Théorie Algorithmique des Nombres et Cryptographie

Ecole de recherche CIMPA-MAURITANIE

Cryptographie basée sur les courbes elliptiques

Sylvain Duquesne

15-26 Février 2016

Université Rennes 1

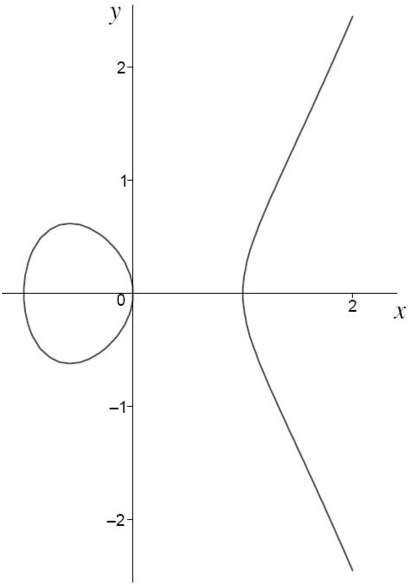
Courbes elliptiques sur $\mathbb{R}$

Définition

Une courbe elliptique est une courbe algébrique projective lisse de genre 1 possédant un point rationnel.
Riemann-Roch $\Rightarrow$ une courbe elliptique définie sur $\mathbb{R}$ peut être représentée par l'ensemble des points $(x, y) \in \mathbb{R}^2$ satisfaisant l'équation

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

où a et $b \in \mathbb{R}$ tq $4a^3 + 27b^2 \neq 0$ auquel on ajoute un point O appelé "point à l'infini".



Structure de groupe

Soit $P = (x, y)$ et Q des points de E. On définit l'opposé de P par $-P = (x, -y)$ et la somme de P et Q par les règles suivantes

- Soit L la droite passant par P et Q
- L recoupe E en un troisième point R
- $P + Q$ est l'opposé R
- si $P = Q$, L est la tangente à la courbe en P
- si $P = O$, alors $P + Q = Q$
- si $P = -Q$, alors $P + Q = O$

Grâce à ces règles d'addition, l'ensemble des points de E forme un groupe commutatif d'élément neutre O

Utilisation en cryptographie

On peut définir le logarithme discret sur E :
Soit P un point sur une courbe elliptique E et $Q = nP = P + P + \cdots + P$, alors n est le logarithme discret de Q en base P.

Protocole d'échange de clé de Diffie-Hellman sur les courbes elliptiques

A et B veulent partager un secret

- A choisit un entier a, calcule aP et l'envoie à B
- B choisit un entier b, calcule bP et l'envoie à A
- A et B calculent tous les 2 abP qui est le secret partagé
- Un attaquant peut connaître P, aP and bP mais ne peut pas retrouver abP sans calculer un log discret.

Remarque

La seule différence avec le log discret sur les corps finis est que la loi de groupe est définie additivement au lieu de multiplicativement

Courbes elliptiques sur les corps finis

Soit p un nombre premier plus grand que 5 et $q = p^r$. Une courbe elliptique définie sur $\mathbb{F}_q$ est donnée par une équation de la forme

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

avec $a, b \in \mathbb{F}_q$ et $4a^3 + 27b^2 \neq 0$.

L'ensemble des points de E forme un groupe de taille environ q. Plus précisément, d'après le théorème de Hasse

$$q + 1 - 2\sqrt{q} \leq \# E \leq q + 1 + 2\sqrt{q}$$

Soit P un point de E d'ordre ℓ ($\ell P = O$). On utilise le logarithme discret sur le sous-groupe de E engendré par P :

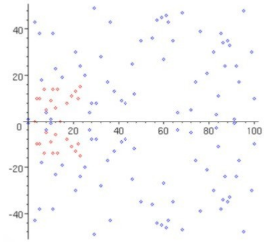
$$G = \{P, 2P, 3P, \cdots, \ell P\}$$

En pratique, on veut $\# E = m\ell$ avec m (très) petit

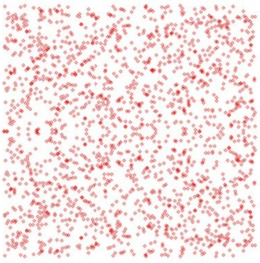
Exemple

Soit E la courbe elliptique définie par l'équation

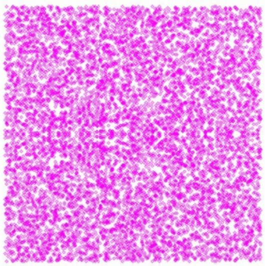
$$y^2 = x^3 + x + 1$$



sur $\mathbb{F}_{31}$ (en rouge)
et $\mathbb{F}_{101}$ (en bleu)



sur $\mathbb{F}_{2003}$



sur $\mathbb{F}_{10007}$

Attaques sur les courbes elliptiques

Attaques génériques

E est un groupe. Le attaques génériques sur le log discret (BSGS, Pollard-ρ) sont donc en $O(\sqrt{\ell})$ où ℓ est le plus grand diviseur premier de $\# E$.

Calcul d'indice

On ne sait pas trouver une "bonne" base de facteurs pour les courbes elliptiques et des heuristiques tendent à prouver qu'on ne peut pas en trouver.

Transfert du logarithme discret

Pour certaines courbes elliptiques, on peut transférer le problème du logarithme discret vers un problème de logarithme discret plus facile à résoudre.

Courbes anomales

Définition

Une courbe elliptique E définie sur un corps premier $\mathbb{F}_p$ est anomale si $\# E = p$

Il existe un isomorphisme facilement calculable

$$\psi : E \to (\mathbb{F}_p, +)$$

Il est alors suffisant de résoudre le logarithme discret dans $(\mathbb{F}_p, +)$ pour le résoudre dans E.

Le log discret dans $(\mathbb{F}_p, +)$

Soient a et $b \in \mathbb{F}_p$ tels que $b = na \bmod p$, retrouver n.

Euclide étendu $\to n = ba^{-1}$.
On utilise le fait que $(\mathbb{F}_p, +)$ a une structure supplémentaire (sa structure de corps)

L'attaque MOV (Menezes-Okamoto-Vanstone 1993)

Le couplage de Weil

Soit P un point de E d'ordre ℓ. Soit k le plus petit entier tel que $q^k = 1 \mod \ell$. Il existe un isomorphisme bilinéaire

$$e : E \times E \to \left(\mathbb{F}_{q^k}\right)^* .$$

Il est facilement calculable si k n'est pas trop grand.
Bilinéarité : $e(nP, Q) = e(P, Q)^n = e(P, nQ)$

Utilisation destructive des couplages

- ECDDH (étant donnés P, aP, bP et cP, décider si $cP = abP$) est facile puisque

  $$e(abP, P) = e(aP, bP)$$

  Il suffit de tester si $e(P, cP) = e(aP, bP)$
- e transfert le log discret sur $(E, +)$ en un log discret sur $\left(\left(\mathbb{F}_{q^k}\right)^*, \times\right)$. Si k est petit, le calcul d'indice permet de calculer un tel log discret.

Réalisation de l'attaque MOV

- Pour la plupart des courbes $k \approx \ell$ (et en pratique $\ell \approx q$) donc $\left(\mathbb{F}_{q^k}\right)^*$ est énorme et le calcul d'indice sur $\left(\mathbb{F}_{q^k}\right)^*$ est bien pire qu'une attaque par force brute sur E.
- Les courbes supersingulières (telles que $\# E = 1 \bmod p$) ont un petit k ($k \leq 6$)

Exemple

Soit p un nombre premier de 256 bits.
Une courbe elliptique définie sur $\mathbb{F}_p$ est censée fournir 128 bits de sécurité (attaques génériques) (equiv. à RSA 3072).

Si la courbe est supersingulière, $k = 2$

attaque MOV $\Rightarrow$ log discret sur un corps fini de 512 bits
$\Rightarrow$ 64 bits de sécurité (equiv. à RSA 512).

Finalement, les courbes supersingulières (et plus généralement avec k petit) doivent être évitées mais on les utilisera quand même

Attaque GHS (Gaudry-Hess-Smart 2002)

La restriction aux scalaires de Weil

$$z^2 = z \text{ sur } \mathbb{C}\ (\approx \mathbb{R}^2) \quad \overset{z=x+iy}{\Longleftrightarrow} \quad \left\{ \begin{array}{rcl} x^2 - y^2 & = & x \\ 2xy & = & y \end{array} \right. \text{ sur } \mathbb{R}$$

Une équation définie sur $\mathbb{F}_{q^g}$ $\Longleftrightarrow$ g équations définies sur $\mathbb{F}_q$

Variété de dimension 1 sur $\mathbb{F}_{q^g}$ $\Longleftrightarrow$ Variété de dimension g sur $\mathbb{F}_q$

Attaque GHS

Cas particulier de la restriction aux scalaires de Weil. Sous certaines conditions

Courbe elliptique définie $\mathbb{F}_{q^g}$ $\Longleftrightarrow$ Courbe de genre g définie sur $\mathbb{F}_q$

$\rightarrow$ Transfert du log discret sur une courbe elliptique vers le log discret sur une courbe hyperelliptique de genre g.

Si $g \geq 4$, le calcul d'indice permet de le calculer avec une meilleure complexité que les attaques génériques.

Réalisations de l'attaque GHS et ses variantes

Exemple

$\mathbb{F}_{2^{155}}$ 3 sous-corps : $\mathbb{F}_2, \mathbb{F}_{2^5}$ et $\mathbb{F}_{2^{31}}$

DL sur $E(\mathbb{F}_{2^{155}}) \Longleftrightarrow$ DL sur une courbe hyp. de genre 31 définie sur $\mathbb{F}_{2^5}$.

Ne marche que pour $\approx 2^{32}$ courbes elliptiques définies sur $\mathbb{F}_{2^{155}}$ (et 2^{104} pour les variantes) mais marche pour une courbe proposée comme standard.

- Si p est premier et $p \in [160, 600]$, l'attaque GHS est impraticable sur $\mathbb{F}_{2^p}$
- GHS est efficace sur $\mathbb{F}_{q^g}$ si g est un Mersenne (31,127)
- Attaques sur $\mathbb{F}_{q^7}, \mathbb{F}_{q^{17}}, \mathbb{F}_{q^{23}}$ et $\mathbb{F}_{q^{31}}$
- GHS impraticable $\not\Longrightarrow$ restriction aux scalaires de Weil impraticable.

Pourquoi utiliser les courbes elliptiques en cryptographie ?

- Pas de meilleures attaques connues que les attaques génériques (excepté pour quelques courbes).
- Plus petit corps de base que RSA ou le DL sur les corps finis (eg 160 bits au lieu de 1024 pour 80 bits de sécurité)
    - Arithmétique du corps de base plus facile à implémenter et plus efficace
    - Clés plus petites qu'avec RSA
    - Génération de clé facile (contrairement à RSA)
    - ECC plus rapide que le DL sur les corps finis
    - ECC plus rapide que RSA pour les opérations privées mais pas pour les opérations publiques (si on choisit $e = 3$ pour RSA)
- Ce fossé entre ECC et les autres systèmes (attaques exponentielles contre sous-exponentielles) s'accroit.
- De nouveaux protocoles deviennent possibles (utilisation des couplages)

Génération des paramètres

Pour construire une courbe elliptique ayant n bits de sécurité

- Choisir un nombre de $2n+\varepsilon$ bits q de la forme p ou 2^p avec p premier (GHS)
- Choisir une courbe elliptique E définie sur $\mathbb{F}_q$ tel que $\# E = m\ell$ (rappel : $\# E \approx q$) avec ℓ premier de $2n$ bits (attaques génériques)
- Si $q = p$, éviter les courbes anomales (vérifier que $\# E \neq p$)
- Eviter les courbes supersingulières (vérifier que $\# E \neq 1 \mod p$ (si $q = p$) ou mod 2 (si $q = 2^p$)). Plus généralement vérifier que $q^k \neq 1 \mod \ell$ pour $k = 1 \cdots 20$ (attaque MOV)
- Choisir un point P au hasard sur E et vérifier que son ordre est ℓ

Remarques

- La courbe et le point peuvent être choisis de façon à optimiser l'arithmétique.
- On peut aussi utiliser les standards

Niveaux de sécurité

Niveau de sécurité valable jusqu'en	80 2010	112 2030	128 >2030	192	256
Clé secrète	Skipjack	triple-DES	AES-128	AES-192	AES-256
Hachage	SHA-1	SHA-224	SHA3-256	SHA3-384	SHA3-512
RSA	1024	2048	3072	8192	15360
$(\mathbb{F}_q)^*$ corps $(\mathbb{F}_q)^*$ clés	1024 160	2048 224	3072 256	7680 384	15360 512
ECC	160	224	256	384	512
RSA/ECC	6.4	9.1	12	21.3	30

Remarque

Les tailles RSA données sont des estimations simplifiées (mais proviennent du gouv. US). Il existe des estimations pires pour RSA qui semblent plus réalistes.

Comptage de points

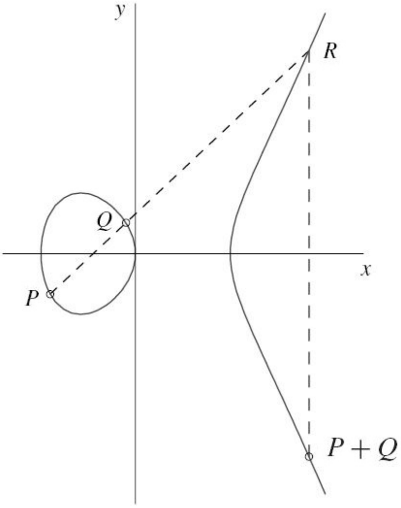
E définie sur $\mathbb{F}_q$ avec $q = p^r$. Déterminer $\# E$ est un problème difficile car $\# E = \log_P(O)$ mais nécessaire (sécurité).
Les meilleurs algorithmes font intervenir des mathématiques de haut niveau.

Algorithme	Complexité	temps de comptage en 160 bits
borne de Hasse + ρ-Pollard	$O\left(\sqrt[4]{q}\right)$	1 an
SEA	$O\left(\log^6 q\right)$	1 s
AGM+SST	$O\left(r^{2.5}\right)$	60 ms

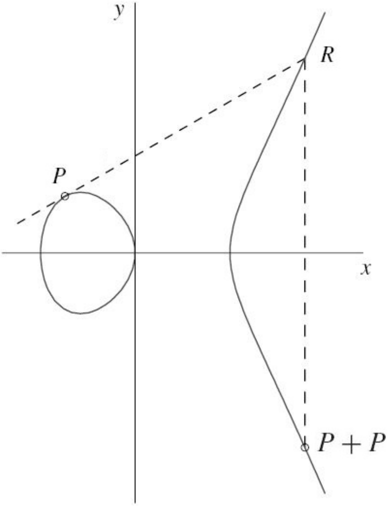
Comment trouver une bonne courbe pour la cryptographie

- Tirer une courbe au hasard
- Compter ses points
- Vérifier toutes les conditions de sécurité (GHS, MOV, anomales, attaques génériques)
- Recommencer si ces conditions ne sont pas vérifiées ($\log(q)$ tests en moyenne)

Remember the geometric group law



Addition



Doubling

Group law

Geometric description $\longrightarrow$ explicit formulas (over $\mathbb{R}$)

The equation of the line passing through $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ is

$$y = \lambda x + y_1 - \lambda x_1$$

with

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1} \qquad \text{if } P \neq Q$$

$$\lambda = \frac{3x_1^2 + a}{2y_1} \qquad \text{if } P = Q$$

These formulas can be extended to finite fields (and we can prove that it is a group law)

(affine) Formulas for group law over $\mathbb{F}_p$, p prime

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

If $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ are two different points in E

- The opposite of P is $-P = (x_1, -y_1)$
- The sum of P and Q is the point (x_3, y_3) with

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1}, \quad x_3 = \lambda^2 - x_1 - x_2 \text{ and } y_3 = \lambda(x_1 - x_3) - y_1$$

- The double of P is the point (x_3, y_3) with

$$\lambda = \frac{3x_1^2 + a}{2y_1}, \quad x_3 = \lambda^2 - 2x_1 \text{ and } y_3 = \lambda(x_1 - x_3) - y_1$$

Cost of the group law

Performing an addition requires one inversion (I), 2 multiplications (M) and one squaring (S) on $\mathbb{F}_p$.
Performing a doubling requires I+2M+2S

Scalar multiplication techniques

Computing nP on an elliptic curve is the central operation for cryptography. Just transpose exponentiation techniques from multiplicative groups (RSA, DL) to additive groups

Example : Double and add

Square and multiply

Input : m, d Output : m^d

$t \leftarrow 1$

for each bit d_i of d from left to right do

  $t \leftarrow t^2$

  if $d_i = 1$ then $t \leftarrow t \times m$

return t

Double and add

Input : P, n Output : nP

$T \leftarrow O$

for each bit n_i of n from left to right do

  $T \leftarrow 2T$

  if $n_i = 1$ then $T \leftarrow T + P$

return T

Scalar multiplication techniques

Cost of the scalar multiplication

Double and add	$\log_2(n)$ doubling and $\frac{\log_2(n)}{2}$ additions in average
sliding windows, w-NAF with w as window size	$\log_2(n)$ doubling and $\frac{\log_2(n)}{w+1}$ additions in average

Doubling must be optimized at the expense of addition

Remarks

- The addition involved in double and add (or better methods) is always an addition with P (or $3P$, ...)
- $-P$ is trivial to compute $\rightarrow$ NAF well adapted

Multi-exponentiation (Shamir's trick)

- Compute $nP + mQ$ (or more) as fast as nP (if $n \geq m$)
- Not specific to ECC

Algorithm

Input : $P, Q, n = (n_{t-1}, \cdots, n_0)_2, m = (m_{t-1}, \cdots, m_0)_2$ with $n_{t-1} \neq 0$.
Output : $nP + mQ$.

- Precompute $PQ = P + Q$
- $T \leftarrow O$
- for each bit n_i of n do

  $T \leftarrow 2T$

  $T \leftarrow T + n_i P + m_i Q$
- Return T

Applications

- Digital signature protocols (ECDSA)
- Gallant-Lambert-Vanstone (GLV) point multiplication

The GLV point multiplication

Requires the existence of an endomorphism ϕ on E (a rational map from E to E which is a group homomorphism)

Hypotheses

- P point of order ℓ on $E(\mathbb{F}_p)$
- The characteristic polynomial of ϕ has a root λ mod ℓ

The map ϕ acts on $< P >$ as the multiplication by λ : $\phi(P) = \lambda P$

Algorithm

Input : $P \in E(\mathbb{F}_p), k < \ell$
Output : kP

- Write $k = k_1 + k_2\lambda$ mod ℓ where $0 \leq k_1, k_2 \leq \sqrt{\ell}$
- Compute $kP = k_1 P + k_2\phi(P)$ using multi-exponentiation techniques

Around 50% speed-up

Projective coordinates over $\mathbb{F}_p$

To avoid inversion, we introduce denominators and compute them separately. So, put $x = \frac{X}{Z}$ and $y = \frac{Y}{Z}$ $\rightarrow$ projective model of the curve.

- A point on E is represented by the triple (X, Y, Z)
- $(X, Y, Z) = (\mu X, \mu Y, \mu Z)$ $\rightarrow$ representation not unique
- The point at infinity becomes $(0, 1, 0)$
- Defining equation over $\mathbb{F}_p$ becomes

$$Y^2 Z = X^3 + aXZ^2 + bZ^3$$

- The opposite of (X, Y, Z) is $(X, -Y, Z)$
- Doubling and addition do not involve inversions
- An inversion is required at the end of the scalar multiplication if we want nP in affine coordinates

Mixed addition

If P is given in affine coordinates (Z=1), additions with P can be speed up

Formulas for projective coordinates in $\mathbb{F}_p$

We just replace x_i by $\frac{X_i}{Z_i}$ and y_i by $\frac{Y_i}{Z_i}$ in affine formulas

Doubling

$$\begin{aligned} X_3 &= 2Y_1Z_1\left(\left(aZ_1^2+3X_1^2\right)^2-8X_1Y_1^2Z_1\right) \\ Y_3 &= \left(aZ_1^2+3X_1^2\right)\left(4X_1Y_1^2Z_1-\left(\left(aZ_1^2+3X_1^2\right)^2-8X_1Y_1^2Z_1\right)\right)-8Y_1^4Z_1^2 \\ Z_3 &= 8Y_1^3Z_1^3 \end{aligned}$$

Doubling requires 7M+5S (6M+5S if we choose a small).

Addition

$$\begin{aligned} C &= \left((Y_2Z_1-Y_1Z_2)^2Z_1Z_2-(X_2Z_1-X_1Z_2)^3-2(X_2Z_1-X_1Z_2)^2X_1Z_2\right) \\ X_3 &= (X_2Z_1-X_1Z_2)C \\ Y_3 &= (Y_2Z_1-Y_1Z_2)\left((X_2Z_1-X_1Z_2)^2X_1Z_2-C\right)-(X_2Z_1-X_1Z_2)^3Y_1Z_2 \\ Z_3 &= (X_2Z_1-X_1Z_2)^3Z_1Z_2 \end{aligned}$$

Addition requires 12M+2S and mixed addition ($Z_2 = 1$) only 9M+2S

Jacobian coordinates

Projective coordinates are not the most logical.
Let (X, Y, Z) such that $x = \frac{X}{Z^2}, y = \frac{Y}{Z^3}$
The equation of the curve becomes

$$Y^2 = X^3 + aXZ^4 + bZ^6$$

and the point at infinity is $(1, 1, 0)$

Variants

- modified Jacobian : (X, Y, Z, aZ^4) allowing to save an operation during the doubling if a is random.
- Jacobian Chudnovsky : (X, Y, Z, Z^2, Z^3) allowing to save an operation during the addition.

In practice, mixed use of various types of coordinates (precomputations, Double+Add, Double+Double)

Formulas for Jacobian coordinates in $\mathbb{F}_p$

Just replace x_i by $\frac{X_i}{Z_i^2}$ and y_i by $\frac{Y_i}{Z_i^3}$ in affine formulas

Doubling

$$A = 4X_1Y_1^2, \quad B = 3X_1^2 + aZ_1^4$$
$$X_3 = -2A + B^2, \quad Y_3 = -8Y_1^4 + B(A - X_3), \quad Z_3 = 2Y_1Z_1$$

The doubling step requires 4M+6S (4M+4S if $a = -3$ is chosen).
4M+4S in modified Jacobian, 5M+6S in Chudnovsky

Addition

$$A = X_1Z_2^2, \; B = X_2Z_1^2, \; C = Y_1Z_2^3, \; D = Y_2Z_1^3, \; E = B - A, \; F = D - C$$
$$X_3 = -E^3 - 2AE^2 + F^2, \quad Y_3 = -CE^3 + F(AE^2 - X_3), \quad Z_3 = Z_1Z_2E$$

The addition step requires 12M+4S (13M+6S in modified, 11M+3S in Chudnovsky) and the "mixed addition" step ($Z_2 = 1$) only 8M+3S (9M+5S in modified, 8M+3S in Chudnovsky)

Isomorphism classes

Isomorphic elliptic curves

- Definition : 2 elliptic curves E_1 and E_2 are isomorphic if the change of variables
$$(x, y) \to (u^2 x + r, u^3 y + u^2 sx + t)$$
transforms the equation of E_1 into the one of E_2.
- Consequence : The groups $E_1(\mathbb{F}_p)$ and $E_2(\mathbb{F}_p)$ are isomorphic.
- Property : the curves defined by the equations $y^2 = x^3 + ax + b$ and $y^2 = x^3 + a'x + b'$ are isomorphic if and only if there exist u such that $u^4 a' = a$ and $u^6 b' = b$. The change of variables is then
$$(x, y) \to (u^2 x, u^3 y)$$

The number of isomorphism classes of elliptic curves defined over $\mathbb{F}_p$ is $2p + 6, 2p + 2, 2p + 4$ or $2p$ depending if p equals $1, 5, 7$ or 11 modulo 12.

Consequence : All the elliptic curves cannot be written with $a = -3$

Isogeny classes

isogeneous elliptic curves

- Definition : an isogeny between 2 elliptic curves E_1 and E_2 is a non-constant rational map from E_1 to E_2 maps the neutral of E_1 to the neutral of E_2.
- Properties :
  - The groups $E_1(\mathbb{F}_p)$ and $E_2(\mathbb{F}_p)$ are homomorphic (an isogeny is a group morphism).
  - 2 elliptic curves are isogeneous if and only if they have the same cardinality

Then, the number of isogeny classes of elliptic curves defined over $\mathbb{F}_p$ is $4\sqrt{p}$.

Theorem : for most of the elliptic curves defined over $\mathbb{F}_p$, one can find an isogeneous curve such that $a = -3$.

Consequence : We can choose $a = -3$ with few loss of generality $\Rightarrow$ standards.

Optimizing Formulas

What do we want ?

- Reduce the number of operations
- Reduce the cost of each operation

How to do it ?

- Use alternative representation of the curve to obtain different formulas.
- Use small coefficients (eg $a = -3$) with few loss of generality.
- Introduce redundant representation of points (eg Jac. Chudnovsky) but must be balanced with bandwidth constraints.
- Replace multiplications by squaring (eg Z_3 in Jac. formulas)

All formulas are listed (with sage codes) on

`http://hyperelliptic.org/EFD`

A good recent reference is "Faster group operations on Elliptic Curves" by Hisil, Wong, Carter, Dawson.

Changing the curve representation

Montgomery form

$$E_m \ : \ By^2 = x^3 + ax^2 + x$$

E_m has a 2-torsion point $\Rightarrow \# E_m$ even

Hessian form

$$E_h \ : \ X^3 + Y^3 + Z^3 = cXYZ$$

E_h has a 3-torsion point $\Rightarrow \# E_h$ multiple of 3

Jacobi form

$$E_j \ : \ y^2 = x^4 + ax^2 + b$$

E_j has a 2-torsion point $\Rightarrow \# E_j$ even

Edwards form

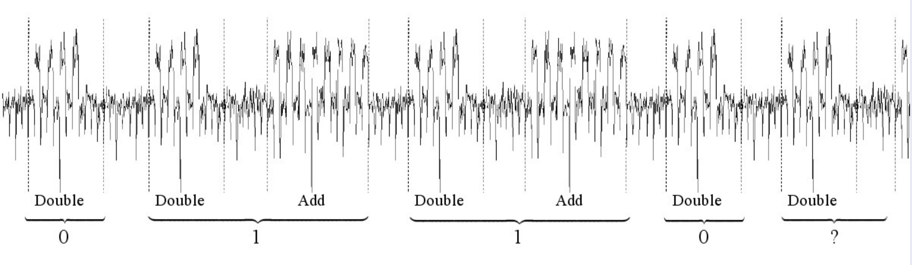
$$E_e \ : \ u^2 + v^2 = c^2(1 + du^2v^2)$$

E_e has a 4-torsion point $(c, 0) \Rightarrow \# E_e$ multiple of 4

Simple side channel attacks

Standard algorithms are sensitive to side channel attacks

SCA on double and add



Based on the fact that addition and doubling have not the same cost

SCA realized by analysis of timing, power consumption, electro-magnetic radiations, ...

Dummy operations

Examples of dummy operations on the curve

- Double and always add
- Recoding the exponent such that the sequence of operations is constant (including some dummy operations) eg DBL, DBL, ADD

Examples of dummy operations on the field

- Include dummy operations so that the cost of a doubling is the same that the cost of an addition
- Use atomic blocks (eg M, A, Neg, A on the base field) and construct doubling and addition using only such blocks

Drawbacks

- Loss of efficiency
- Vulnerable to fault attacks

Unified Formulas

Use curve representation such that doubling and addition use same formulas

Curves in Jacobi form

If $\# E = 0 \bmod 2$, E can be represented by $Y^2 = \varepsilon X^4 - 2\delta X^2 Z^2 + Z^4$
Even if $P = Q$, the sum of P and Q is given by

$$\begin{aligned}
X_3 &= X_1 Y_2 Z_1 + X_2 Y_1 Z_2 \\
Y_3 &= (Z_1^2 Z_2^2 + \varepsilon X_1^2 X_2^2)(Y_1 Y_2 - 2\delta X_1 X_2 Z_1 Z_2) + 2\varepsilon X_1 X_2 Z_1 Z_2 (X_1^2 Z_2^2 - X_2^2 Z_1^2) \\
Z_3 &= (Z_1^2 Z_2^2 - \varepsilon X_1^2 X_2^2)
\end{aligned}$$

These formulas require 12M+2S (or 8M+4S in most cases)

Formulas also exist for

- $\# E = 0 \bmod 3$: Hessian curves (6M+6S)
- unconditionally (13M+5S)
- Edwards form (9M+2S)
- Huff form (11M)

Montgomery ladder

Idea : Avoid the computation of y to improve efficiency
Drawback 1 : addition $P + Q$ is only possible if $P - Q$ is known
To compute nP, we use pairs (T_1, T_2) of consecutive multiples of P

Algorithm

Input : $P \in E$, n integer
Output : the x coordinate of nP
$(T_1, T_2) \leftarrow (O, P)$
For each bit n_i of n do
    if $n_i = 0$ then $T_1 \leftarrow 2T_1$ and $T_2 \leftarrow T_1 + T_2$
    if $n_i = 1$ then $T_1 \leftarrow T_1 + T_2$ and $T_2 \leftarrow 2T_2$
return T_1

At each step, $T_2 - T_1 = P$ so $T_2 + T_1$ can be computed

Drawback 2 : the y-coordinate of nP is not known (but can be recovered)

Both an addition and a doubling are performed for each bit

Montgomery Form

An elliptic curve in Montgomery form is given by an equation

$$By^2 = x^3 + Ax^2 + x$$

Transformation into Montgomery form (cf TD)

- A curve in Montgomery form is always transformable into short Weierstrass form
- A curve in short Weierstrass form (ie defined by $y^2 = x^3 + ax + b$) is transformable into Montgomery form if
  - the polynomial $x^3 + ax + b$ has at least one root α in $\mathbb{F}_p$
  - $3\alpha^2 + a$ is a square in $\mathbb{F}_p$

Remark : a curve in Montgomery form has a subgroup of order 4 so that its cardinality is a multiple of 4

Formulas for the Montgomery ladder over $\mathbb{F}_p$

For curves in Montgomery form

Addition : $P + Q$ if $P - Q$ is known and equal to $[x, y]$

$$\begin{aligned} X_3 &= ((X_2 - Z_2)(X_1 + Z_1) + (X_2 + Z_2)(X_1 - Z_1))^2 \\ Z_3 &= x\,((X_2 - Z_2)(X_1 + Z_1) - (X_2 + Z_2)(X_1 - Z_1))^2 \end{aligned}$$

Doubling

$$\begin{aligned} 4X_1Z_1 &= (X_1 + Z_1)^2 - (X_1 - Z_1)^2 \\ X_3 &= (X_1 + Z_1)^2(X_1 - Z_1)^2 \\ Z_3 &= 4X_1Z_1\left((X_1 - Z_1)^2 + \tfrac{A+2}{4}4X_1Z_1\right) \end{aligned}$$

Remarks

- Both an addition and a doubling need 3M+2S
- Best scalar multiplication (6M+4S per bit) and SCA resistant
- Formulas available for curves not in Montgomery form but need

Edwards form

Theorem

Let $c, d \in \mathbb{F}_p$, with d not a square, then the curve given by

$$C : u^2 + v^2 = c^2(1 + du^2v^2)$$

is isomorphic to the elliptic curve given by

$$y^2 = (x - c^4d - 1)(x^2 - 4c^4d)$$

The point $(0, c)$ is the neutral element for the group law on C which is

$$(u_1, v_1) + (u_2, v_2) = \left(\frac{u_1v_2 + u_2v_1}{c(1 + du_1u_2v_1v_2)}, \frac{v_1v_2 - u_1u_2}{c(1 - du_1u_2v_1v_2)}\right)$$

The opposite of (u, v) is $(-u, v)$

Remarks

- An addition requires 10M+S (9M for mixed addition) and a doubling 3M+4S (assuming c and d small)
- Variants exist (inverted Edwards, twisted Edwards)

Comparisons of systems of coordinates

	Dbl	Dbl $a = -3$ or small coeff	Add	Mixed add
Affine	I+2M+2S	I+2M+2S	I+2M+S	-
Projective	7M+5S	6M+5S	12M+2S	9M+2S
Jacobian	4M+6S	4M+4S	12M+4S	8M+3S
Mod. Jacobian	4M+4S	4M+4S	13M+6S	9M+5S
Montgomery	"3M+2S"	"2M+2S"	"3M+2S"	-
Edwards ($c = 1$)	3M+4S	3M+4S	10M+S	9M
twist Edwards by -1	4M+4S	4M+4S	8M	7M
Jacobi	14M	12M	14M	-
Hessian	12M	12M	12M	-

Formulas

- Chap. 13.2 of "Handbook of Elliptic and Hyperelliptic Curve Crypto."
- Chapter 2.6 of "Elliptic Curves : Number Theory and Cryptography"
- `http://hyperelliptic.org/EFD` (with sage codes)

Differential side channel attacks

Hypotheses

- Can ask the computation of kP for any chosen P where k is the private key (Access to a decipher oracle)
- Can analyse some leaks (as power consumption) during the computation of kP
- Want to recover k

Principle (assuming a double and add is used)

- Ask the computation of many $kP_i \rightarrow$ timings (or consumptions) T_i (depending on the values of P_i)
- Compute the quantities $2P_i + P_i \rightarrow$ timings (or consumptions) t_i

If the two sets $\{T_i\}$ and $\{t_i\}$ are correlated, the first bit of k is 1

Countermeasures based on randomization of the datas $\rightarrow$ dependence between T_i and the values of P_i lost

Countermeasures against differential side channel attacks (mainly due to Coron)

Scalar randomization

- $kP = (k + r\ell)P$
- $kP = (k + r)P - rP$
- Use redundant representation of scalar

Point randomization

- $kP = k(P + R) - kR$
- Take advantage of the redundant representation of points
  eg projective coordinates : $(X_P, Y_P, Z_P) = (rX_P, rY_P, rZ_P)$

isomorphism randomization

- Use isomorphic curve will change the coefficients and the point representation (remember an isomorphism is defined by some u)
- Use isomorphic field representation

Point Compression

Problem : In protocols (DH key exchange) with n bits of security ($2n$ bits keys), objects are points $[x, y]$ requiring $4n$ bits.
Remark : At most 2 points with same x-coordinate ($[x, y]$ and $[x, -y]$)
$\rightarrow$ store only x and one extra bit should be sufficient.

Compression

Keep only x and the parity bit of y.
Indeed, if y is even, $-y = p - y$ is odd.

Decompression

- Compute $x^3 + ax + b$
- Compute its square roots in $\mathbb{F}_p$ (y and $-y$)
- Choose the good root thanks to the parity bit.

Can also use Montgomery arithmetic where only the x-coordinate is used

Elliptic curves in standards

- Almost the same curves in every standards, eg P192
- Use of $a = -3$ for optimizing Jacobian coordinates
- Not compatible with fastest/secure methods (Montgomery ladder, unified coordinates, Edward curves)
- Use Mersenne or pseudo Mersenne primes for fast reduction

192 bits standard curve

$$
\begin{aligned}
p &= 2^{192} - 2^{64} - 1 \\
a &= -3 \\
b &= 2455155546008943817740293915197451784769108058161191238065 \\
n &= 6277101735386680763835789423176059013767194773182842284081 \\
Gx &= 602046282375688656758213480587526111916698976636884684818 \\
Gy &= 174050332293622031404857552280219410364023488927386650641
\end{aligned}
$$

Reminding pairings

Definition

In cryptography, a pairing is a map

$$e : (G_1, +)x(G_2, +) \rightarrow (G_3, x)$$

- bilinear, ie $e(g_1 + g_1', g_2) = e(g_1, g_2)e(g_1', g_2)$
- non degenerate, ie $\forall g_1 \in G_1, \exists g_2 \in G_2$ tq $e(g_1, g_2) \neq 1$
- easy to compute

Applications

- decisional Diffie-Hellman is easy.
- Transfert of discret log.
- tri-partite key-exchange.
- identity based cryptography.
- Short signatures
- Broadcast encryption

Function fields of curves

Definition

Let C be a plan affine curve defined over a field K by an equation

$$c(x,y)=0$$

A function f on C is a rational function with

- coefficients in $\overline{K}$
- variables x and y such that $c(x,y)=0$

We are interested in the functions evaluated on points of C with values in $\overline{K}\cup\{\infty\}$.

$$f\in\overline{K}(C)=\overline{K}(x,y)/c(x,y)$$

Zeroes and poles of functions

- A function f is said to have a **zero** at a point P of C if $f(x_P,y_P)=0$
- A function f is said to have a **pole** at a point P of C if $f(x_P,y_P)=\infty$

Order of zeroes and poles

How many times a point is vanishing a function ?

Uniformizer

For any point P on C, there exists a function u_P with $u_P(P) = 0$ such that every function f can be written in the form

$$f = u_P^r g, \text{ with } r \in \mathbb{Z} \text{ and } g(P) \neq 0, \infty$$

If $r > 0$, f is said to have a zero of order r at P
If $r < 0$, f is said to have a pole of order $|r|$ at P

Case of elliptic curves

- For a point $P = (x_P, y_P)$ with $y_P \neq 0$, one can take $\mathbf{u_P} = \mathbf{x} - \mathbf{x_P}$
- For a point $P = (x_P, 0)$, one can take $\mathbf{u_P} = \mathbf{y}$
- For the point at infinity, one can take $\mathbf{u}_\infty = \frac{\mathbf{x}}{\mathbf{y}}$

Our first divisors

Theorem

Let C be a curve and f a function on C that is not 0.

- f has finitely many zeroes and poles.
- Counting multiplicities (orders), f has as many poles as zeroes.
- If f has no zero or pole, then f is constant.

Divisors is just a way to give zeroes and poles of a function

Divisors of functions

Let f be a function on C having zeroes (and poles) P_i with order n_i. The divisor of f is the **formal** sum

$$\mathrm{div}(f) = \sum n_i P_i$$

These divisors of functions are called principal divisors

Divisors

Definition

For each point P on a curve C, we define the formal symbol $[P]$. A divisor D on C is a finite linear combination of such symbols with integer coefficients :

$$D = \sum a_i[P_i]$$

- The degree of divisor D is $\sum a_i$.
- The support of D is the set $\{P_i \in C | a_i \neq 0\}$.
- A function f can be evaluated at D by

$$f(D) = \prod f(P_i)^{a_i}$$

Group properties

- The set of divisors on C is a group denoted $Div(C)$
- The set of divisors of degree 0 is a subgroup of $Div(C)$: $Div^0(C)$
- The set of divisors of functions is a subgroup of $Div^0(C)$: $Princ(C)$

The Picard group

We say that 2 divisors D_1 and D_2 are equivalent if $D_1 - D_2$ is principal. The quotient group $Pic(C) = Div^0(C)/Princ(C)$ is called the Picard group

Theorem

Let E be an elliptic curve, then the map

$$\begin{array}{rcl} E & \rightarrow & Pic(E) \\ P & \mapsto & [P] - [P_\infty] \end{array}$$

is a group isomorphism

The Picard group is a generalization of the group structure of elliptic curves.

Consequence

$P_1 + P_2 = P_3$ on the curve means that there exists a function f such that

$$[P_1] + [P_2] - [P_3] - [P_\infty] = div(f)$$

Fundamental example

Goal : compute the function f involved in the sum of P_1 and P_2

$$[P_1] + [P_2] - [P_3] - [P_\infty] = div(f)$$

Find a function having P_1 and P_2 as zeroes

This means find a function vanishing in P_1 and P_2.
Let $l(x, y)$ be the line function passing through P_1 and P_2

$$l(x, y) = y - \lambda x - y_1 + \lambda x_1$$

where λ is the slope.
If $P_1 = P_2$, we want a line passing two times by P_1, ie with multiplicity 2 : the tangent to the curve

Find the divisor of l

P_1 and P_2 are zeroes,
The only pole is P_∞ with order 3 (as y),
The third zero is the third intersection point R between the line and the curve

$$div(l) = [P_1] + [P_2] + [R] - 3[P_\infty]$$

Find a function having R and $-R$ as zeroes

Let $v(x, y)$ be the vertical line vanishing $R = (x_R, y_R)$

$$v(x, y) = x - x_R$$

If $P_3 = (x_R, -y_R)$ we have

$$div(v) = [R] + [P_3] - 2[P_\infty]$$

Finally, we have

$$div(l/v) = [P_1] + [P_2] - [P_3] - [P_\infty]$$

Computing the function of a principal divisor

Question : given a principal divisor $D = \sum a_i([P_i] - [P_\infty])$ (with $a_i > 0$ for simplicity), compute f such that $div(f) = D$

Principle

- Write D as $[Q_0] - [P_\infty] + [Q_1] - [P_\infty] + \cdots + [Q_k] - [P_\infty]$
- Initialize T to Q_0 and f to 1.
- For each i
  - Compute the function h involved in the sum of T and Q_i
  - Update T to $T + Q_i$
  - Update f to $f \times h$

At each step, we have

- $T = Q_0 + \cdots + Q_i$
- $div(f) = [Q_0] + \cdots + [Q_i] - [Q_0 + \cdots + Q_i] - i[P_\infty]$.

At the end, we have $div(f) = D$

Case of the scalar multiplication

In cryptography, the operation ℓP is central. If P has order ℓ, the divisor $\ell[P] - \ell[P_\infty]$ is principal

Miller's algorithm

Input : a point P on E of order ℓ.
Output : the function f such that $div(f) = \ell[P] - \ell[P_\infty]$

- Initialisation : $T \leftarrow P, f \leftarrow 1$
- For each bit of ℓ (from left to right), do
  - $f \leftarrow f^2 \times h_{T,T}$
  - $T \leftarrow 2T$
  - If the bit is 1, do
    - $f \leftarrow f \times h_{T,P}$
    - $T \leftarrow T + P$
- Return f.

where $h_{A,B}$ is the function involved in the addition of A and B

Torsion points on elliptic curves

Definition

The set of *m*-torsion points is

$$E[m] = \{P \in E(\overline{K}) | mP = P_\infty\}$$

Structure of the torsion points

Let E be an elliptic curve over K an m not divisible by $char(K)$, then

$$E[m] \simeq \mathbb{Z}/m\mathbb{Z} \oplus \mathbb{Z}/m\mathbb{Z}$$

The case of *p*-torsion for $char(K)|p$

There are only two possible cases

- $E[p] \simeq \mathbb{Z}/p\mathbb{Z}$ and E is said to be ordinary
- $E[p] \simeq 0$ and E is said to be supersingular

The Weil pairing

Let μ_n be the group of n-th roots of unity of $\overline{K}$

Definition

$$\begin{array}{rcl} e_w : E[n] \times E[n] & \to & \mu_n \\ (P, Q) & \mapsto & \frac{f_P(D_Q)}{f_Q(D_P)} \end{array}$$

where

- f_P is a function such that $div(f_P) = n[P] - n[P_\infty]$.
- D_Q is a divisor equivalent to $[Q] - [P_\infty]$ s.t. $supp(D_Q) \cap supp(div(f_P)) = \emptyset$.
- Idem for f_Q and D_P.

In practice one can choose $D_P = [P + R] - [R]$ and $D_Q = [Q + S] - [S]$ for some R, S in E so that

$$e_w(P, Q) = \frac{f_P(Q + S) f_Q(R)}{f_P(S) f_Q(P + R)}$$

Properties of the Weil pairing

Bilinearity

For all points $P, Q, P_1, P_2, Q_1, Q_2 \in E[n]$, we have

$$\begin{aligned} e_w(P_1 + P_2, Q) &= e_w(P_1, Q)e_w(P_2, Q) \\ e_w(P, Q_1 + Q_2) &= e_w(P, Q_1)e_w(P, Q_2) \end{aligned}$$

This implies that

$$e_w(kP, Q) = e_w(P, kQ) = e_w(P, Q)^k$$

Other properties

- Non degeneracy : for all $P \in E[n] - P_\infty, \exists Q \in E[n]$ s.t. $e_w(P, Q) \neq 1$
- $e_w(P, Q) = e_w(Q, P)^{-1}$
- $e_w(P_\infty, Q) = 1$
- $e_w(P, P) = 1$

Computing the Weil pairing

Miller's algorithm gives f_P and f_Q.
Just have to evaluate them at $Q+S, S, R, P+R$.

Refinements (for computing $f_P(Q+S)/f_P(S)$)

- Evaluate the intermediate functions at each step (but $Q+S$ and S should not be one of the intermediate values T).
- Evaluate $f_P(S)$ and $f_P(Q+S)$ at the same time but not $f_P(Q+S)/f_P(S)$.
- Replace each step $f \leftarrow f^2 \times h$ with $h = l/v$ by

$$\begin{array}{rcl} f_1 & \leftarrow & f_1^2 \times l(Q+S) \times v(S) \\ f_2 & \leftarrow & f_2^2 \times v(Q+S) \times l(S) \end{array}$$

- Be careful to the last step.
- Operations are in $\overline{K}$ and so can be very large. If $K = \mathbb{F}_q$, have a good arithmetic on extension fields $\mathbb{F}_{q^k}$.

The embedding degree

$K = \mathbb{F}_q$ and $n = \ell$ is prime

Definition

The embedding degree is the smallest integer k such that $\ell | q^k - 1$, i.e. $\mathbb{F}_{q^k}$ is the smallest extension of $\mathbb{F}_q$ containing μ_ℓ

Balasubramanian-Koblitz theorem

If $k > 1$, the ℓ-torsion is defined over $\mathbb{F}_{q^k}$

$$e_w : E(\mathbb{F}_{q^k})[\ell] \times E(\mathbb{F}_{q^k})[\ell] \rightarrow \mathbb{F}_{q^k}{}^* / \left(\mathbb{F}_{q^k}{}^*\right)^\ell$$

Size of k

- For an arbitrary curve, k is as large as $q \Rightarrow$ difficult computations.
- Supersingular curves have small k.
- Ordinary curves with small k are rare and difficult to find.

The Tate pairing

Let E be an elliptic curve defined over $\mathbb{F}_q$ and containing a subgroup of prime order ℓ. Let k be the embedding degree relatively to ℓ.

$$\begin{array}{rcl} e_T : E(\mathbb{F}_q)[\ell] \times E(\mathbb{F}_{q^k})/\ell E(\mathbb{F}_{q^k}) & \rightarrow & \mathbb{F}^*_{q^k}/(\mathbb{F}^*_{q^k})^\ell \\ P, Q & \mapsto & f_{\ell,P}(D_Q)^{\frac{q^k-1}{\ell}} \end{array}$$

with

- $f_{\ell,P}$ the function such that $Div(f_{\ell,P}) = \ell[P] - \ell[P_\infty]$.
- D_Q a divisor representing Q having disjoint support with $Div(f)$.
- The final exponentiation provides an unique representative.

Remarks

- Trivial if $Q \in E(\mathbb{F}_q)[\ell]$: not a symmetric pairing.
- Computed thanks to Miller algorithm and a final exponentiation.
- Requires only one function instead of 2 for the Weil pairing.

The magical tool for improvements

Lagrange theorem

Let G be a multiplicative group of cardinality $\#G$, then

$$\forall g \in G, g^{\#G} = 1$$

- if $G = \mathbb{F}_p^* \Rightarrow$ little Fermat Theorem
- if $G = (\mathbb{Z}/n\mathbb{Z})^* \Rightarrow$ Euler theorem (for RSA proof)
- if $G = (\mathbb{F}_{q^e})^* \Rightarrow g^{q^e-1} = 1$

Application to pairing computation

Let e strictly dividing k (the embedding degree), we have $q^e - 1 | \frac{q^k-1}{\ell}$ so that

$$\forall g \in \mathbb{F}_{q^e}, g^{\frac{q^k-1}{\ell}} = 1$$

Consequence : any subfield factor in $f_{\ell,P}(D_Q)$ can be discarded during the pairing computation

Forgetting the divisors

Remember : we cannot choose $D_Q = [Q] - [P_\infty]$ because $P_\infty \in supp(f_{\ell,P})$.
Let $D = \ell[P+R] - \ell[R]$ for some a random point R. D is principal and equivalent to $\ell[P] - \ell[P_\infty]$

$$div(f') = \ell[P+R] - \ell[R] \sim div(f_{\ell,P})$$

Then $e_T(P,Q) = f'(D_Q)^{\frac{q^k-1}{\ell}}$ and now D_Q can be chosen to be $[Q] - [P_\infty]$

$$\begin{aligned} e_T(P,Q) &= f'([Q]-[P_\infty])^{\frac{q^k-1}{\ell}} \\ &= \left(\frac{f'(Q)}{f'(P_\infty)}\right)^{\frac{q^k-1}{\ell}} \\ &= f'(Q)^{\frac{q^k-1}{\ell}} \end{aligned}$$

because $P_\infty \in E(\mathbb{F}_q) \Rightarrow f'(P_\infty) \in \mathbb{F}_q^* \Rightarrow$ discarded.
This quantity does not depend on R so that finally

$$e_T(P,Q) = f_{\ell,P}(Q)^{\frac{q^k-1}{\ell}} \text{ (but } f_{\ell,P}(Q) \neq f_{\ell,P}(D_Q))$$

Tate pairing computation if $k > 1$ and $P \in E(\mathbb{F}_q), Q \notin E(\mathbb{F}_q)$

Compute $e_T(P, Q)$ with $Q = [x_Q, y_Q]$

- $T \leftarrow P$, $f_1, f_2 \leftarrow 1$
- For each bit ℓ_i of ℓ from left to right do
  - $\lambda \leftarrow$ the slope of the tangent line to E in $T = [x_T, y_T]$
  - $f_1 \leftarrow f_1^2 \times (y_Q - \lambda(x_Q - x_T) - y_T)$
  - $T \leftarrow 2T \qquad (T = (x_{2T}, y_{2T}))$
  - $f_2 \leftarrow f_2^2 \times (x_Q - x_{2T})$
  - if $\ell_i = 1$ then
    - $\lambda \leftarrow$ the slope of the line passing trough $T = [x_T, y_T]$ and P
    - $f_1 \leftarrow f_1 \times (y_Q - \lambda(x_Q - x_T) - y_T)$
    - $T \leftarrow T + P \qquad (T = (x_{2T}, y_{2T}))$
    - $f_2 \leftarrow f_2 \times (x_Q - x_{2T})$
- return $(f_1/f_2)^{\frac{q^k-1}{\ell}}$

The MNT curves

Theorem

A prime order ordinary curve defined over $\mathbb{F}_p$ is verifying $k = 6$ iff there is a x such that

- $p = 4x^2 + 1$
- $t = 1 \pm 2x$ (so that $\#E = 4x^2 \mp 2x + 1$)

For cryptographic application, it is sufficient to find a (sparse) x such that p and $\#E = p + 1 - t$ are prime. Then construct a curve using the CM method.

Final exponentiation

$p^6 - 1 = (p^3 - 1)(p + 1)(p^2 - p + 1)$

The exponent involved in the hard part of the final exponentiation is

$$\frac{p^2 - p + 1}{p + 1 - t} = p \pm 2x$$

Twists of elliptic curves (in char ≥ 5)

Definition

Let E be an elliptic curve defined over $\mathbb{F}_q$.
An elliptic curve $\tilde{E}$ is a twist of degree d of E if there exists an isomorphism $\varphi_d : \tilde{E} \to E$ defined over $\mathbb{F}_{q^d}$ with d minimal.
In char. ≥ 5, the only available degrees for twists are $2, 3, 4$ and 6

The twisted pairing is defined by

$$\begin{array}{rcl} e_t : E(\mathbb{F}_p)[\ell] \times \tilde{E}(\mathbb{F}_{p^{k/d}})[\ell] & \to & \mathbb{F}_{p^k}{}^* / \left(\mathbb{F}_{p^k}{}^*\right)^\ell \\ P, Q & \mapsto & e_T(P, \varphi_d(Q)) \end{array}$$

Remarks

- This is equivalent to choose the second input of the Tate pairing in the image of φ_d
- Also available in small characteristic
- Many computations in $\mathbb{F}_{q^{k/d}}$ instead $\mathbb{F}_{q^k}$

The case of quadratic twists

Let E be an elliptic curve defined over $\mathbb{F}_p$ by $y^2 = x^3 + ax + b$ and $\ell | \#E$.
Assume the embedding degree k relatively to ℓ is even.
Let $\nu \in \mathbb{F}_{p^{k/2}}$ not a square (so $\mathbb{F}_{p^k} = \mathbb{F}_{p^{k/2}}[\sqrt{\nu}]$).
The quadratic twist $\tilde{E}$ is then defined by the equation

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

E are $\tilde{E}$ isomorphic over $\mathbb{F}_{p^k}$ via

$$\begin{array}{rcl} \varphi_2 : \tilde{E}\left(\mathbb{F}_{p^k}\right) & \rightarrow & E\left(\mathbb{F}_{p^k}\right) \\ (x, y) & = & (x, y\sqrt{\nu}) \end{array}$$

The second input of the twisted Tate pairing has the form $(x_Q, y_Q\sqrt{\nu})$ with x_Q and $y_Q \in \mathbb{F}_{p^{k/2}}$, so

- Evaluation in Q is twice faster
- $x_Q \in \mathbb{F}_{p^{k/2}} \Rightarrow f_2 \in \mathbb{F}_{p^{k/2}} \Rightarrow$ denominator elimination

The Weierstrass form of $\tilde{E}$ is

$$y^2 = x^3 + ax\nu^{-2} + b\nu^{-3}$$

Twisted Tate pairing computation if $k = 2e$

Input : $P \in E(\mathbb{F}_q)[\ell]$, $Q = (x_Q, y_Q\sqrt{\nu})$ with $x_Q, y_Q \in \mathbb{F}_{q^e}$
Output : $e_t(P, Q)$

$T = P, f = 1, \ell = (\ell_{n-1}..\ell_0)_2$
For i from $n-2$ to 0 do
  $\lambda \leftarrow$ the slope of the tangent line at T to E
  $f \leftarrow f^2\,(y_Q\sqrt{\nu} - \lambda(x_Q - x_T) - y_T)$
  $T \leftarrow 2T$
  if $\ell_i = 1$ then
    $\lambda \leftarrow$ the slope of the line passing through T and P
    $f \leftarrow f\,(y_Q\sqrt{\nu} - \lambda(x_Q - x_T) - y_T)$
    $T \leftarrow T + P$
$f \leftarrow f^{q^e - 1}$
$f \leftarrow f^{(q^e+1)/\Phi_k(q)}$
$f \leftarrow f^{\Phi_k(q)/\ell}$
Return f

The Barreto-Naehrig (BN) curves

Prime order curves ($\rho = 1$) given by an equation $y^2 = x^3 + b$ satisfying

- $p = 36u^4 + 36u^3 + 24u^2 + 6u + 1$
- $t = 6u^2 + 1$

Properties

- $k = 12$, optimal for 128 bits security level.
- Existence of a twist of order 6 given by $(x, y) \to (x\gamma^2, y\gamma^3)$ if γ^6 is neither a square nor a cube in $\mathbb{F}_{p^2}$.
- u can be chosen sparse (and $\ell = p + 1 - t$ is also sparse).
- Fast final exponentiation finale in $O(u^3)$ without multiexponentiation.

$$p^{12} - 1 = (p^6 - 1)(p^2 + 1)(p^4 - p^2 + 1)$$

$$f^{\frac{(p^4 - p^2 + 1)}{\ell}} = f^{p^3} \left[(f^p)^2 \, f^{p^2} \right]^{6u^2+1} b \, (f^p \, f)^9 \, a \, f^4 \text{ with } a = f^{-6u-5}, b = a \, a^p$$

Twisted Tate pairing computation on BN curves

Input : $P \in E(\mathbb{F}_p)[\ell]$, $Q = (x_Q\gamma^2, y_Q\gamma^3)$ with $x_Q, y_Q \in \mathbb{F}_{p^2}$
Output : $e_t(P, Q)$

$T = P, f = 1, \ell = (\ell_{n-1}..\ell_0)_2$
For i from $n-2$ to 0 do
  $\lambda \leftarrow$ the slope of the tangent line at T to E
  $f \leftarrow f^2\,(y_Q\gamma^3 - \lambda(x_Q\gamma^2 - x_T) - y_T)$
  $T \leftarrow 2T$
  if $s_i = 1$ then
    $\lambda \leftarrow$ the slope of the line passing through T and P
    $f \leftarrow f\,(y_Q\gamma^3 - \lambda(x_Q\gamma^2 - x_T) - y_T)$
    $T \leftarrow T + P$

$f \leftarrow f^{p^6-1}$

$f \leftarrow f^{p^2+1}$

$f \leftarrow f^{\frac{p^4-p^2+1}{\ell}}$

Return f

Example of BN curve

The most used BN curve ensures 126 bits of security

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

with $u = -(2^{62} + 2^{55} + 1)$ so that

- $p = 36u^4 + 36u^3 + 24u^2 + 6u + 1$ is prime
- $p + 1 - t = 36u^4 + 36u^3 + 18u^2 + 6u + 1$ is prime

$\mathbb{F}_{p^{12}}$ is defined by the following tower of extensions

- $\mathbb{F}_{p^2} = \mathbb{F}_p[i]/(i^2 + 1)$
- $\mathbb{F}_{p^6} = \mathbb{F}_{p^2}[\beta]/(\beta^3 - (1 + i))$
- $\mathbb{F}_{p^{12}} = \mathbb{F}_{p^6}[\gamma]/(\gamma^2 - \beta) = \mathbb{F}_{p^2}[\gamma]/(\gamma^6 - (1 + i))$

The second input of the twisted pairing has the form

$$(x_Q\gamma^2, y_Q\gamma^3) \text{ with } x_Q, y_Q \in \mathbb{F}_{p^2}$$

To find such a curve :

- pick a random sparse u
- check if p and $p + 1 - t$ are prime
- check if $\mathbb{F}_{p^{12}}$ can be nicely generated

Variants of the Tate pairing

The Ate pairing

$$\begin{array}{rcl} e_A : E(\mathbb{F}_p)[\ell]\times E(\mathbb{F}_{p^k})[\ell]\cap Ker(\pi_p - p) & \to & \mathbb{F}_{p^k}^*/(\mathbb{F}_{p^k}^*)^\ell \\ P, Q & \mapsto & f_{T,Q}(P)^{\frac{p^k-1}{\ell}} \end{array}$$

where π_p is the Frobenius map, $T+1$ its trace and $Div(f_{T,Q}) = TQ - (TQ) - (T-1)P_\infty$.

e_A is a power of e_T. Analog of η pairing but the roles of P and Q are exchanged.

The twisted-Ate pairing

If E has a twist of order d, changing $T = t-1$ by T^e with $e = k/pgcd(k,d)$, we can exchange back P and Q.
Only interesting if $T^e < \ell$, ie if T is small.

Optimal pairings

We can always find a pairing having a Miller loop of length $\log_2(p)/\varphi(k)$