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.*

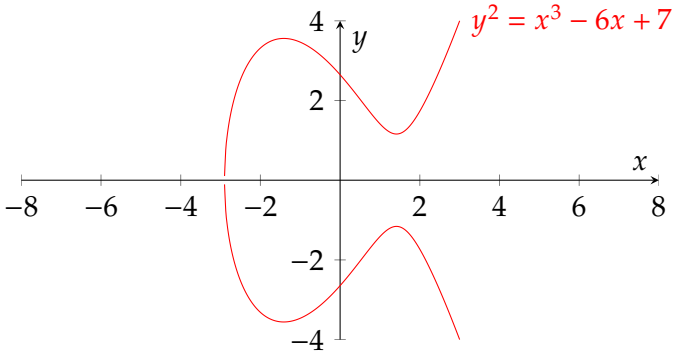


Figure 1: The Elliptic Curve defined by $y^2 = x^3 - 6x + 7$ over $\mathbb{R}$.

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,*

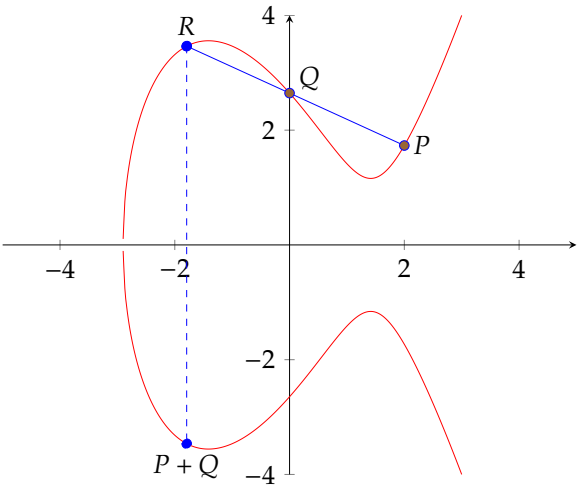
1. *(Identity)* $P + \infty = \infty + P = P$.
2. *(Vertical Line) If* $x_1 = x_2$ *and* $y_1 = -y_2$ *then* $P + Q = \infty$.
3. *(Vertical Tangent) If* $y_1 = 0$ *then* $P + P = \infty$.
4. *(Tangent)* $P + P = (x, y)$ *where* $\lambda = \dfrac{3x_1^2 + a}{2y_1}$, $x = \lambda^2 - 2x_1$, *and* $y =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

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

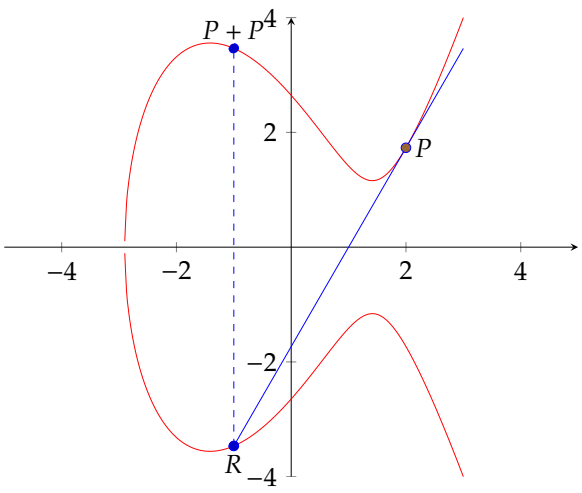
Simplifying, we get that,

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

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 $(x_3, -y_3)$.

2 Finite Fields

We recall the definition of a finite field.

Definition 2.1 (Field)**.** *A tuple* $(F, +, \cdot)$ *is a field if the following properties are satisfied:*

1. $(F, +)$ *is a commutative group. That is,*
   - *(a)* Closure. *If* $a, b \in F$ *then* $a + b \in F$.
   - *(b)* Associativity. *For all* $a, b, c \in F$, $(a + b) + c = a + (b + c)$.
   - *(c)* Identity. *There is an identity element* $0 \in F$ *such that* $0 + a = a + 0 = a$ *for all* $a \in F$.
   - *(d)* Inverse. *For all elements* $a \in F$, *there exists* $-a \in F$ *such that* $a + (-a) = -a + a = 0$.
   - *(e)* Commutativity. $a + b = b + a$ *for all* $a, b \in F$.
2. $(F \setminus \{0\}, \cdot)$ *is a commutative group. The identity element is called* 1.
3. Distributivity. *For all* $a, b, c \in F$, $(a + b) \cdot c = a \cdot c + b \cdot c$.

Examples of fields include rational numbers $\mathbb{Q}$, real numbers $\mathbb{R}$. Integers $\mathbb{Z}$ are not a field because they do not have multiplicative inverses for non-zero elements.

Theorem 2.2. *Every finite field has size* p^k *for prime* p *and positive integer* k. *There exists a unique finite field of size* p^k *for all primes* p *and positive integers* k.

We will not show this. We will however describe a construction of finite fields of size 2^k.

Definition 2.3 (Irreducible Polynomial)**.** *A polynomial* $f(x)$ *over a field* $\mathbb{F}$ *is* irreducible *if and only if there do not exist polynomials* g_1, g_2 *such that* $\deg(g_1) < \deg(f)$ *and* $\deg(g_2) < \deg(f)$ *and* $f(x) = g_1(x) \cdot g_2(x)$.

Let $f(x)$ be an irreducible polynomial of degree k over GF(2). To give some examples: $x^2 + 1 = (x + 1)(x + 1)$. While $x^2 + x + 1$ is irreducible.

Theorem 2.4. *Let* $f(x)$ *be an irreducible polynomial of degree* k *over* GF(2). *Then* $\mathrm{GF}(2)[x]/(f)$ *is a field where* $\mathrm{GF}(2)[x]$ *is the set of all polynomials over* GF(2).

Example 2.5. $\mathbb{F}_{2^2} = \{0, 1, x, x + 1\}$ with irreducible polynomial $x^2 + x + 1$. Addition is to simply add the polynomials over GF(2). And to multiply, first multiply the two polynomials and then compute the remainder modulo $f(x) = x^2 + x + 1$. e.g., $x(x + 1) = x^2 + x = 1$ after reducing mod f. And $(x + 1)(x + 1) = x^2 + 2x + 1 = x^2 + 1 = x$. To find the inverse, use extended Euclid's algorithm for polynomials. This also enables division.

Similarly we can construct $\mathrm{GF}(2^8)$ used in AES by using the irreducible polynomial $f(x) = x^8 + x^4 + x^3 + x + 1$.

3 The Extended Euclid's algorithm

The extended Euclid's algorithm computes not only the gcd, but also a witness x, y such that $ax + by = \gcd(a, b)$.

This algorithm is efficient because every two recursions, the size of the inputs decreases by one bit. And hence the algorithm terminates in poly log depth. The proof of correctness is an argument by strong induction. Observe that if the recursive call returned the correct value, then the current invocation would also return a witness.

```
def Euclid(a, b):
  if b == 0:
    return a
  return Euclid(b, a % b)
```

```
def ExtEuclid(a, b):
  if b == 0:
    # As gcd(a,0) = a = a*1 + 0*0.
    return (a, 1, 0)
  (d, x1, y1) = ExtEuclid(b, a % b)
  # As d = b*x1 + (a%b)*y1 and
  # a = b*(a//b) + (a%b).
  return (d, y1, x1 - (a//b)*y1)
```

```
gcd(7,5)
  gcd(5,2)
    gcd(2,1)
      gcd(1,0)
      out (1,1,0)
    out (1,0,1)
  out (1,1,-2)
out (1,-2,3)
```

Figure 2: Euclid's Algorithm and Extended Euclid's Algorithm for non-negative inputs.