnables (facteur 2 ou 3) mais je ne mettrais pas ma main à couper que c'est beaucoup plus fiable qu'avec 3 car ça reste très petit. Enfin il y a fort à parier qu'avec une implémentation avec un exposant public aléatoire les opérations publiques deviendraient aussi catastrophique que les opérations privées sans pour autant que celles ci ne deviennent plus rapides.

Pour finir, même si c'est une opération à priori rare, il est intéressant de noter que la génération de clé (trouver deux nombres premier pour RSA, trouver un point sur la courbe pour ECC) est beaucoup plus rapide pour ECC.