The discrete logarithm problem and its application in Cryptography

Roger Oyono

University of French Polynesia, Tahiti

Lectures in Cryptography for Master class
Madrid, April 2009

Discrete logarithm problem (DLP) (1)

Two main problems on which public key cryptography is based:

- Integer factorisation (in RSA).
- DLP (ElGamal Cryptosystem, Diffie-Hellman key exchange):

  Let G be a cyclic finite group and $g \in G$ be a generator of G. The discrete logarithm problem (DLP) in G is the following:

  Given an element $h \in G$, find the smallest positive integer x such that

  $$h = [x]g \text{ (additive group)} \quad / \quad h = g^x \text{ (multiplicative group)} .$$

  We will denote such an x with $DL_g(h)$.

Discrete logarithm problem (DLP) (2)

As we will see later, a cryptographically suitable group G must satisfy the following conditions:

- representation is easy and compact.
- fast arithmetic.
- DLP is computationally hard.
- group order can be computed efficiently.

CDHP and DDHP

- The computational Diffie-Hellman Problem (CDHP) is the problem:

$$\text{Given } g, h_x = [x]g \text{ and } h_y = [y]g\,, \text{ compute } [xy]g\,.$$

- The resolution of the DLP implies the resolution of the CDHP.
- The decisional Diffie-Hellman Problem (DDHP) is the problem:

$$\text{Given } g, h_x = [x]g, h_y = [y]g \text{ and } h_z = [z]g\,, \text{ decide if } h_z = [xy]g.$$

- There are groups G for which DDHP is easier than CDHP or DLP, but we do not know how to answer this question in general.

Overview (1)

- Efficient scalar multiplication
- Solving the DLP in generic groups
  - Pohlig-Hellman
  - Shanks' Baby step - Giant step
  - Pollard rho
- Cryptographic protocols based on the DLP
  - Key exchange
  - Encryption
  - Signature
  - Security: what is a cryptographically secure group?

Overview (2)

- Subexponential algorithms for the DLP in finite (prime) fields
  - Generalities
  - Smooth numbers, factor base and subexponentiality
  - Adleman's algorithm
- Pairing in Cryptography
  - Generalities
  - Identity based Cryptography (IBE)
  - Tripartite key exchange.
- Elliptic curves
  - Generalities
    - Why interesting?
    - Group Law
  - DLP on "special elliptic curves"
  - Pairing with elliptic curves (Weil pairing)

Example: binary left to right (1)

The following algorithm is based on the binary expansion of n:

$$[(n_{\ell-1} \dots n_0)_2]P = [2]([(n_{\ell-1} \dots n_1)_2]P) \oplus [n_0]P$$

Example: $45 = (101101)_2$

P

$2P$

$2(2P) \oplus P$

$2(2(2P) \oplus P) \oplus P$

$2(2(2(2P) \oplus P) \oplus P)$

$2(2(2(2(2P) \oplus P) \oplus P)) \oplus P \quad = [45]P$

Scalar multiplication using binary left to right (2)

Algorithm (binary left to right (1))

IN: $P \in G$ and $n \in \mathbb{N}$

$n = (n_{\ell-1} \dots n_0), n_{\ell-1} = 1.$

OUT: $[n]P \in G$.

1. $R \leftarrow P$
2. for $i = \ell - 2$ down to 0 do
   1. $R \leftarrow [2]R$
   2. if $n_i = 1$ then $R \leftarrow R \oplus P$
3. return R

cost: $O(\log n)$ doublings /additions in the group G.

Generic groups (1)

A generic group is a group where we can only:

- Represent group elements (uniquely)
- Apply the group operation to a pair of elements to obtain a new element.

The representation of the group elements gives us no information on the structure of the group.

The group operation may be done using an oracle.

Most groups are not generic groups, but we can look at them as generic groups if we ”forget” the extra information...

Algorithms for solving the DLP for generic groups give us an upper bound on how hard things are!

Generic groups (2)

In generic groups, we will present three methods to compute $DL_g(h)$:

- Baby step - Giant step (Shanks)
- Pollard ρ
- Pollard kangaroo

and one more method that take advantage of the decomposition of the group order

- Pohlig-Hellman

Pohlig-Hellman

Idea: Non trivial subgroups can make the DLP easier!

Suppose the additive cyclic group $G = \langle g \rangle$ has order

$$N = p_1^{\alpha_1} \cdot p_2^{\alpha_2} \cdot \ldots \cdot p_k^{\alpha_k}$$

If we know $DL_g(h)$ modulo $p_i^{\alpha_i}$ for every i, then we can compute $DL_g(h)$ via the Chinese remainder theorem.

From the group order, we have:

$$G \simeq G_1 \times G_2 \times \cdots \times G_k$$

with

$$G_i \simeq \mathbb{Z}/p_i^{\alpha_i}\mathbb{Z}$$

We can restrict the DLP from G to G_i:

Define $g_i = \left[\frac{N}{p_i^{\alpha_i}}\right] g$ and $h_i = \left[\frac{N}{p_i^{\alpha_i}}\right] h$.

We can compute $DL_{g_i}(h_i)$ in a group of order $p_i^{\alpha_i}$ (instead of N).

We have

$$DL_{g_i}(h_i) \equiv \frac{DL_g(h_i)}{DL_g(g_i)} \equiv \frac{DL_g(\left[\frac{N}{p_i^{\alpha_i}}\right] h)}{DL_g(\left[\frac{N}{p_i^{\alpha_i}}\right] g)} \equiv \frac{\left[\frac{N}{p_i^{\alpha_i}}\right] DL_g(h)}{\left[\frac{N}{p_i^{\alpha_i}}\right] DL_g(g)} \equiv DL_g(h),$$

and g_i has order $p_i^{\alpha_i}$, so

$$DL_g(h) \equiv DL_{g_i}(h_i) \bmod p_i^{\alpha_i}.$$

Prime Powers

Assume now that $G = \langle g \rangle \simeq \mathbb{Z}/p^{\alpha}\mathbb{Z}$ and $h \in G$. For $DL_g(h) = x$, write

$$x = x_0 + x_1 p + x_2 p^2 + \ldots + x_{\alpha-1} p^{\alpha-1} \pmod{p^{\alpha}}$$

with $x_i \in [0, p-1]_{\mathbb{Z}}$.

Let $g' = [p^{\alpha-1}]g$, then g' has order p and the equality $[x]g = h$ becomes:

$$[x_0]g' = [x]g' = [p^{\alpha-1}]h$$

x_0 can be find by computing $DL_{g'}([p^{\alpha-1}]h)$ in $\langle g' \rangle$ (a subgroup of order p). We also compute x_1 via a DLP in $\langle g' \rangle$:

$$[x_1]g' = p^{\alpha-2}([-x_0]g + h)$$

We iterate this approach to compute $x_2, x_3, \ldots, x_{\alpha-1}$ and thus x.

Pohlig-Hellman: An example

Consider in $\mathbb{F}^*_{11251}$ the DLP

$$23^x = 9689\,.$$

Since $p-1 = 11250 = 2 \cdot 3^2 \cdot 5^4$, the Pohlig-Hellman algorithm should work well.
We thus have to solve

p_i	α_i	$g^{(p-1)/p_i^{\alpha_i}}$	$h^{(p-1)/p_i^{\alpha_i}}$	Solve $\left(g^{(p-1)/p_i^{\alpha_i}}\right)^x = h^{(p-1)/p_i^{\alpha_i}}$
2	1	11250	11250	1
3	2	5029	10724	4
5	4	5448	6909	511

Pohlig-Hellman: An example $\cdots$ cont

The second step is to use the CRT to solve the simultaneous congruences

$$x \equiv 1 \pmod 2, \quad x \equiv 4 \pmod{3^2}, \quad x \equiv 511 \pmod{5^4}$$

The smallest solution is $x = 4261$, so $23^{4261} = 9689$.
For more details, we explain how to solve $5448^x = 6909$. The first step is to solve

$$\left(5448^{5^3}\right)^{x_0} = 6909^{5^3}, \quad \text{i.e.} \quad 11089^{x_0} = 11089.$$

The answer is: $x_0 = 1$.
The next step is to solve

$$\left(5448^{5^3}\right)^{x_1} = (6909 \cdot 5448^{-x_0})^{5^2} = (6909 \cdot 5448^{-1})^{5^2}$$

Pohlig-Hellman: An example $\cdots$ cont

which reduces to $11089^{x_1} = 3742$. The answer is $x_1 = 2$. Continuing, we solve

$$\left(5448^{5^3}\right)^{x_2} = (6909 \cdot 5448^{-x_0 - x_1 \cdot 5})^5 = (6909 \cdot 5448^{-11})^5$$

which reduces to $11089^{x_2} = 1$, and thus $x_2 = 0$. The final step is to solve

$$\left(5448^{5^3}\right)^{x_3} = 6909 \cdot 5448^{-x_0 - x_1 \cdot 5 - x_2 \cdot 5^2} = 6909 \cdot 5448^{-11}$$

which reduces to $11089^{x_3} = 6320$, which has solution $x_3 = 4$.
The final answer is

$$x = 1 + 2 \cdot 5 + 4 \cdot 5^3 = 511\,.$$

Pohlig-Hellman: A cryptographically weak example

Consider a finite abelian group G of order

$$\#G = 2^{29}3^{21}5^{14}7^{5}11^{9}101^{3}$$

$\#G$ is a 160 bits number ...

Using Pohlig-Hellman with a exhaustive search for the discrete log on the (sub)groups of prime order, we can solve the DLP in less than 3000 group operations.

That's less than the cost of 12.5 scalar multiplications!

Shanks' Baby step - Giant step

Let $G = \langle g \rangle$, and n be a good upper bound of $\#G$. Let $u \approx \sqrt{n}$.

Considering the u-adic expansion of $x = DL_g(h)$

$$x = x_0 + ux_1, \text{ with } x_i \in [0, u-1],$$

we get

$$[x]g = h \Longleftrightarrow [x_1]([u]g) = h - [x_0]g\,.$$

To solve the DLP in G:

- We construct the list

$$S = \{h, h-[g], h-[2]g, \ldots, h-[u-1]g\} \quad \text{(Baby step)}$$

- We compute succesively the values $[x_1]([u]g)$ for $x_1 = 0, 1, \ldots$ and stop when such an element belongs to S (Giant step).

Cost

- We have u Baby steps, each taking 1 group operation.
- Computing $[u]g$ takes $O(\log u)$ group operations.
- We have u Giant steps, each taking 1 group operation.
- Additional cost for finding a match in the two lists: $O(u \log u)$
- The total cost is $u + u + O(\log u)$, which is $O(\sqrt{n})$.
- The memory requirements is also $O(\sqrt{n})$.

Random walks

Let G be a finite group of order N (in practice $G = \langle g \rangle$).

- A random map is a function $F : G \longrightarrow G$ such that the image of $x \in G$ is chosen (uniformly) at random in G.
- A random walk in G is a sequence of elements of G, starting at x_0, such that $x_{i+1} = F(x_i)$. The sequence $x_0, x_1, x_2, \ldots$ is eventually periodic (G is finite). We are interested in the value of i for which the first repetition occurs.
- Claim: The average time for the first repetition is $\sqrt{\pi/2}\sqrt{N}$.
- Proof: Starting from x_0, choose the image of x_i at random the first time you see x_i. The first repetition occurs at the first time when your random choice is an element that was chosen at a previous step. Use the Birthday Paradox.

Pollard ρ

Once again, we want to compute $DL_g(h)$ for $h \in G = \langle g \rangle$, a group of prime order N.

If we define

$$F(x) = [\alpha_x]g + [\beta_x]h,$$

and $x_0 = [\alpha_0]g + [\beta_0]h$ for randomly chosen $\alpha_x, \beta_x, \alpha_0$ and β_0, then the the first repetition (the point where we close the loop) gives us a relation of the form

$$[\alpha_i]g + [\beta_i]h = [\alpha_j]g + [\beta_j]h$$

Pollard ρ

We group the $g's$ and $h's$ together, and we get:

$$[\beta_i - \beta_j]h = [\alpha_j - \alpha_i]g\,.$$

With a little bit of luck, $\gcd(N, \beta_i - \beta_j) = 1$, and we have

$$DL_g(h) \equiv (\alpha_j - \alpha_i)/(\beta_i - \beta_j) \pmod{N}\,.$$

The expected time for the algorithm is $O(\sqrt{N})$.

But in this form, the algorithm has memory $O(\sqrt{N})$...

Although, it is possible to reduce the memory complexity to $O(1)$ using distinguished points and pseudo-Random walks (Floyd's method for cycles detection).

Pollard ρ - choice of the "random function"

- After getting the partition of G into three stes S_1, S_2, S_3 with $\mathcal{O} \notin S_2$, define the following "random walk":

$$x_{i+1} = F(x_i) = \begin{cases} h + x_i & \text{if } x_i \in S_1 \\ 2x_i & \text{if } x_i \in S_2 \\ g + x_i & \text{if } x_i \in S_3 \end{cases}$$

- If $x_i = [\alpha_i]g + [\beta_i]h$ then

$$\alpha_{i+1} = \begin{cases} \alpha_i & \text{if } x_i \in S_1 \\ 2\alpha_i & \text{if } x_i \in S_2 \\ \alpha_i + 1 & \text{if } x_i \in S_3 \end{cases}$$

and

$$\alpha_{i+1} = \begin{cases} \beta_i + 1 & \text{if } x_i \in S_1 \\ 2\beta_i & \text{if } x_i \in S_2 \\ \beta_i & \text{if } x_i \in S_3 \end{cases}$$

Floyd's cycles detection

- Given (x_1, x_2) compute (x_2, x_4), then (x_3, x_6) and so on $\cdots$
- Given the pair (x_i, x_{2i}), we compute $(x_{i+1}, x_{2i+2}) = (F(x_i), F(F(x_{2i})))$
- We stop when we find a collision: $x_m = x_{2m}$.
- **Exercise:** Prove that it the tails has length λ and the cycle length μ, then

$$m = \mu \left\lceil \frac{\lambda}{\mu} \right\rceil$$

- Since $\lambda \leq m \leq \lambda + \mu$, we see that $m = O(\sqrt{N})$.
- Detect a collision with $O(1)$ storage.

Theorem: (V. Shoup)

In a ”black box group” of prime order ℓ it takes at least $O(\sqrt{\ell})$ operations to solve the discrete logarithm problem.

Principal goals of the Cryptography

- Historically, the most important goal of the cryptography was to secure private communication (Encryption).
- Nowadays, there are other goals
  - authentification
  - non-repudiation
  - integrity

The discover of public key cryptography provides methods to realize the above goals:

- asymmetric encryption
- Signature
- Key exchange (for session key in symmetric encryptions)
- electronic voting, etc ...

Diffie-Hellman Key exchange

Let $G = \langle g \rangle$ be a finite abelian cyclic group of order N.

Alice	unsecure channel	Bob
choose $x_A \in_R [1, N]$		
compute $k_A := [x_A]g$	$\longrightarrow k_A$	
		choose $x_B \in_R [1, N]$
	$k_B \longleftarrow$	compute $k_B := [x_B]g$
compute $k_{AB} := [x_A]k_B$		compute $k_{AB} := [x_B]k_A$

Massey-Omura encryption

Let G be a finite cyclic group of prime order N. We consider message (to encrypt) as elements m of G.

Alice	unsecure channel	Bob
choose $x_A \in_R [1, N]$ s.t. $\gcd(x_A, N) = 1$. compute $a := [x_A]m$	$\longrightarrow a$	
		choose $x_B \in_R [1, N]$ s.t. $\gcd(x_B, N) = 1$. compute $b := [x_B]a = [x_A x_B]m$
	$b \longleftarrow$	
compute $a' := [x_A^{-1}]b = [x_B]m$	$\longrightarrow a'$	compute $b' := [x_B^{-1}]a' = m$

Massey-Omura

- The eavesdropper knows $[x_A]m, [x_A x_B]m$ and $[x_B]m$. If we denote $y_A = x_A^{-1}, y_B = x_B^{-1}$ and $h = [y_A y_B]m$, then we see that Eve knows $h, [y_A]h, [y_B]h$ and wants to find $[y_A y_B]h$, this is the DHP.
- This protocol requires an authentication scheme: If C pretends to be B, then A will send $[x_C]m$ to C and so C can read the message m.
- This encryption scheme is purely from theoretical interest (pedagogic) and is rarely used in practice.

  It is more convenient to generate a session key (via Diffie-Hellman) for a use in a symmetric encryption (hybrid encryption).
- Principle: Both users are concerned to encrypt a message m.
- Crucial point: the encryption is probabilistic.

ElGamal Encryption (1984)

- public parameters: A finite cyclic group $G = \langle g \rangle$.
- Bob's public key: $h = [x]g$
- Bob's private key: x
- To encrypt a message $m \in G$ that Alice want to send to Bob, Alice use the public key h of Bob and choose $k \in_R [1, N-1]$ to compute

$$a = [k]g\,, \text{ and } b = [k]h + m\,.$$

- Alice send (a, b) to Bob.
- Bob can recover the message by computing

$$b - [x]a = [k]h + m - [kx]g = [kx]g - [kx]g + m = m\,.$$

Security of ElGamal

- ElGamal has a 2-to-1 message expansion.
- **Theorem.** ElGamal is secure against chosen cipertext attacks. More precisely, suppose Eve has access to an oracle that decrypts arbitrary ciphertexts encrypted using arbitrary ElGamal public keys. Then she can use the oracle to solve the DHP. So it is secure if one assumes that DHP is hard.
- **Proof:** Eve is given the two values $[n_1]g$ and $[n_2]g$ and she is required to compute $[n_1 n_2]g$.
  Eve chooses an arbitrary value for c_2 and tells the oracle that the public key is $[n_1]g$ and the ciphertext is $([n_2]g, c_2)$. The oracle returns to her the supposed plaintext m that satisfies

$$m = c_2 - [n_1]c_1 = c_2 - [n_1 n_2]g$$

  After the oracle tells the value of m, she just computes $-m + c_2 = [n_1 n_2]g$

Security of ElGamal

- ElGamal has a 2-to-1 message expansion.
- **Theorem.** ElGamal is secure against chosen cipertext attacks. More precisely, suppose Eve has access to an oracle that decrypts arbitrary ciphertexts encrypted using arbitrary ElGamal public keys. Then she can use the oracle to solve the DHP. So it is secure if one assumes that DHP is hard.
- **Proof:** Eve is given the two values $[n_1]g$ and $[n_2]g$ and she is required to compute $[n_1 n_2]g$.
  Eve chooses an arbitrary value for c_2 and tells the oracle that the public key is $[n_1]g$ and the ciphertext is $([n_2]g, c_2)$. The oracle returns to her the supposed plaintext m that satisfies

$$m = c_2 - [n_1]c_1 = c_2 - [n_1 n_2]g$$

  After the oracle tells the value of m, she just computes $-m + c_2 = [n_1 n_2]g$

ElGamal encryption example

- Prime $p = 809$, then $909 - 1 = 808$ is divisble by $q = 101$.
- Compute a generator $g = 16$ of subgroups of $\mathbb{F}_p^*$ of order q.
- Alice chooses the private key $a = 68$ and computes

$$g^a = 16^{68} \equiv 46 \pmod{p}$$

- Alice public key is $(p = 808, g = 16, h = 46)$ which can be published.
- Alice private key is $a = 68$ which she keep secret.

ElGamal encryption example

- To encrypt the message $m = 100$, Bob selects a random integer $k = 89$ and computes

$$r = g^k = 342 \quad \text{and} \quad s = mh^k = 745$$

- Bob then sends the ciphertext (r, s) to Alice.
- To decrypt, Alice first computes

$$r^{-a} = 342^{33} = 49 \pmod{809}$$

and recovers m by computing

$$m = 745 \cdot 49 \equiv 100 \pmod{809}.$$

ElGamal Signature

- public parameters: A finite cyclic group $G = \langle g \rangle$.
- Bob's public key: $h = [x]g$
- Bob's private key: x
- Hypothesis: There is a (public function) $f : G \longrightarrow \mathbb{Z}/N\mathbb{Z}$.
- To sign a message $m \in [1, N-1]$, Bob choose $k \in_R [1, N-1]$ to compute $a = [k]g$.
- Bob compute $b \in \mathbb{Z}/N\mathbb{Z}$ with

$$m \equiv xf(a) + bk \pmod{N}.$$

- Bob send the message m and its signature $s = (a, b)$ to Alice.
- Alice accepts the signature if

$$[f(a)]h + [b]a = [xf(a) + kb]g = [m]g.$$

Security of those protocols

The security of those protocols depends on

- The choice of the (pseudo-) random generators
- The problem of distribution of public key's (PKI)
- The choice of hash function
- Hardware attacks, etc ...

Furthermore, for those simple protocols, we do not know if their security is equivalent to the DLP (but for CDHP).

Suitable groups

A cryptographically suitable group G must satisfy:

- Representation of its elements in an easy and compact way.
- Fast arithmetic, i.e. fast scalar multiplication.
- DLP is computationally hard, in best case only the generic methods works.

Consequence of Pohlig-Hellman reduction: It is important to know the group order, or better to compute it efficiently. Furthermore, the value or this order is used in some protocols.

The minimal amount of computations that we suppose infeasible is $\approx 2^{80}$.

$\Longrightarrow$ The cardinality of the group order should have at least a 160-bit prime factor to avoid the generic attacks.

Finite fields

- Prime fields: $q = p$
  - Multiplication: product of two integers, and reduction modulo p.
  - Inverse: extended euclidian algorithm.
- Finite fields of characteristic 2 ;

$$\mathbb{F}_2[x]/(f(x)) = \left\{ \sum_{i=0}^{n-1} c_i x^i \ : c_i \in \mathbb{F}_2, 0 \leq i < n \right\}.$$

  - Multiplication : product of polynomials with coefficients in $\mathbb{F}_2$, and reduction modulo the defining polynomial $f(x)$.
  - Inverse: extended euclidian algorithm for polynomials.

$\Longrightarrow$ Extremly efficient arithmetic on those finite fields.

Index calculus attacks in prime fields

- Index calculus is a method to compute discrete logarithms, also called indices.
- p prime, elements of $\mathbb{F}_p$ represented by numbers in $\{0, 1, \ldots, p-1\}$; g generator of multiplicative group.
- If $h \in \mathbb{F}_p$ factors as $h = h_1 \cdot h_2 \cdots h_n$ then

$$h = g^{a_1} \cdot g^{a_2} \cdots g^{a_n} = g^{a_1 + a_2 + \cdots + a_n}$$

  with $h_i = g^{a_i}$.
- Knowledge of the a_i, i.e. the discrete logarithms of h_i to base g gives knowledge of the discrete logarithm of h to base g.
- If h factors appropriately ...

Smooth numbers

An integer is said to be B-smooth if its decomposition in prime factors only contains primes $p \leq B$.

To evaluate the proportion of smooth numbers, we introduce the function

$$\phi(x,y) = \#\{1 \leq n \leq x\,;\; n \text{ is } y-\text{smooth}\}\,.$$

For $y = 23$ we obtain the following proportions:

x	100	1000	10000	100000
$\frac{\phi(x,y)}{x}$	76 %	37 %	14 %	4 %

Subexponentiality

Definition: subexponential functions

- Let $N > 0, 0 \leq \alpha \leq 1, c > 0$.

$$L_N(\alpha, c) := \exp\left(c(\log N)^{\alpha}(\log\log N)^{1-\alpha}\right)$$

- If $\alpha = 0$, then $L_N(\alpha, c) = (\log N)^c$: polynomial in the length of N.
- If $\alpha = 1$, then $L_N(\alpha, c) = \exp c(\log N) = N^c$: exponential in the length of N.
- We say that $L_N(\alpha, c)$ is subexponential if $0 < \alpha < 1$.

N.B.: There exists algorithms for the "special" integer factorization ($n = p \cdot q$) with a subexponential running time: the fastest known method is the Number field sieve with time complexity

$$O\left(\exp\left((1.923 + o(1))(\log N)^{\frac{1}{3}}(\log\log N)^{\frac{2}{3}}\right)\right)$$

where $o(1) = \theta(n) \longrightarrow 0$ for $n \longrightarrow +\infty$.

Smoothness theorem

Theorem (Canfield, Erdós, Pomerance)

Fix a number $0 < \varepsilon < 1$ and let X and B increase together while satisfying

$$(\ln X)^{\varepsilon} < \ln B < (\ln X)^{1-\varepsilon}$$

Then the number of B-smooth numbers less than X satisfies

$$\phi(X, B) = X \cdot U^{-U(1+o(1))}$$

where $U = \frac{\ln X}{\ln B}$.

Theorem fundamental

For any $c > 0$, when $x \longrightarrow +\infty$, then

$$\frac{\phi(x, L_x(\frac{1}{2}, c))}{x} \sim \frac{1}{\sqrt{L_x(\frac{1}{2}, \frac{1}{c})}} \sim \frac{1}{L_x(\frac{1}{2}, \frac{1}{2c})}$$

Adleman's algorithm in prime fields

Let p a prime number, g a generator of $\mathbb{F}_p^* = (\mathbb{Z}/p\mathbb{Z})^*, h \in \langle g \rangle$.

- Choice of the "factors base":
  - Bound of smoothness B,
  - $\mathcal{F}_B = \{p_i, p_i \text{ prime }, p_i < B\}$.
  - How to compute the $DL_g(p_i)$ for the $p_i \in \mathcal{F}_B$? ($p_i = g^{DL_g(p_i)}$)
- Find "some relations":
  - For a random $r \in_R [0, p-2]$, compute $g^r \pmod p$.
  - If the obtained number is B-smooth, it gives "a relation"

$$g^r = \prod_{p_i \in \mathcal{F}_B} p_i^{\alpha_i} = \prod_{p_i \in \mathcal{F}_B} g^{DL_g(p_i)\alpha_i} = g^{\sum_{p_i \in \mathcal{F}_B} DL_g(p_i)\alpha_i}$$

  such that $r \equiv \sum_{p_i \in \mathcal{F}_B} DL_g(p_i)\alpha_i \pmod{p-1}$.
  - Iterate the last step to get at least $\#\mathcal{F}_B$ relations.

Adleman's algorithm ...

- Linear algebra:
  - We have a linear system (in the unknown $DL_g(p_i)$) with more equations than unknown. We solve it to obtain $DL_g(p_i)$ for all p_i.
  - This step needs to be done only once per field and generator, it does not depend on the target DLP $h = g^x$.
- Solving the original DLP:

  How now to solve the DLP for $h \in \langle g \rangle$, i.e. how to compute $DL_g(h)$?
  Choose randomly $r \in [1, p-2]$ until $g^r \cdot h \pmod{p}$ is B-smooth.
  Then,

$$g^r \cdot h = \prod_{p_i \in \mathcal{F}_B} p_i^{\beta_i} \text{ and thus } DL_g(h) = \sum_{p_i \in \mathcal{F}_B} DL_g(p_i)\beta_i - r\,.$$

Adleman's algorithm . . . an example

Given a prime $p = 18443$ and base element $g = 37$ we want to find x such that

$$37^x \equiv 211 \pmod{18443}.$$

- We use the factor base $\mathcal{F} = \{2, 3, 5\}$.
- We need at least 3 relations: by choosing random powers of 37 $\pmod{18443}$, we get the $\mathcal{F}$-smooth numbers:

$$\begin{aligned} g^{12708} &\equiv 2^3 \cdot 3^4 \cdot 5 \pmod{p} \\ g^{11311} &\equiv 2^3 \cdot 5^2 \pmod{p} \\ g^{15400} &\equiv 2^3 \cdot 3^3 \cdot 5 \pmod{p} \\ g^{2731} &\equiv 2^3 \cdot 3 \cdot 5^4 \pmod{p} \end{aligned}$$

Adleman's algorithm . . . an example

- Using the notations

$$x_2 = \log_g(2), \quad x_3 = \log_g(3), \quad x_5 = \log_g(5),$$

we get the linear relations

$$\begin{array}{rcl} 12708 & \equiv & 3x_2 + 4x_3 + x_5 \pmod{18442} \\ 11311 & \equiv & 3x_2 + 2x_5 \pmod{18442} \\ 15400 & \equiv & 3x_2 + 3x_3 + x_5 \pmod{18442} \\ 2731 & \equiv & 3x_2 + x_3 + 4x_5 \pmod{18442} \end{array}$$

- Note that the above congruences are modulo $p - 1 = 18442 = 2 \cdot 9221$.

Adleman's algorithm . . . an example

- Solving these linear system modulo 2 and 9221, we get

$$\begin{array}{rlll} & (x_2, x_3, x_5) & \equiv & (1,0,1) \pmod 2 \\ & (x_2, x_3, x_5) & \equiv & (5733, 6529, 6277) \pmod{9221} \\ \Longrightarrow & (x_2, x_3, x_5) & \equiv & (5733, 15750, 6277) \pmod{18442} \end{array}$$

- To solve $37^x = 211 \pmod{18443}$, we compute the value of $211 \cdot 37^{-k} \pmod{18443}$ for random values of k until we find an $\mathcal{F}$ -smooth number:

$$211 \cdot 37^{-9549} \equiv 2^5 \cdot 3^2 \cdot 5^2 \pmod{18443}$$

$$\begin{array}{rll} \Longrightarrow & \log_g(211) & = 9549 + 5\log_g(2) + 2\log_g(3) + 2\log_g(5) \\ & & \equiv 8500 \pmod{18442}. \end{array}$$

Analysis of Adleman's algorithm

Principle

It is much easier to find some relation if B is large, however we then need much more relation (since $\mathcal{F}_B$ will be large too)!

We will choose B to be of the form

$$B = L_p\left(\frac{1}{2}, \rho\right).$$

From the smoothness theorem, the probability that a random element in $\mathbb{F}_p^*$ is B-smooth is

$$\mathbb{P} = \frac{1}{L_p\left(\frac{1}{2}, \frac{1}{2\rho}\right)}.$$

Analysis of Adleman's algorithm

- The average time we will need to find the $\#\mathcal{F}_B$ relation is:

$$L_p\left(\frac{1}{2}, \frac{1}{2\rho}\right) \cdot L_p\left(\frac{1}{2}, \rho\right) = L_p\left(\frac{1}{2}, \rho + \frac{1}{2\rho}\right).$$

- Linear algebra: The matrix representing the linear sytem is sparse ($O(\log p)$ non zero terms in each row). We can then use adequate algorithms with quadratic (in the length of the matrix) running time.

  The cost of the linear algebra is:

$$L_p\left(\frac{1}{2}, \rho\right)^2 = L_p\left(\frac{1}{2}, 2\rho\right).$$

Analysis of Adleman's algorithm

- The cost of the final step (the smoothness relation of $g^r \cdots$) is equivalent to the cost of one smoothness relation.
- The total cost of the algorithm is

$$L_p\left(\frac{1}{2}, 2\rho\right) + L_p\left(\frac{1}{2}, \rho + \frac{1}{2\rho}\right) = L_p\left(\frac{1}{2}, \max\left(2\rho, \rho + \frac{1}{2\rho}\right)\right).$$

- The optimal value is obtained when $\rho = \frac{1}{\sqrt{2}}$, which gives the complexity

$$L_p(\frac{1}{2}, \sqrt{2}).$$

- Running time with much more clever way of finding relations is

$$O\left(\exp\left((1.923 + o(1))(\log p)^{\frac{1}{3}}(\log\log p)^{\frac{2}{3}}\right)\right)$$

Let $q = 2^n$. The field with q elements $\mathbb{F}_q$ is isomorphic to

$$\mathbb{F}_2[x]/(f(x)) = \left\{ \sum_{i=0}^{n-1} c_i x^i \ : \ c_i \in \mathbb{F}_2, 0 \leq i < n \right\}$$

where $f \in \mathbb{F}_2[x]$ is an irreducible polynomial of degree n.
Adleman's algorithm can be trivially extended to such fields :

- Factoring into powers of small primes is replaced by factoring into irreducible polynomials of small degree.
- Same approach works, same problem of balancing size of factorbase (and thus complexity of the matrix step) and the likelihood of splitting completely over the factors base.

Example in characteristic two $\cdots$

Given the field $\mathbb{F} = \mathbb{F}_{2^{11}} = \mathbb{F}_2[x]/(f)$ where f is the irreducible polynomial $f = x^{11} + x^8 + x^6 + x^2 + 1$.

The field $\mathbb{F}$ is generated by x. Using the factor base $\mathcal{F} = \{x, x+1, x^2+x+1\}$, we want to find n such that

$$x^n \equiv b \pmod{f}$$

where b is the polynomial $b = x^{10} + x^9 + x^7 + x^6 + x^3 + x^2 + 1$.

- We find as before a relation:

$$x^8 \cdot b \equiv (x+1)^2 \cdot (x^2+x+1) \pmod{f}$$

such that

$$b \equiv x^{2039}(x+1)^2 \cdot (x^2+x+1) \pmod{f}$$

since $x^8 \cdot x^{2039} \equiv 1 \pmod{f}$.

Example in characteristic two $\cdots$

- Now we would like to compute $n_1 = \log_x(x+1)$ and $n_2 = \log_x(x^2+x+1)$, i.e.

$$x^{n_1} \equiv x+1 \pmod{f}, \quad x^{n_2} \equiv x^2+x+1 \pmod{f}$$

- We compute the two relations:

$$\begin{aligned} x^{11} &\equiv (x+1)^4 \cdot (x^2+x+1)^2 \pmod{f} \\ x^{94} &\equiv (x+1)^3 \cdot (x^2+x+1)^3 \pmod{f} \end{aligned}$$

and we get the congruences system

$$\begin{aligned} 11 &\equiv 4n_1 + 2n_2 \pmod{\#\mathbb{F}-1} \\ 94 &\equiv 3n_1 + 3n_2 \pmod{\#\mathbb{F}-1} \end{aligned}$$

- Note that the above congruences are modulo $\#\mathbb{F}-1 = 2047$.

Example in characteristic two $\cdots$

- Solving this linear system (with CRT), we get

$$n_1 \equiv 1680 \pmod{2047}, \quad n_2 \equiv 1763 \pmod{2047}$$

- To solve $x^n = b \pmod{f}$, we compute

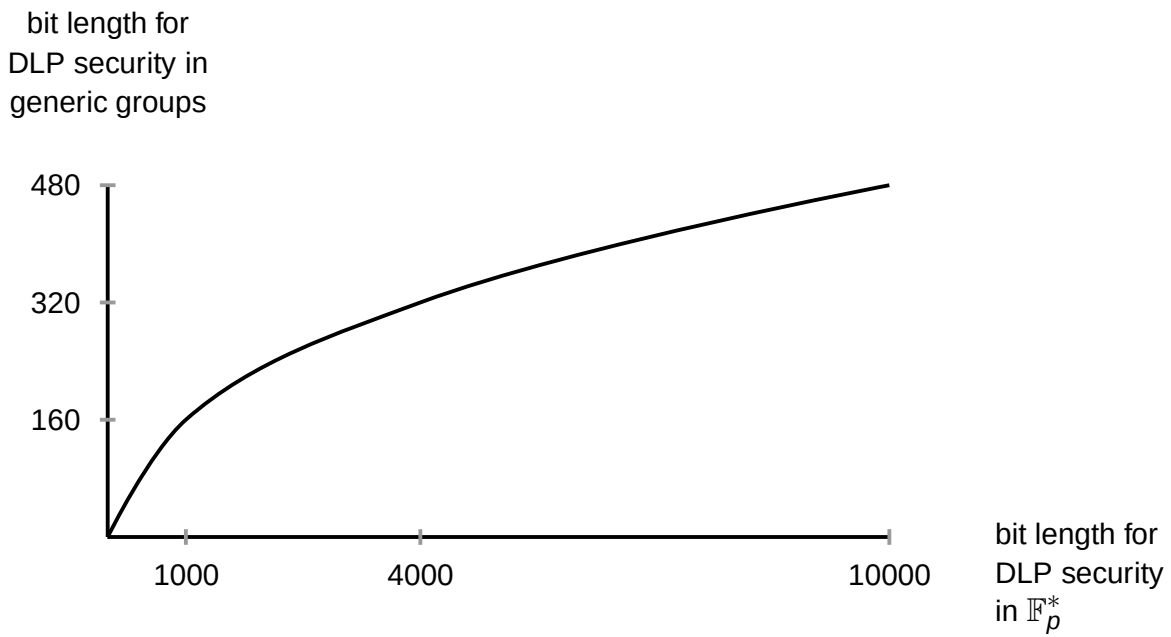
$$b = x^{2039} \cdot x^{2n_1} \cdot x^{n_2} = x^{7162=} \equiv x^{3 \cdot 2047} \cdot x^{1021} \equiv x^{1021} \pmod{f}$$

Cryptographic interests

Best known attack for $G = \mathbb{F}_q^*$: $L_q\left(\frac{1}{3}, c\right)$
Best known attack for generic groups: $2^{n/2}$

For the same security level, the bit length of the group order

of generic groups beahves like the cubic root of the bit length of $\#\mathbb{F}_q^*$



Elliptic curves

Let $K = \mathbb{F}_q$ be the finite field with q elements. An elliptic curve over K is given by a non-singular equation

$$(1) \quad E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$

where $a_i \in K$. For a field extension L of K, the set of rational points of E is

$$E(L) := \left\{(x,y) \in L^2 : (x,y) \text{ satisfy } (1)\right\} \cup \{\mathcal{O}\},$$

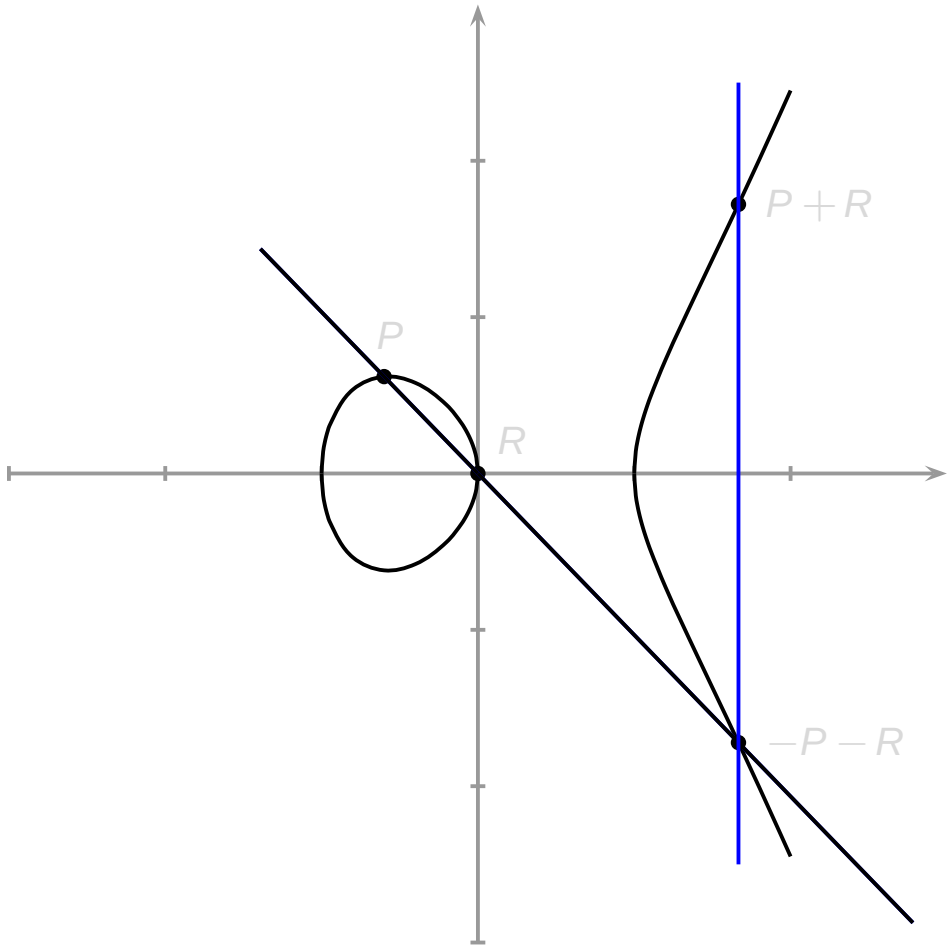
where $\mathcal{O}$ denotes the point at infinity.

A point of E is an element of $E(\bar{K})$ where $\bar{K}$ is the algebraic closure of K.

For any extension L of K, the set $E(L)$ forms an abelian group with identity element $\mathcal{O}$.

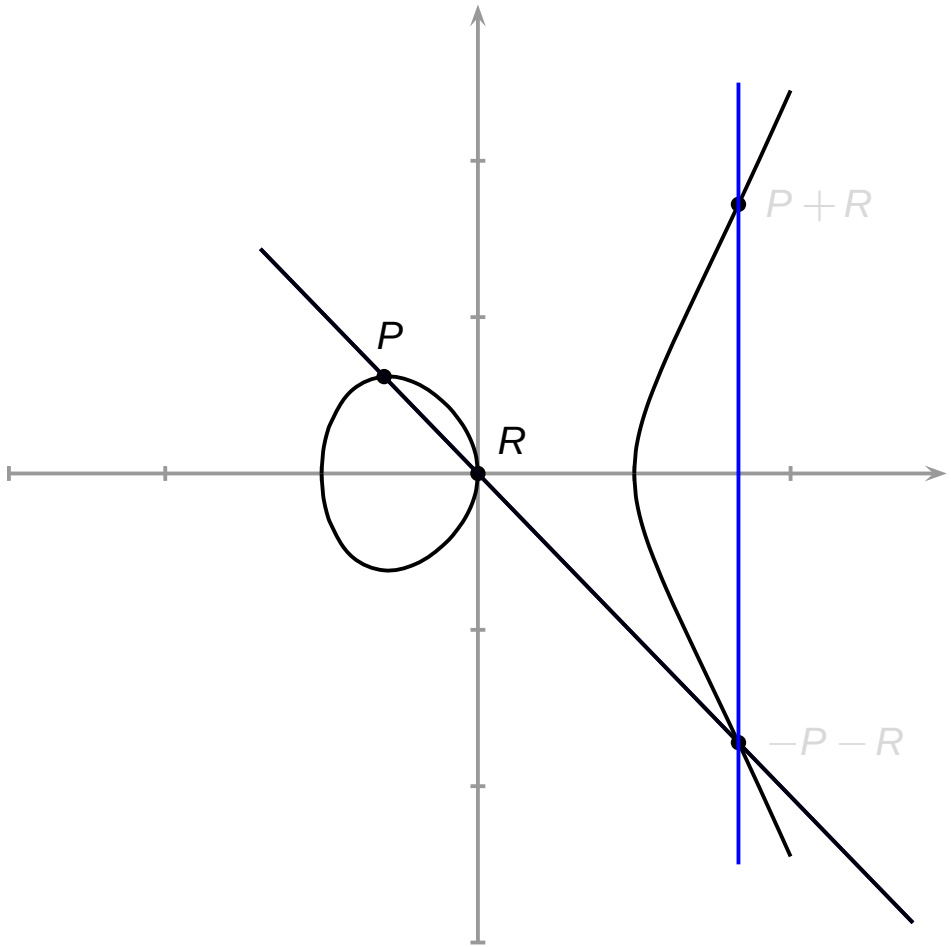
Elliptic curves: group law in $E(\mathbb{R})$

$E : y^2 = x^3 - x$



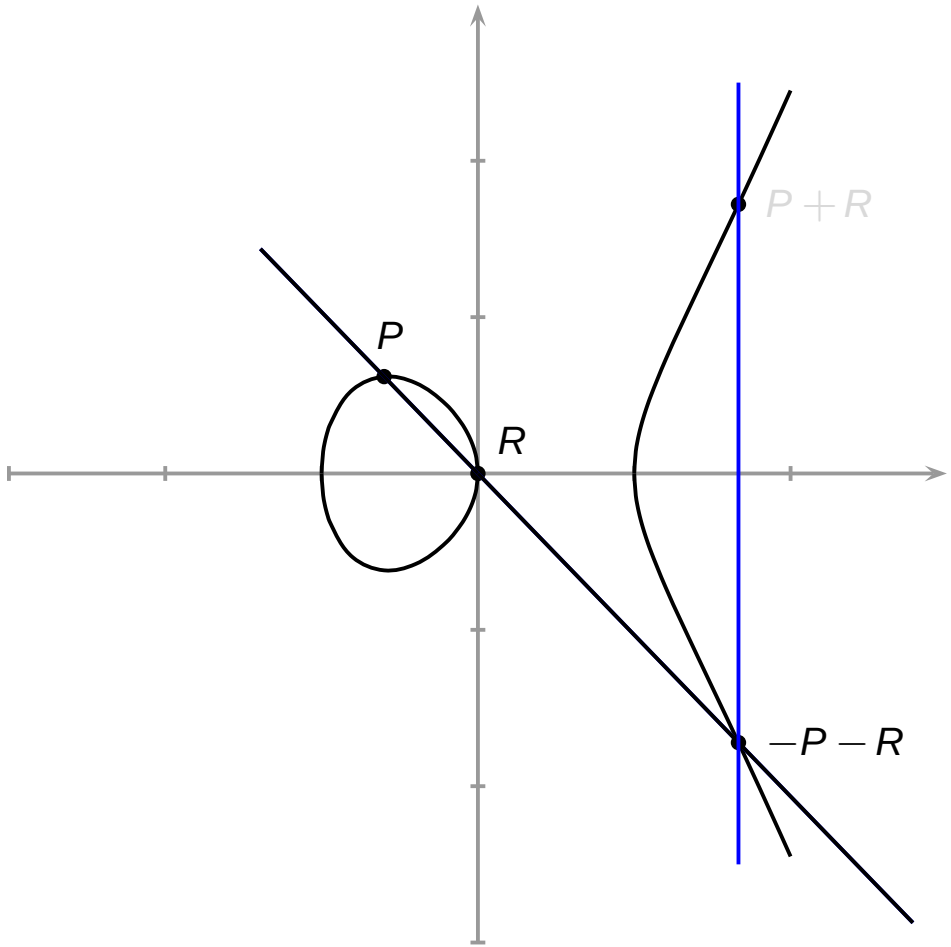
Elliptic curves: group law in $E(\mathbb{R})$

$E : y^2 = x^3 - x$



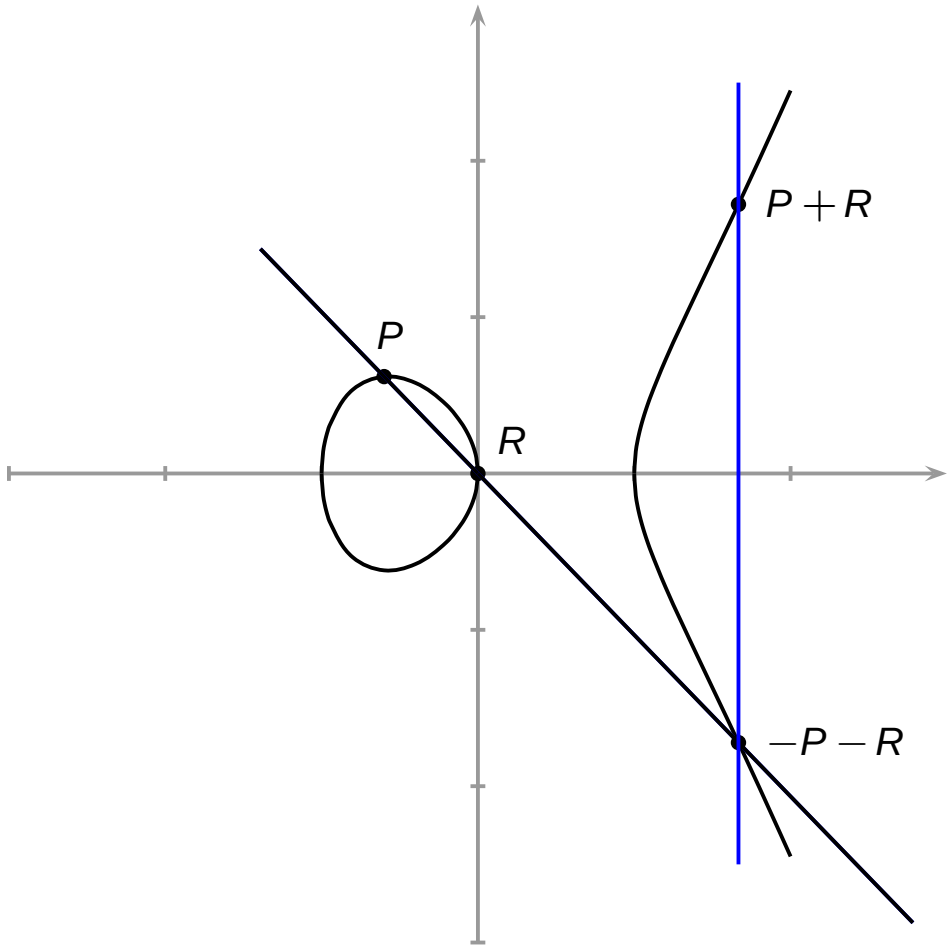
Elliptic curves: group law in $E(\mathbb{R})$

$E : y^2 = x^3 - x$



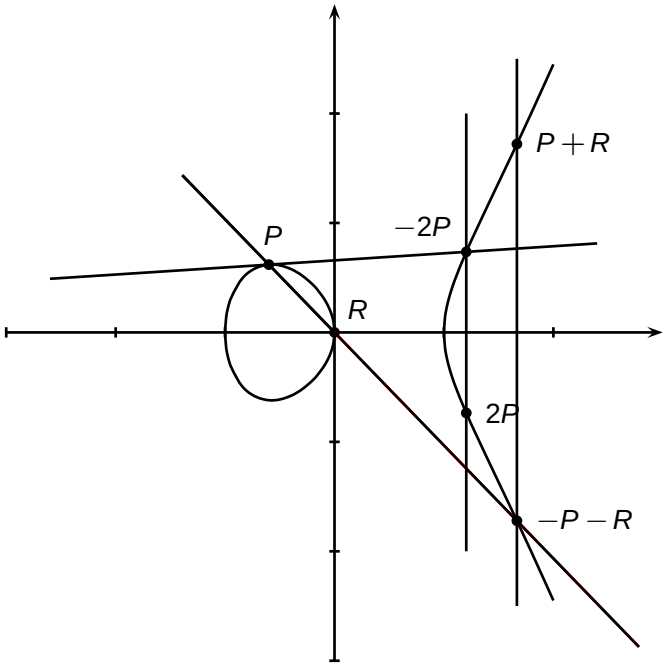
Elliptic curves: group law in $E(\mathbb{R})$

$E : y^2 = x^3 - x$



Elliptic curves: group law (q odd)

$$E : y^2 = x^3 + a_4x + a_6,\ a_i \in \mathbb{F}_q$$



For $(x_1, y_1) \neq (x_2, -y_2)$:

$$(x_1, y_1) \oplus (x_2, y_2) = (x_3, y_3)$$
$$= (\lambda^2 - x_1 - x_2, \lambda(x_1 - x_3) - y_1),$$

avec

$$\lambda = \begin{cases} (y_2 - y_1)/(x_2 - x_1) & \text{si } x_1 \neq x_2, \\ (3x_1^2 + a_4)/(2y_1) & \text{si } x_1 = x_2 \end{cases}$$

$\Rightarrow$ Addition and Doubling differ considerably.:

1 I, 2M, 1S vs. 1 I, 2M, 2S

Projective Coordinates

$P = (X_1 : Y_1 : Z_1), Q = (X_2 : Y_2 : Z_2), P \oplus Q = (X_3 : Y_3 : Z_3)$ on
$E : Y^2Z = X^3 + a_4XZ^2 + a_6$

Addition: $P \neq \pm Q \quad A = Y_2Z_1 - Y_1Z_2, B = X_2Z_1 - X_1Z_2$
$C = A^2Z_1Z_2 - B^3 - 2B^2X_1Z_2$
$X_3 = BC, Z_3 = B^3Z_1Z_2$
$Y_3 = A(B^2X_1Z_2 - C) - B^3Y_1Z_2$

Doubling: $P = Q \neq -P$
$A = a_4Z_1^2 + 3X_1^2, B = Y_1Z_1,$
$C = X_1Y_1B, D = A^2 - 8C$
$X_3 = 2BD, Z_3 = 8B^3.$
$Y_3 = A(4C - D) - 8Y_1^2B^2$

No inversion is needed and the computation times are $12M + 2S$ for a general addition and $7M + 5S$ for a doubling.

$\cdots$ and other different coordinates systems for $y^2 = x^3 + ax + b$

system	points	correspondence
affine ($\mathcal{A}$)	(x, y)	
projective ($\mathcal{P}$)	(X, Y, Z)	$(X/Z, Y/Z)$
jacobi ($\mathcal{J}$)	(X, Y, Z)	$(X/Z^2, Y/Z^3)$
Chudnovsky jacobi ($\mathcal{J}^C$)	(X, Y, Z, Z^2, Z^3)	$(X/Z^2, Y/Z^3)$
jacobi modified ($\mathcal{J}^m$)	(X, Y, Z, aZ^4)	$(X/Z^2, Y/Z^3)$

system	addition			doubling		
affine ($\mathcal{A}$)	2M	1S	1I	2M	2S	1I
projective ($\mathcal{P}$)	12M	2S	–	7M	5S	–
jacobi ($\mathcal{J}$)	12M	4S	–	4M	6S	–
Chudnovsky jacobi ($\mathcal{J}^C$)	11M	3S	–	5M	6S	–
jacobi modified ($\mathcal{J}^m$)	13M	6S	–	4M	4S	–

New efficient and "complete" formulae using Edward's model for elliptic curves: $\Longrightarrow$ Lange & Berstein's talks.

Number of points

Hasse's theorem

In cryptograhy, we usually consider elliptic curves over finite fields $\mathbb{F}_q$.

The number of $\mathbb{F}_q$-rational points of E is also finite, a bound is given by Hasse's theorem:

$$\#E(\mathbb{F}_q) = q + 1 - t,$$

with $|t| \leq 2\sqrt{q}$. The integer t is called the trace of E.

DLP on special elliptic curves

For a ”generic” elliptic curve, the best known attack is Pollard ρ (combined with Pohlig-Hellman).

$\Longrightarrow$ Elliptic curves behave like generic groups.

Although, there are some classes of specific curves with much faster attack :

- MOV Reduction
- Anomalous curves
- Curves with non-trivial automorphisms group
- Weil descent

MOV reduction

Definition

Let G a subgroup of $E(\mathbb{F}_q)$ of prime order $N | \#E(\mathbb{F}_q)$. The MOV degree is the smallest integer k such that $N | q^k - 1$.

Theorem (Menezes-Okamoto-Vanstone, Frey-Rück)

The DLP in G can be reduced to the DLP in $\mathbb{F}_{q^k}^*$.

Idea of the proof: Use the Weil pairing to embedd G in $\mathbb{F}_{q^k}$.

Remark: The DLP can be solved in a subexponential running time in $\mathbb{F}_{q^k}$. However, for a random elliptic curve E, k is very large!

For elliptic curves with trace $t = 0$, we then have $\#E(\mathbb{F}_p) = p + 1 | p^2 - 1$ and thus $k = 2$. Supersingular elliptic curves over prime fields are thus less suitable for DLP based cryptography .

Weil descent and anomalous curves

Weil descent

In some case, the DLP in $E(\mathbb{F}_{2^n})$ can be reduced in a DLP of an hyperelliptic curve of large genus over a smaller field.

There exists subexponential attacks for large genus curves.

The curves defined over $E(\mathbb{F}_{2^n})$ where n is composite are in danger regarding this attack.

An anomalous elliptic curve is a curve over $\mathbb{F}_p$ with $\#E(\mathbb{F}_p) = p$, such that $\#E(\mathbb{F}_p) \simeq (\mathbb{F}_p, +)$.

Theorem (Smart, Satoh-Araki, Semaev)

The above isomorphism can be given explicitly.

The DLP on such groups can be computed very efficiently.

Standards

- ANSI Public Key Cryptography for the Financial Services Industry
  - X9.62-1998 – The Elliptic Curve Digital Signature Algorithm (ECDSA)
  - X9.63-1999 – Key Agreement and Key Transport Using Elliptic Curve Cryptography (ECIES etc.)
- NIST – FDigital Signature Standard FIPS 186-2 (revision 2000)
- IEEE P1363a – Standart Specifications for Public Key Cryptography
- Standarts for Efficient Cryptography Group (Certicom)
- ISO 15946

What is a Pairing

The pairing explosion $\cdots$

- Pairings originally used destructively in MOV / Frey-Rück attack.
- 2000 / 2001: Papers by Sakai-Ohgish-Kasahara, Joux, Boneh-Franklin.

Basic properties

- Finite groups $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}$, all of prime order r.
- A **bilinear** map $e : \mathbb{G}_1 \times \mathbb{G}_2 \longrightarrow \mathbb{G}$, i.e.
  - $e(P+Q,R) = e(P,R) \cdot e(Q,R)$
  - $e(P,R+S) = e(P,R) \cdot e(P,S)$
  - It follows: $e(aP,bR) = e(P,R)^{ab} = e(bP,aR) = \cdots$
- Nondegeneracy: for every $\mathcal{O} \neq P$ in $\mathbb{G}_1$ there exists a $Q_2 \in \mathbb{G}_2$ s.t. $e(P,Q) \neq 1$.
- Computability: $e(P,R)$ can be efficiently computed.

DDH problem is easy using "nice pairing" in $\mathbb{G}_1 \times \mathbb{G}_1$.

One round-Joux's 3-Partite Diffie-Hellman key exchange

Let $\mathbb{G}$ be a group of prime order q, $e : \mathbb{G} \times \mathbb{G} \longrightarrow \mathbb{G}_t$ be a bilinear map, and g be a generator of $\mathbb{G}$. Let $\hat{g} = e(g, g) \in \mathbb{G}_t$.

- Alice picks $a \in_R \mathbb{Z}_q$, Bob picks $b \in_R \mathbb{Z}_q$, and Carls picks $c \in_R \mathbb{Z}_q$.
- Alice, Bob, and Carls compute (and publish) g^a, g^b, and g^c respectively.
- Alice computes $e(g^b, g^c)^a = \hat{g}^{abc}$, Bob computes $e(g^c, g^a)^b = \hat{g}^{abc}$, and Carls computes $e(g^a, g^b)^c = \hat{g}^{abc}$.

ID based cryptography

- Sakai-Ohgishi-Kasahara (2000) ID-based key exchange, Boneh and Franklin (Crypto 2001), ID-based cryptography.
- Idea: user's identity defines his public key.

Consequences: Advantage in ID-based crypto if recipient is not in system or sender wants to force use of a fresh key (other applications possible).

- No need for PKI, can avoid need for authentication.
- Set-up requires a trusted authority (TA) which can compute the secret key for a given public key.

Boneh-Franklin IBE

- Select two hash functions $H_1 : \{0,1\}^* \longrightarrow \mathbb{G}_1$ and $H_2 : \mathbb{G} \longrightarrow \{0,1\}^n$ where n is the length of the plaintexts.
- TA choose an arbitrary point $P \in \mathbb{G}_1$.
- Master secret key of TA is s, public key is $P_{pub} = [s]P$.
- Public key of ID represented by string ID is $H_1(ID) \in \mathbb{G}_1$.
- Secret key $d_{ID} = [s]H_1(ID) \in \mathbb{G}_1$ computable only by TA.
- Encryption: Inputs are message M and an identity ID.
  - Choose random $t \in \mathbb{Z}_r$
  - Compute the ciphertext $C = \langle [t]P, M \oplus H_2(e(H_1(ID), P_{pub})^t) \rangle$.
- Decryption: Given the ciphertext $\langle U, V \rangle$ and the private key d_{ID}, compute

$$M = V \oplus H_2(e(d_{ID}, U)).$$

Boneh-Franklin IBE

Both sender (who has t) and receiver (who has d_{ID}) can compute $e(H_1(ID), P)^{st}$:

$$e(H_1(ID), P)^{st} = e(H_1(ID), [s]P)^t = e(H_1(ID), P_{pub})^t$$

$$e(H_1(ID), P)^{st} = e([s]H_1(ID), [t]P) = e(d_{ID}, U).$$

Security assumptions

Just a few $\cdots$

- Clearly, these systems require that the DLP is hard in the groups.
- Additionally we define the following computational and decisional problems. To ease notation let $\mathbb{G}_1 = \mathbb{G}_2$ and $g = e(P, P)$.
  - Computational Bilinear Diffie-Hellman Problem (CBDHP): Compute $g^{s_A s_B s_C}$ given $[s_A]P, [s_B]P, [s_C]P$ and P.
  - Decisional Bilinear Diffie-Hellman Problem (DBDHP): Given $P, [s_A]P, [s_B]P, [s_C]P$ and g^r decide whether $g^r = g^{s_A s_B s_C}$.

Elliptic curves again $\cdots$

Torsion points

- Let $m \in \mathbb{N}$ and E/K be an elliptic curve defined over a field K. The set of *m*-torsion points of E is the set

$$E[m] := E(\bar{K})[m] = \{P \in E(\bar{K}) : [m]P = \mathcal{O}\}$$

- For the field extension $K \subset L$, then

$$E(L)[m] = \{P \in E(L) : [m]P = \mathcal{O}\}$$

  is the set of L-rational torsion points of E.
- If $p \nmid m$, then $E[m] \simeq \mathbb{Z}/m\mathbb{Z} \times \mathbb{Z}/m\mathbb{Z}$.

Divisors

- A divisor on E is a formal sum $D = \sum_P n_P(P)$ (almost all $n_P = 0$) where $P \in E(\overline{\mathbb{F}}_q)$.

  Examples:

$$\begin{array}{lcl} D_1 & = & (P_1)+2(P_2)+3(P_3)-10^{121}(P_4)+31(P_5) \\ D_2 & = & -7(P_1) \qquad\qquad -31(P_5)+11(Q_1)+(Q_2)-3(Q_3) \\ D_1+D_2 & = & -6(P_1)+2(P_2)+3(P_3)-10^{121}(P_4) \qquad +11(Q_1)+(Q_2)-3(Q_3), \end{array}$$

- The set of all divisors forms an abelian group $\mathrm{Div}(E)$.
- The degree of the divisor $D = \sum_P n_P(P)$ is the integer $\sum_P n_P$.
- For a function $f \in \overline{\mathbb{F}}_q(E)^*$ we associate the principal divisor $\mathrm{div}(f)$ defined by

$$\mathrm{div}(f) = \sum_P v_P(f)(P).$$

Divisors

- A divisor on E is a formal sum $D = \sum_P n_P(P)$ (almost all $n_P = 0$) where $P \in E(\overline{\mathbb{F}}_q)$.
  Examples:

$$\begin{aligned} D_1 &= (P_1)+2(P_2)+3(P_3)-10^{121}(P_4)+31(P_5) \\ D_2 &= -7(P_1) \qquad\qquad -31(P_5)+11(Q_1)+(Q_2)-3(Q_3) \\ D_1+D_2 &= -6(P_1)+2(P_2)+3(P_3)-10^{121}(P_4) \qquad +11(Q_1)+(Q_2)-3(Q_3), \end{aligned}$$

- The set of all divisors forms an abelian group $\mathrm{Div}(E)$.
- The degree of the divisor $D = \sum_P n_P(P)$ is the integer $\sum_P n_P$.
- For a function $f \in \overline{\mathbb{F}}_q(E)^*$ we associate the principal divisor $\mathrm{div}(f)$ defined by

$$\mathrm{div}(f) = \sum_P v_P(f)(P).$$

Divisors

- A divisor on E is a formal sum $D = \sum_P n_P(P)$ (almost all $n_P = 0$) where $P \in E(\overline{\mathbb{F}}_q)$.
  Examples:

$$\begin{aligned} D_1 &= (P_1)+2(P_2)+3(P_3)-10^{121}(P_4)+31(P_5) \\ D_2 &= -7(P_1) \qquad\qquad -31(P_5)+11(Q_1)+(Q_2)-3(Q_3) \\ D_1+D_2 &= -6(P_1)+2(P_2)+3(P_3)-10^{121}(P_4) \qquad +11(Q_1)+(Q_2)-3(Q_3), \end{aligned}$$

- The set of all divisors forms an abelian group $\mathrm{Div}(E)$.
- The degree of the divisor $D = \sum_P n_P(P)$ is the integer $\sum_P n_P$.
- For a function $f \in \overline{\mathbb{F}}_q(E)^*$ we associate the principal divisor $\mathrm{div}(f)$ defined by

$$\mathrm{div}(f) = \sum_P v_P(f)(P).$$

Divisors

- A divisor on E is a formal sum $D = \sum_P n_P(P)$ (almost all $n_P = 0$) where $P \in E(\overline{\mathbb{F}}_q)$.
  Examples:

$$\begin{array}{lcl} D_1 & = & (P_1)+2(P_2)+3(P_3)-10^{121}(P_4)+31(P_5) \\ D_2 & = & -7(P_1) \qquad\qquad -31(P_5)+11(Q_1)+(Q_2)-3(Q_3) \\ D_1+D_2 & = & -6(P_1)+2(P_2)+3(P_3)-10^{121}(P_4) \qquad +11(Q_1)+(Q_2)-3(Q_3), \end{array}$$

- The set of all divisors forms an abelian group $\mathrm{Div}(E)$.
- The degree of the divisor $D = \sum_P n_P(P)$ is the integer $\sum_P n_P$.
- For a function $f \in \overline{\mathbb{F}}_q(E)^*$ we associate the principal divisor $\mathrm{div}(f)$ defined by

$$\mathrm{div}(f) = \sum_P v_P(f)(P).$$

Divisors

- A divisor on E is a formal sum $D = \sum_P n_P(P)$ (almost all $n_P = 0$) where $P \in E(\overline{\mathbb{F}}_q)$.
  Examples:

$$\begin{array}{lcl} D_1 & = & (P_1)+2(P_2)+3(P_3)-10^{121}(P_4)+31(P_5) \\ D_2 & = & -7(P_1) \qquad\qquad -31(P_5)+11(Q_1)+(Q_2)-3(Q_3) \\ D_1+D_2 & = & -6(P_1)+2(P_2)+3(P_3)-10^{121}(P_4) \qquad +11(Q_1)+(Q_2)-3(Q_3), \end{array}$$

- The set of all divisors forms an abelian group $\mathrm{Div}(E)$.
- The degree of the divisor $D = \sum_P n_P(P)$ is the integer $\sum_P n_P$.
- For a function $f \in \overline{\mathbb{F}}_q(E)^*$ we associate the principal divisor $\mathrm{div}(f)$ defined by

$$\mathrm{div}(f) = \sum_P v_P(f)(P).$$

Divisors

- We define the sum of a divisor as

$$\mathrm{Sum}(D) = \mathrm{Sum}(\sum_{P \in E} n_P(P)) = \sum_{P \in E} [n_P]P \in E(\overline{F_q}).$$

- A divisor $D = \sum_{P \in E} n_P(P)$ on E is the divisor of a rational function on E if and only if

$$\deg(D) = 0, \quad \mathrm{Sum}(D) = \mathcal{O}.$$

- In particular: if $[m]P = \mathcal{O}$, so there exists a function $f_{m,P}$ s.t.

$$\mathrm{div}(f_{m,P}) = m(P) - m(\mathcal{O}).$$

Weil pairing

Let $P, Q \in E[\ell]$, and let $f_{\ell,P}$ and $f_{\ell,Q}$ be rational functions on E satisfying

$$\mathrm{div}(f_{\ell,P}) = \ell(P) - \ell(O)$$

and

$$\mathrm{div}(f_{\ell,Q}) = \ell(Q) - \ell(O).$$

The Weil pairing of P and Q (with respect to ℓ) is the quantity

$$e_\ell(P,Q) = \frac{f_{\ell,P}(Q+S)}{f_{\ell,P}(S)} \Big/ \frac{f_{\ell,Q}(P-S)}{f_{\ell,Q}(-S)}$$

where S is any point satisfying $S \notin \{O, P, -Q, P-Q\}$.
Remark: e_ℓ is well defined, i.e. it doesn't depend on the point S.

Properties of the Weil pairing

- $e_\ell(P,Q)^\ell = 1$.
- e_ℓ is bilinear.
- e_ℓ is alternating, i.e. $e_\ell(P,P) = 1$ for all $P \in E[\ell]$.
- e_ℓ is nondegenerate, i.e. if $e_\ell(P,Q) = 1$ for all $Q \in E[\ell]$, then $P = \mathcal{O}$.
- For a basis $\{P_1, P_2\}$ of $E[\ell]$. Any $P \in E[\ell]$ can be written as

$$P = [a_P]P_1 + [b_P]P_2$$

with $a_P, b_P \in \mathbb{Z}/\ell\mathbb{Z}$.
In this case, we have

$$e_\ell(P,Q) = e_\ell(P_1,P_2)^{a_P b_Q - a_Q b_P}.$$

Miller's algorithm

To compute Weil pairing we need to know the functions $f_{\ell,P}$ with divisors $\mathrm{div}(f_{\ell,P}) = \ell(P) - \ell(O)$.

- Let $f_{i,P}$, $i \in \mathbb{Z}$, be a function on E with

$$\mathrm{div}(f_{i,P}) = i(P) - ([i]P) - (i-1)(O).$$

  $f_{i,P}$ is called a Miller function.

- The special case $i = \ell$ leads to

$$\mathrm{div}(f_{\ell,P}) = \ell(P) - ([\ell]P) - (\ell-1)(O) = \ell(P) - \ell(O),$$

  since $[\ell]P = O$.

Miller's algorithm $\cdots$

Can we compute $f_{i+j,P}$ from $f_{i,P}$ and $f_{j,P}$?

- Compute the divisor of the product

$$\begin{aligned}
\operatorname{div}(f_{i,P} \cdot f_{j,P}) &= i(P) - ([i]P) - (i-1)(O) \\
&\quad + j(P) - ([j]P) - (j-1)(O) \\
&= (i+j)(P) - ([i]P) - ([j]P) - (i+j-2)(O) \\
&= (i+j)(P) - ([i+j]P) - (i+j-1)(O) \\
&\quad + ([i+j](P)) - ([i]P) - ([j]P) + (O) \\
&= \operatorname{div}(f_{i+j,P}) + ([i+j]P) - ([i]P) - ([j]P) + (O)
\end{aligned}$$

- The sum of the divisor is "almost" the divisor of $f_{i+j,P}$.

Miller's algorithm $\cdots$

For the lines occuring in the addition of $[i]P + [j]P = [i+j]P$:

- The first line l goes through $[i]P$, $[j]P$ and $-[i+j]P$, so

$$\mathrm{div}(l) = ([i]P) + ([j]P) + (-[i+j]P) - 3(\mathcal{O}).$$

- The second line v is a vertical line through $[i+j]P$ and $-[i+j]P$, so

$$\mathrm{div}(v) = ([i+j]P) + (-[i+j]P) - 2(\mathcal{O}).$$

- It follows

$$\mathrm{div}(\frac{l}{v}) = \mathrm{div}(l) - \mathrm{div}(v) = ([i]P) + ([j]P) - ([i+j]P) - (\mathcal{O}).$$

Miller's algorithm $\cdots$

- We already know

$$\mathrm{div}(f_{i,P} \cdot f_{j,P}) = \mathrm{div}(f_{i+j,P}) + ([i+j]P) - ([i]P) - ([j]P) + (\mathcal{O})$$

- as well as

$$\mathrm{div}(l) - \mathrm{div}(v) = ([i]P) + ([j]P) - ([i+j]P) - (\mathcal{O}).$$

- In particular : $\mathrm{div}(f_{i+j,P}) = \mathrm{div}(f_{i,P} \cdot f_{j,P}) + \mathrm{div}(l) - \mathrm{div}(v)$.
- which gives the Miller's formula

$$f_{i+j,P} = f_{i,P} \cdot f_{j,P} \cdot \frac{l}{v}$$

- We can choose normalized functions, i.e. $f_{1,P} = 1$.

Efficient pairing (Miller's algorithm)

- Doubling step:

$$f_{2i,P} = f_{i,P}^2 \cdot \frac{l_{[i]P,[i]P}}{v_{[2i]P}}.$$

- Adding step:

$$f_{i+1,P} = f_{i,P} \cdot f_{1,P} \cdot \frac{l_{[i]P,P}}{v_{[i+1]P}}.$$

- $l_{R,S}$ is the line passing through P and R (tangent if $P = R$), and v_R is the vertical line through R.

Efficient pairing (Miller's algorithm)

Miller's algorithm

IN: $P \in E[\ell]$, $Q \in E(\overline{\mathbb{F}_p})$ and $\ell = (\ell_m, \cdots, \ell_0)_2$.
OUT: $f_{\ell,P}(Q)$.

- $R \leftarrow P, f \leftarrow 1$
- for $i = m-1$ down to 0 do
  - $f \leftarrow f^2 \frac{l_{R,R}(Q)}{v_{[2]R}(Q)}$
  - $R \leftarrow [2]R$
  - if $\ell_i = 1$ then
    - $f \leftarrow f \frac{l_{R,P}(Q)}{v_{R+P}(Q)}$
    - $R \leftarrow R + P$
  - end if
- end for
- return f

Embedding degree

Let $E/\mathbb{F}_p$ an elliptic curve and $\ell \geq 1$ s.t. $p \nmid \ell$. The embedding degree of E with respect to ℓ is the smallest value of k s.t.

$$E(\mathbb{F}_{p^k})[\ell] \simeq \mathbb{Z}/\ell\mathbb{Z} \times \mathbb{Z}/\ell\mathbb{Z}.$$

Assume $\ell \neq p$ is a prime s.t. there is a point of $E(\mathbb{F}_p)$ of order ℓ. Then the embedding degree of E with respect to ℓ is given by one of the following cases:

- The embedding degree of E is one. (This cannot happen if $\ell > \sqrt{p}+1$ - exercise).
- $p \equiv 1 \pmod{\ell}$ and the embedding degree is ℓ.
- $p \not\equiv 1 \pmod{\ell}$ and the embedding degree is the smallest value of $k \geq 2$ s.t.

$$p^k \equiv 1 \pmod{\ell}.$$

MOV algorithm

Let $E/\mathbb{F}_p$ an elliptic curve. Let $P \in E(\mathbb{F}_p)$ a point of order ℓ, where ℓ is a large prime. Let k be the embedding degree of E with respect to ℓ and $Q \in \langle P \rangle$.

1. Compute $N = \#E(\mathbb{F}_{p^k})$.
2. Choose $T \in_R E(\mathbb{F}_{p^k})$ with $T \notin E(\mathbb{F}_p)$.
3. Compute $T' = [N/\ell]T$. If $T' = \mathcal{O}$, GOTO 2. Otherwise T' is a point of order ℓ.
4. Compute the Weil pairing $\alpha = e_\ell(P, T') \in \mathbb{F}_{p^k}^*$ and $\beta = e_\ell(Q, T') \in \mathbb{F}_{p^k}^*$.
5. If k is not too large, solve the DLP $\alpha^n = \beta$ in $\mathbb{F}_{p^k}^*$.
6. Then, $Q = [n]P$.

MOV algorithm

Why MOV solve the ECDLP?

- T' is generally independent of $P \quad \Longrightarrow \{P, T'\}$ forms a basis of $E[\ell]$.
- The nondegeneracy of e_ℓ implies:

$$e_\ell(P, T')^r = 1 \quad \text{iff} \quad \ell \mid r$$

  i.e. $e_\ell(P, T')$ is a non-trivial ℓ^{th} root of unity in $\mathbb{F}_{p^k}^*$.
- If $Q = [j]P$, then

$$e_\ell(P, T')^n = e_\ell(Q, T') = e_\ell([j]P, T') = e_\ell(P, T')^j$$

  and thus $e_\ell(P, T')^{n-j} = 1$, i.e. $n \equiv j \pmod{\ell}$.

MOV algorithm

How practical is MOV?

- If k is large, say $k > (\ln p)^2$, then MOV is "infeasible". For example, if $p \sim 2^{160}$, then $k > 4000$.
- A randomly chosen elliptic curve $E/\mathbb{F}_p$ has "almost always" embedding degree much larger than $(\ln p)^2$, MOV is in general not useful.
- However, supersingular curves have embedding degree $k \leq 6$ (Menezes, Okamoto, Vanstone).
- For example, $y^2 = x^3 + x$ is supersingular for any prime $p \equiv 3 \pmod 4$ and it has embedding degree 2 for any $\ell > \sqrt{p} + 1$.

Distortion map

For applications in crypto, we need non-alternating pairing!

- The Weil pairing is alternating, i.e. $e_m(P,P) = 1$ for all $P \in E[m]$.
- If $P_1 = [a]P$ and $P_2 = [b]P$ then

$$e_m(P_1, P_2) = e_m([a]P, [b]P) = e_m(P,P)^{ab} = 1.$$

- If possible, find a "nice map" $\phi : E \longrightarrow E$ with the property that P and $\phi(P)$ are independent.
- Evaluate

$$e_m(P_1, \phi(P_2)) = e_m([a]P, [b]\phi(P)) = e_m(P, \phi(P))^{ab}.$$

Distortion map

Let $\ell \geq 3$ prime, $P \in E[\ell]$. A map $\phi : E \longrightarrow E$ is said to be an ℓ-distortion map for P if it has the following properties:

- $\phi([n]P) = [n]\phi(P)$ for all $n \geq 1$
- The number $e_\ell(P, \phi(P))$ is a primitive ℓ^{th} root of unity, i.e.

$$e_\ell(P, \phi(P))^r = 1 \quad \text{iff} \quad r \text{ is a multiple of } \ell.$$

- The modified Weil pairing $\hat{e}_\ell$ on $E[\ell]$ (relative to ϕ) is defined by

$$\hat{e}_\ell(Q, Q') = e_\ell(Q, \phi(Q')).$$

If Q and Q' are multiple of P then $\hat{e}_\ell(Q, Q') = 1$ iff $Q = \mathcal{O}$ or $Q' = \mathcal{O}$.

Example of distortion map

Example

Let

- prime $p \equiv 3 \pmod 4$
- $E : y^2 = x^3 + x$
- $\alpha \in \mathbb{F}_{p^2}$ s.t. $\alpha^2 = -1$
- prime $\ell \geq 3$ s.t. there is a point $P \in E(\mathbb{F}_p)[\ell]$.

Then $\phi : E \longrightarrow E$ with $\phi(x, y) = (-x, \alpha y)$ is an ℓ-distortion map for P.

- Take now $p = 547$ and take $\mathbb{F}_{p^2} = \{a + bi : a, b \in \mathbb{F}_p\}$, where $i^2 = -1$.
- $\#E(\mathbb{F}_{547}) = 548 = 2^2 \cdot 137$ and $P = (67, 481) \in E(\mathbb{F}_{547})[137]$.

Example of distortion map

- The distortion map gives $\phi(P) = (-67, 481i) \in E(\mathbb{F}_{547^2})$
- To compute $e_{137}(P, \phi(P))$ we choose a random point $S = (256 + 110i, 441 + 15i) \in E(\mathbb{F}_{547^2})$
- Miller's algorithm gives:

$$\frac{f_P(\phi(P) + S)}{f_P(S)} = \frac{376 + 138i}{384 + 76i} = 510 + 96i$$

$$\frac{f_{\phi(P)}(P - S)}{f_{\phi(P)}(-S)} = \frac{498 + 286i}{393 + 120i} = 451 + 37i$$

- Then

$$\hat{e}_{137}(P, P) = e_{137}(P, \phi(P)) = \frac{510 + 96i}{451 + 37i} = 37 + 452i \in \mathbb{F}_{547^2}$$

- Of course, $(37 + 452i)^{137} = 1$.

Thank you for your attention!