Admin:

Quizzes graded soon (tonight)

Project: meet with TAs this week

presentations start May 6th (two weeks)

Today:

Elliptic Curves

Bilinear Maps

BLS signatures

TOPIC:

DATE:

FILE UNDER:

PAGE:

Elliptic Curves

- finite group (analagous to $\mathbb{Z}_p^*$)
- why?
  - DL problem seems harder (exponential)
    Thus can use shorter reps
    (256 bits instead of 2048)
  - can (sometimes) define <u>bilinear maps</u>
    on them, which are very cool
- why not?
  - new math (?) (not really...)
  - computing size of group takes some work

$$|\mathbb{Z}_p^*| = p-1$$

$$|E_{ab}| \in [p+1-2\sqrt{p},\ p+1+2\sqrt{p}]$$

(Schoof's alg)

- Curve25519 uses $p = 2^{255}-19$
  base point $x=9$

$$y^2 = x^3 + 486662x^2 + x$$

$$\text{order}\langle x\rangle \approx 2^{252} + 27\ldots493 \text{ (prime)}$$

Massachusetts Institute of Technology
6.857: Network and Computer Security
Professors Ronald L. Rivest and Yael Tauman Kalai

Recitation 6
March 16, 2018
Akshay Degwekar

Recitation 6 : Elliptic Curves & Number Theory

We review elliptic curves, finite fields GF(2^k) and the extended Euclid's algorithm.

1 Elliptic Curves

We begin by defining Elliptic Curves.

Definition 1.1 (Elliptic Curve). *An* Elliptic Curve *over a field* $\mathbb{F}$ *is a curve given by an equation of the form:*

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

where $a, b \in \mathbb{F}$ *such that the* discriminant $\Delta = 4a^3 + 27b^2 \neq 0$, *that is, the polynomial* $x^3 + ax + b$ *has distinct roots.*

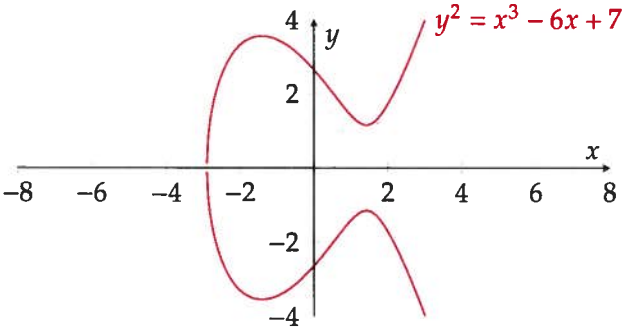


Figure 1: The Elliptic Curve defined by $y^2 = x^3 - 6x + 7$ over $\mathbb{R}$.

We want to define a group structure over the points on the elliptic curve. We do that next.

Definition 1.2. *The* Group *E defined by the elliptic curve* ($y^2 = x^3 + ax + b$) *over field* $\mathbb{F}$ *is defined as the set of points:*

$$E = \left\{(x, y) \in \mathbb{F}^2 \mid y^2 = x^3 + ax + b\right\} \cup \left\{\infty\right\},$$

with the identity element ∞ *and the group operation* $+$ *defined as follows:*

Let $P(x_1, y_1)$ *and* $Q(x_2, y_2)$ *be points in E. Then,*

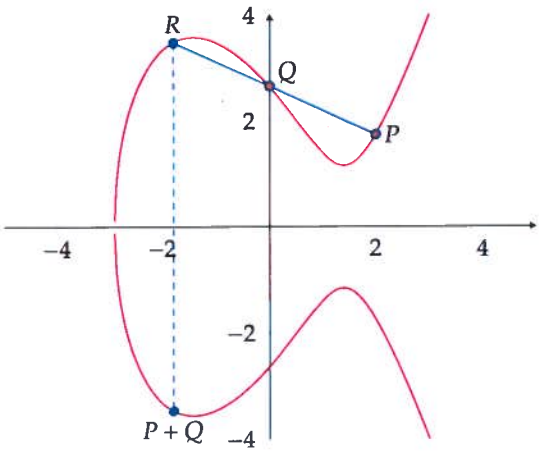
1. *(Identity)* $P + \infty = \infty + P = P$.
2. *(Vertical Line) If* $x_1 = x_2$ *and* $y_1 = -y_2$ *then* $P + Q = \infty$.
3. *(Vertical Tangent) If* $y_1 = 0$ *then* $P + P = \infty$.
4. *(Tangent)* $P + P = (x, y)$ *where* $\lambda = \dfrac{3x_1^2 + a}{2y_1}$, $x = \lambda^2 - 2x_1$, *and* $y = -(\lambda(x - x_1) + y_1)$.
5. *(General Case) Let* $x_1 \neq x_2$ *then* $P + Q = (x, y)$ *where* $\lambda = \dfrac{y_2 - y_1}{x_2 - x_1}$, $x = \lambda^2 - x_1 - x_2$ *and* $y = -(\lambda(x - x_1) + y_1)$.

Observe that the computation as described is independent of which field is used.

Theorem 1.3. *$(E, +)$ is a group.*

The identity, commutativity, inverse all follow from the definition. We will not prove that the operation is associative, but it is. We describe the geometric intuition behind these and the corresponding calculations next.

The General Case : $P + Q$



The line between (x_1, y_1) and (x_2, y_2) is given by

$$y = \frac{y_2 - y_1}{x_2 - x_1}(x - x_1) + y_1$$

where $\lambda = (y_2 - y_1)/(x_2 - x_1)$ is the slope and $\nu = y_1 - \lambda x_1$ is the intercept. So, to compute the point $R(x_3, y_3)$, we need to compute the intersection of the curve E with the line above. That is,

$$(\lambda x + \nu)^2 = x^3 + ax + b$$

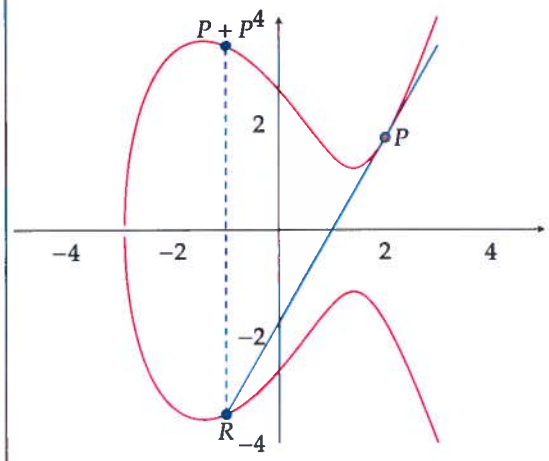
Simplifying, we get that,

$$x^3 - \lambda^2 x^2 + x(a - 2\lambda\nu) + (b + \nu^2)$$

We know two of the roots: x_1, x_2. To find the third, use the fact that the second term is the sum of roots.[a] Hence, $\lambda^2 = x_1 + x_2 + x_3$. Hence $x_3 = \lambda^2 - x_1 - x_2$. And $y_3 = \lambda(x_3 - x_1) + y_1$. Then the point $P + Q$ is $(x_3, -y_3)$.

[a] This follows from comparing $(x-x_1)(x-x_2)(x-x_3) = x^3 - (x_1 + x_2 + x_3)x^2 + \ldots$ with the equation above.

Tangents: $P + P$



The tangent at $y = f(x)$ has slope $f'(x)$ (the derivative). In this case, $y = \sqrt{x^3 + ax + b}$. Hence,

$$\lambda = f'(x) = \frac{(3x^2 + a)}{\sqrt{x^3 + ax + b}} = \frac{3x^2 + a}{2y}$$

So, the line through (x_1, y_1) is,

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

Here also, we need to find the intersection of the curve with the line, knowing that x_1 is a repeated root. So, we get $x_3 = \lambda^2 - 2x_1$ and $y_3 = (\lambda(x_3 - x_1) + y_1)$. Then the point $P + P$ is $(x_3, -y_3)$.

"Gap group" is one in which

- DDH is easy ("Decision Diffie Hellman")

  [Recall: given (g, g^a, g^b, g^c), to decide if $ab = c \pmod{\text{order}(g)}$]

but

- CDH is hard ("Computational Diffie Hellman")

  [Recall: given (g, g^a, g^b), to compute g^{ab}]

(Note that CDH easy $\Rightarrow$ DDH easy)

This difference in difficulty between DDH ("easy") and CDH ("hard") forms a "gap".

— How can one construct a "gap group"?

— What good would that be?

Bilinear maps

Suppose: G_1 is group of prime order q, with generator g

"shadow group" → G_2 is group of prime order q, with generator h

[we use multiplicative notation for both groups]

See Fig. (next page) → and there exists a (bilinear) map

$$e: G_1 \times G_1 \longrightarrow G_2$$

such that

$e(g,g) = h$

$$(\forall a,b)\ e(g^a, g^b) = h^{ab}$$!!!

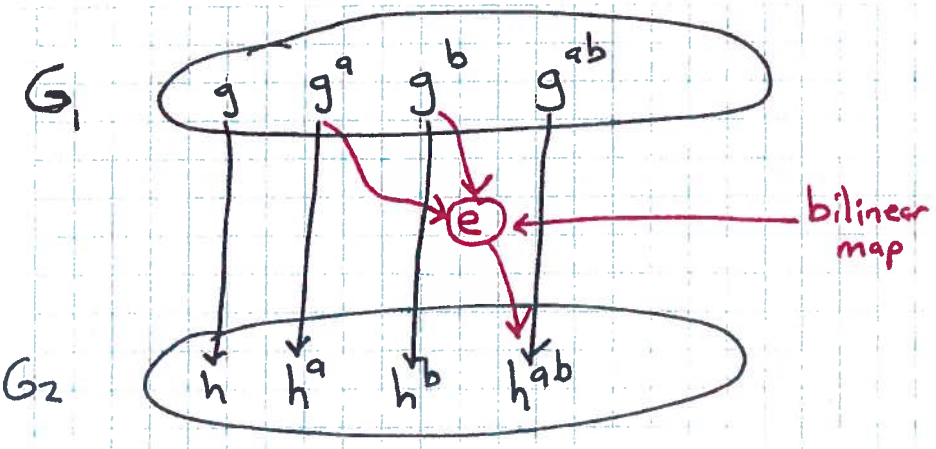
$$= e(g, g^{ab})$$
$$= e(g,g)^{ab}$$
$$= e(g, g^b)^a$$
$$= e(g, g^a)^b$$
$$= e(g^b, g^a)$$
$$\ldots$$

Bilinear maps also called "pairing functions"

They have an enormous number of applications. *

We are, of course, interested in efficiently computable bilinear maps.

* google: "The pairing-based crypto lounge"



$|G_1| = |G_2| = q$ (prime)

g generates G_1

h generates G_2

CDH hard in G_1 & in G_2

DDH easy in G_1 (using e)

"$e(g^a, g) = h^a$ computes "shadows""

Note: If discrete log was easy in G_2 then it would be easy in G_1.

$$DL_{G_1, g}(g^a) = DL_{G_2, h}(h^a) = a$$

Theorem:

If there is a bilinear map

$$e: G_1 \times G_1 \to G_2$$

between two groups of prime order q,

then DDH is easy in G_1

Proof:

Given (g, g^a, g^b, g^c) (elements of G_1)

then

$$c = ab \pmod q \iff \underbrace{e(g^a, g^b) = e(g, g^c)}_{\equiv}$$

$$\underbrace{h^{ab} = h^c}_{\equiv}$$

$$ab = c \pmod q$$

So: accept (g, g^a, g^b, g^c) iff $e(g^a, g^b) = e(g, g^c)$. ▨

Even though DDH is easy in G_1, CDH may still be hard; we may have a "gap group".

How to construct gap groups (with bilinear maps):

- This is not simple! We give just a sketch.
- G_1 will be "supersingular" elliptic curve

  e.g. elliptic curve defined by points on

  $$y^2 = x^3 + ax + b \pmod p$$

  where $p = 2 \pmod 3$, $p \geq 5$

  $a = 0$

  $b \in Z_p^*$ (can choose $b=1$)

- G_2 is finite field $\mathbb{F}_{p^k}$ for some small k

  (can use subgroups of G_1 & G_2 by choosing generators of order $\approx 2^{160}$ say...)

- e (bilinear map) is implemented as a "Weil pairing" or a "Tate pairing".

Application 1:

Digital signatures

(Boneh, Lynn, Shacham (2001))

Signatures are short (e.g. 160 bits)!

Public: groups G_1, G_2 of prime order q

pairing function $e: G_1 \times G_1 \to G_2$

g = generator of G_1

H = hash fn (C.R.) from messages to G_1

Note use of multiplicative notation here

Secret key: x where $0 < x < q$

Public key: $y = g^x$ (in G_1)

To sign message M:

Let $m = H(M)$ (in G_1)

Output $\sigma = \sigma_x(M) = m^x$ (in G_1)

Note: Signature may be short! Just one element of G_1.

To represent point on elliptic curve, just give x, then one more bit to say which y is wanted (there are only two square roots of y^2: $\pm y$)

To verify (y, M, σ):

Check $e(g, \sigma) \stackrel{?}{=} e(y, m)$ where $m = H(M)$

$e(g,m)^x$ in both cases

Theorem: BLS signature scheme secure against existential forgery under chosen message attack in ROM assuming CDH is hard in G_1.

Application 2:

Three-way key agreement (Joux, generalizing DH)

Recall DH: $A \to B : g^a$
$B \to A : g^b$
key $= g^{ab}$

Joux: Suppose G_1 has generator g
Suppose $e : G_1 \times G_2$ is a bilinear map.

$A \to B, C : g^a$

$B \to A, C : g^b$

$C \to A, B : g^c$

A computes $e(g^b, g^c)^a = e(g,g)^{abc}$ ←
B computes $e(g^a, g^c)^b = e(g,g)^{abc}$ ← $= !$
C computes $e(g^a, g^b)^c = e(g,g)^{abc}$ ←

key $= e(g,g)^{abc}$

Bilinear Diffie-Hellman problem

Secure assuming "BDH" ≡

given g, g^a, g^b, g^c, e
hard to compute $e(g,g)^{abc}$

Four-way key agreement is open problem!

(multilinear maps!)

Application 3:

Identity-based encryption (IBE) [Boneh, Franklin '01]

TTP (trusted third party) publishes

G_1, G_2, e (bilinear map), g (generator of G_1), y

where $y = g^s$ & s is TTP's master secret.

Let H_1 be random oracle mapping names (e.g. "alice@mit.edu") to elements of G_1^*

Let H_2 be random oracle mapping G_2 to $\{0,1\}^*$ (PRG).

Want to enable anyone to encrypt message for Alice knowing only TTP public parameters & Alice's name

Encrypt(y, name, M):

$r \xleftarrow{R} Z_q^*$ (here prime $q = |G_1| = |G_2|$)

$g_A = e(Q_A, y)$ where $Q_A = H_1(\text{name})$

output $(g^r, M \oplus H_2(g_A^r))$

Decrypt ciphertext $c = (u, v)$:

- Alice obtains $d_A = Q_A^s$ from TTP (once is enough)

  where $Q_A = H_1(\text{name})$.

  This is Alice's decryption key.

  Note that TTP also knows it!

  Note that message may be encrypted before Alice gets d_A.

- Compute $v \oplus H_2(e(d_A, u))$

$$= v \oplus H_2(e(Q_A^s, g^r))$$
$$= v \oplus H_2(e(Q_A, g)^{rs})$$
$$= v \oplus H_2(e(Q_A, g^s)^r)$$
$$= v \oplus H_2(e(Q_A, y)^r)$$
$$= v \oplus H_2(g_A^r)$$
$$= M$$

Security: Semantically secure in ROM assuming BDH.

ID-based signature (Hess 2002; Dutta survey §4.10)

note use of additive notation

master secret = s
master public = $P_{pub} = sP$ (P generates G_1)

$H_1 : \{0,1\}^* \to G_1$

$H : \{0,1\}^* \times G_2 \to \mathbb{Z}_q^*$

Extract: user gives ID. Public id $= H_1(ID) = Q_{ID}$
Secret key $= s \cdot Q_{ID} = S_{ID}$

Sign (S_{ID}, m): $P_1 \in_R G_1^*$

$k \in_R \mathbb{Z}_q^*$

$r = e(P_1, P)^k$

$v = H(m, r)$

$u = v S_{ID} + k P_1$ } = signature

Verify: $(Q_{ID}, m, (u,v))$:

$r = e(u, P) \cdot e(Q_{ID}, -P_{pub})^v$

accept if $v = H(m, r)$

Secure against existential forgery in ROM under adaptive chosen message attack assuming weak-DH problem is hard.

given (P, Q, sP) for $P, Q \in G_1$
output sQ