Today in Cryptography (5830)

Elliptic curve cryptography

Based in part on slides from Martin R. Albrecht and Kenny Paterson

Asymmetric crypto so far

- RSA
  - Work in $\mathbf{Z}_N^*$ for large composite N = pq
  - RSA assumption: given X^e mod N can't recover X without secret key d
- Discrete log problem (DLP)
  - Worked in prime-order subgroup of $\mathbf{Z}_p^*$ for prime p. g is a generator of subgroup. q is size of subgroup
  - Discrete log assumption: given g^x mod p can't recover x

Comparison

Security level	RSA (log N)	DLP in finite field (log p)	DLP subgroup size (log q)
80	1024	1024	160
112	2048	2048	224
128	3072	3072	256
256	15360	15360	512

Exponentiation time performance scales with $O(n^3)$ for bit length n numbers

Numerical estimates from best attack times using best known algorithms runtimes

Finite fields

- Finite field is a finite set with basic operations:
  - Addition, subtraction, multiplication, division
- Integers modulo prime p is a field.
  - Notated F_p or GF(p) .
  - The set is {0,1,…,p-1}
    - Addition is a + b mod p
    - Multiplication is ab mod p .
- We've been using this implicitly, just making it explicit. We use them for ***elliptic curves***

Elliptic curves

- Are discrete log based systems. They use a new kind of group defined relative to a finite field. We will only need curves over $\mathbf{F}_p$
- Independently suggested for cryptographic applications by Victor Miller and Neal Koblitz in 1985
- They are now the go-to state-of-art in practice

Comparison

Security level	RSA size (log N)	DLP in finite field (log p)	DLP subgroup size (log q)	ECC group size (log q)
80	1024	1024	160	160
112	2048	2048	224	224
128	3072	3072	256	256
256	15360	15360	512	512

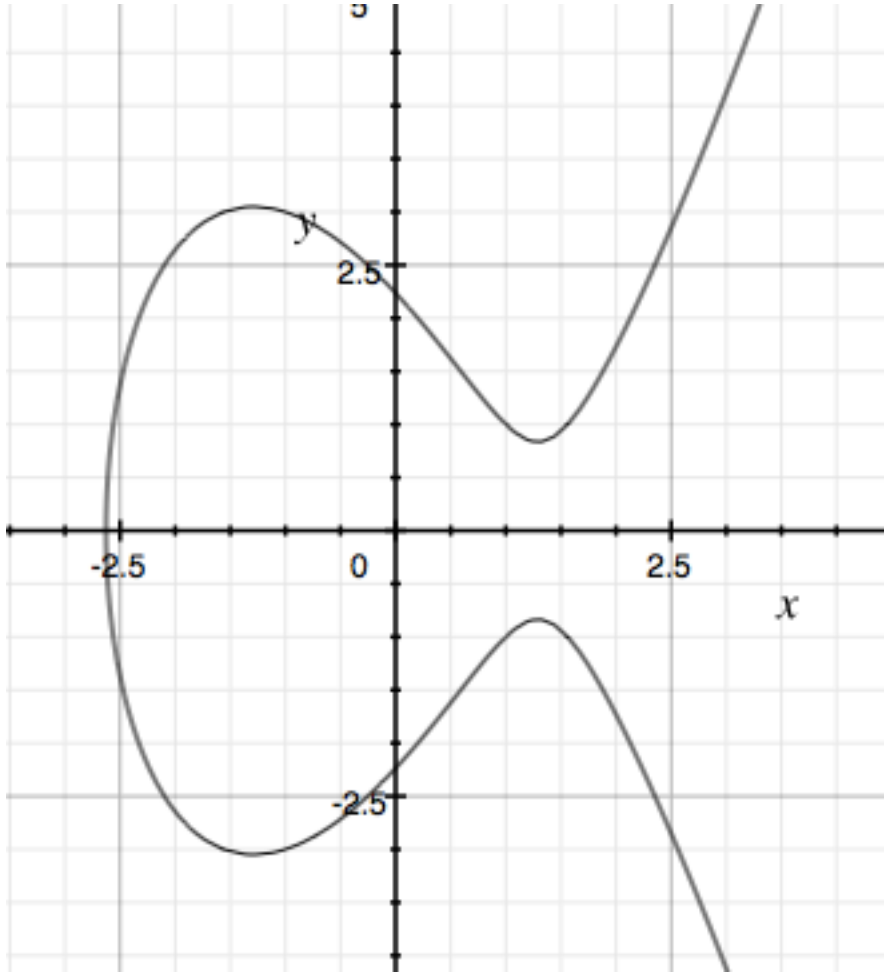
ECC has smallest representations and fastest performance of all asymptotic primitives we will see

Elliptic curves

- An elliptic curve is set of x,y points in $\mathbf{F}_p$ defined by an equation

$$E = \{(x, y) \mid y^2 = x^3 + ax + b \bmod p\}$$

  a,b are fixed values also from $\mathbf{F}_p$
- Plus one special point $\mathcal{O}$ called the "point at infinity"
- Technical condition: $4a^3 + 27b^2 \neq 0$
  - Otherwise curve is not non-singular



$y^2 = x^3 - 5x + 5$ (over the reals)

Elliptic curves

- Example: $y^2 = x^3 + 2x + 4 \mod 5$ (i.e., over $\mathbf{F}_5$)

x	0	1	2	3	4
x^3	0	1	3	2	4
$2x$	0	2	4	1	3
4	4	4	4	4	4
y^2	4	2	1	2	1
y	2, 3		1, 4		1, 4

E has 7 points: $O, (0,2), (0,3), (2,1), (2,4), (4,1), (4,4)$

Elliptic curves as groups

Recall that a *group* is a set G along with an operation * such that (s.t.) for all a,b,c in G:

- *Closure*: a*b in G
- *Associativity*: (a*b)*c = a*(b*c)
- *Identity*: exists 1 in G s.t. 1*a = a*1 = a
- *Inverses*:

  for all a in G, there exists a^{-1} in G s.t. $a*a^{-1} = 1$

Abelian groups: a*b = b*a

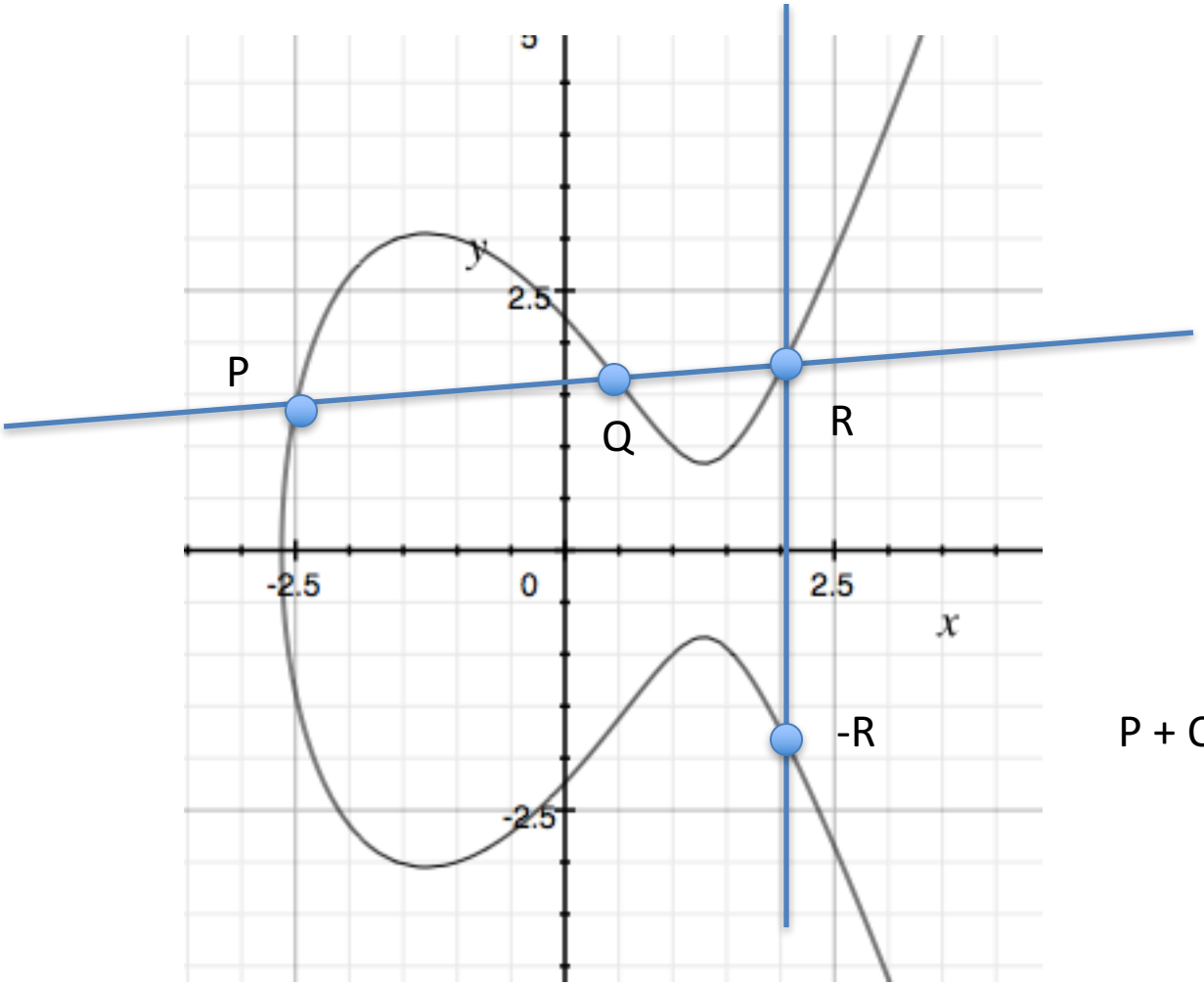
Elliptic curve group operation

- The operation * is called “point addition” and we usually denote this P + Q for P,Q in E

- What does it mean to “add” two points

P = (x1,y1) and Q = (x2,y2)?

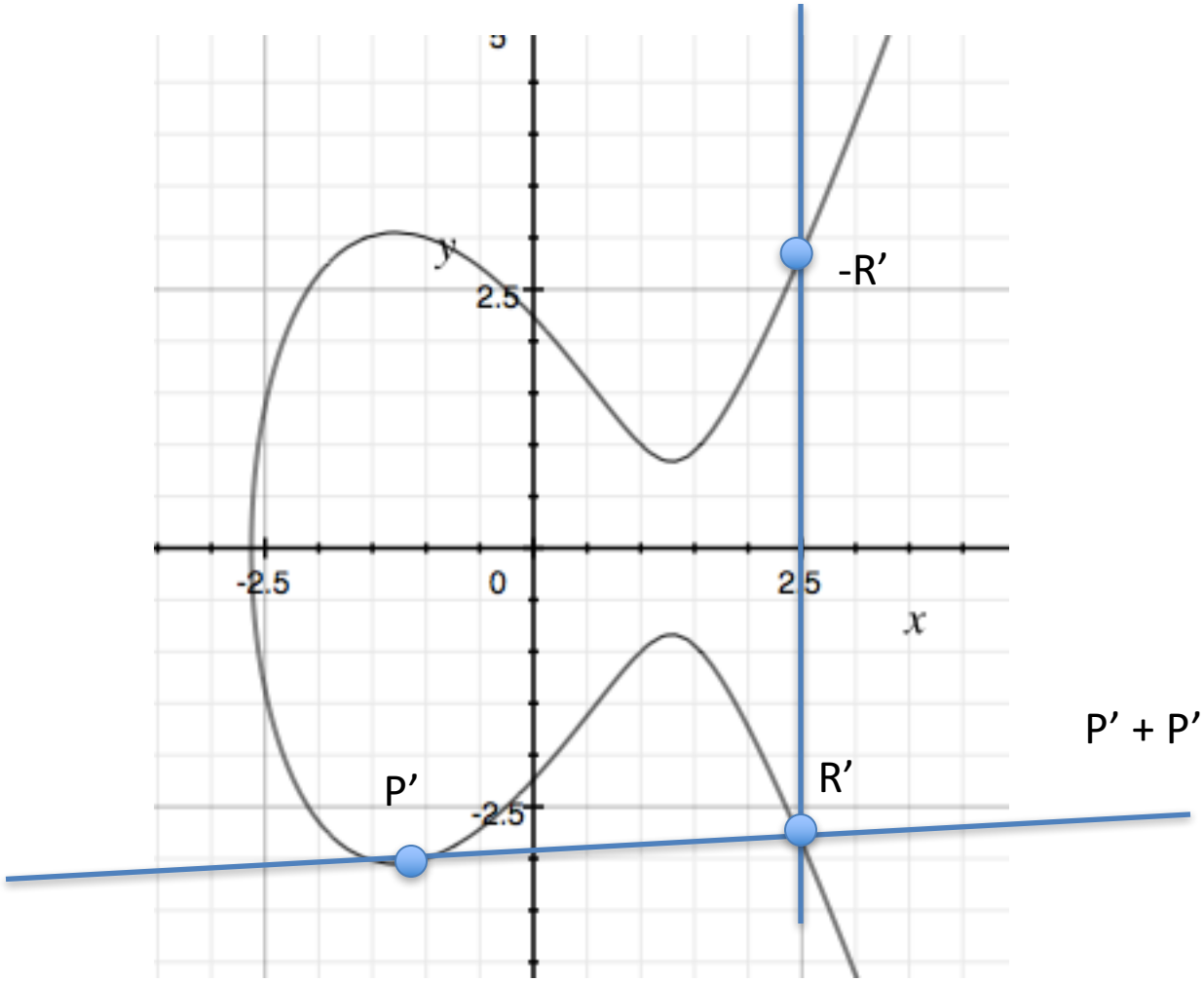
Adding two points P = (x1,y1) and Q = (x2,y2) with x1 ≠ x2



P + Q = -R

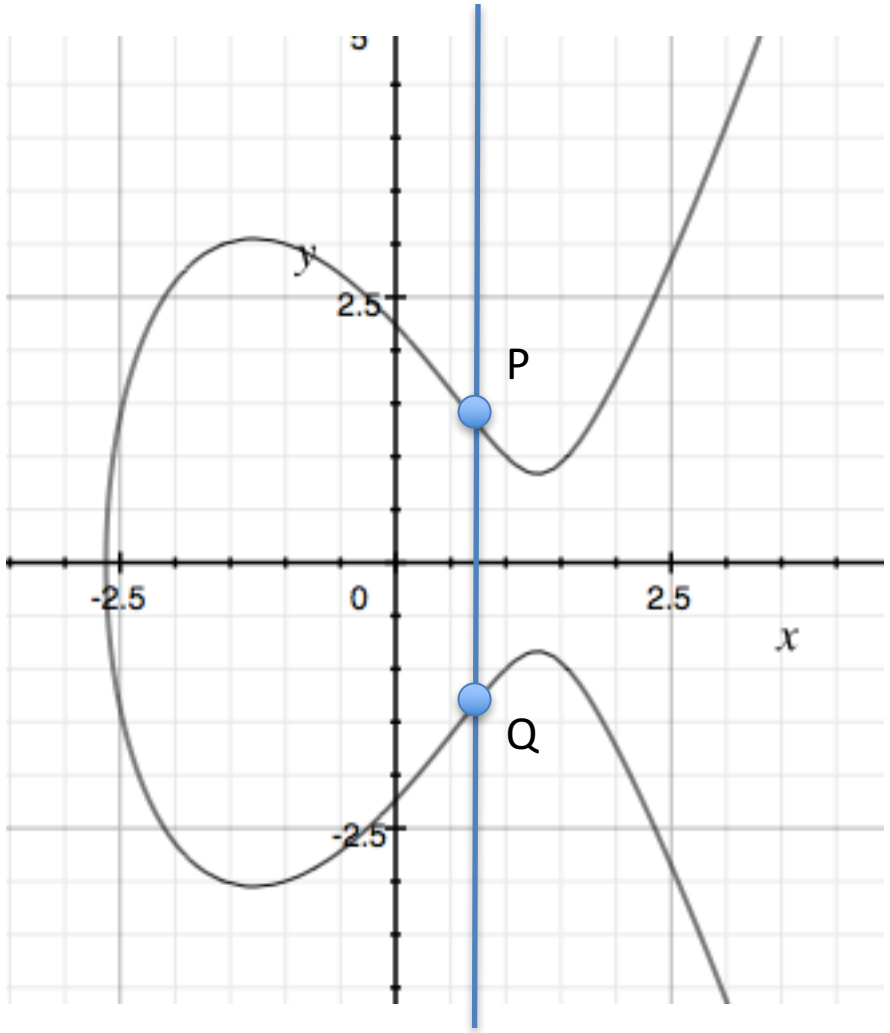
$y^2 = x^3 - 5x + 5$

Adding P’ = (x1,y1) to itself



$y^2 = x^3 - 5x + 5$

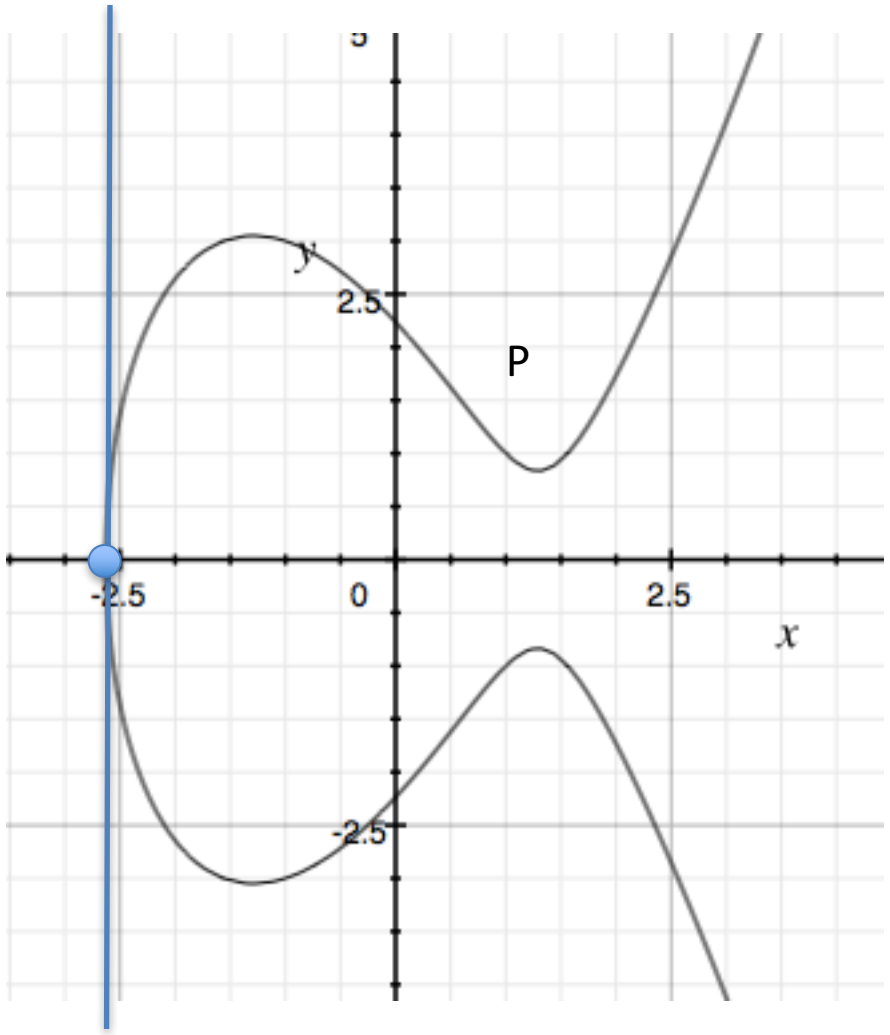
Adding two points P = (x1,y1) and Q = (x2,y2) with x1 = x2



P + Q = $\mathcal{O}$

$y^2 = x^3 - 5x + 5$

Adding two points P = (x1,y1) and Q = (x2,y2) with x1 = x2, y1 = y2 = 0



P + P = $\mathcal{O}$

$y^2 = x^3 - 5x + 5$

Elliptic curve group operation

- Let $\mathcal{O}$ be the identity for the group
  - P+$\mathcal{O}$ = $\mathcal{O}$ + P = P for any point
- Let P^{-1} be (x,-y) for P = (x,y)
  - By definition $P + P^{-1} = \mathcal{O}$
  - $\mathcal{O}^{-1} = \mathcal{O}$
- P + Q has geometric interpretation
  - P + Q + R = $\mathcal{O}$
  - P’ + P’ = -R’
- Over finite fields same, but calculate algebraically

Elliptic curve group operation (over F_p)

- Adding P = (x1,y1) , Q = (x2,y2):
  - Slope is D = (y2-y1) / (x2-x1) mod p
- Line through P,Q is

  $y = D(x-x1) + y1 \mod p$
- Substitute in to curve equation:

  $(D(x-x1) + y1)^2 = x^3 + ax + b \mod p$
- Only three values of x satisfy equation: x1,x2 and
  - $x3 = D^2 - x1 - x2 \mod p$
  - So: $y3 = D(x3-x1) + y1 \mod p$
- Above only works for x1 ≠ x2. P = Q case handled similarly to compute slope of tangent

Elliptic curve group operation (over F_p)

P + Q for P = (x1,y1) Q = (x2,y2)

If P = $\mathcal{O}$ then return Q

If Q = $\mathcal{O}$ then return P

If x1 = x2 and y1 = -y2 then return $\mathcal{O}$

If x1 = x2 and y1 = y2 = 0 then return $\mathcal{O}$

If x1 = x2 then D = $(3x1^2 + a)/(2y1) \bmod p$

Else D = $(y2-y1) / (x2-x1) \bmod p$

$x3 = D^2 - x1 - x2 \bmod p$

$y3 = D(x2-x1) + y1 \bmod p$

Return (x3,-y3)

Elliptic curve group operation (over F_p)

- Amazingly, point addition is a group operation
  - *Closure*: P + Q on curve for all P, Q
  - *Associativity*: P + (Q + Z) = (P + Q) + Z
  - *Abelian*: P + Q = Q + P

- Scalar multiplication nP is just adding P to itself n times
  - This is analogous to "exponentiation" in $\mathbf{Z}_p^*$
  - Can compute with double and add algorithm (same as square and multiply)
- Can pick generator P that defines cyclic subgroup of E

  { 0P, 1P, 2P, 3P,…, qP } = all points of interest on curve

  (choose so q is big prime)

Building elliptic curve groups

- How do we find suitable curves?
  - Pick large prime p
  - Pick values for a and b to define

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

  - Determine size of group, see if it is prime
- There are efficient algorithms for all this
- Short, better answer: Use predefined ones
  - NIST curves
  - Curve25519

Elliptic curve DH

X

Y

Pick random x from $\mathbf{Z}_q$
$X = xP$

Pick random y from $\mathbf{Z}_q$
$Y = yP$

$K = H(xY)$

$K = H(yX)$

X

Y

Pick random x from $\mathbf{Z}_{|G|}$
$X = g^x$

Pick random y from $\mathbf{Z}_{|G|}$
$Y = g^y$

$K = H(Y^x)$

$K = H(X^y)$

Elliptic curve DLP

- Given xP compute x
- Same as g^x compute x, just different group!

- Best known algorithm against well-chosen ECC group version runs in time $q^{0.5}$

Baby-Step Giant-Step algorithm

- ECDLP: Given xP for random x, compute x
- Rewrite x as x = am + b with m = ceil($q^{0.5}$)

  xP + (-am)P = bP
- For b = 1, ..., m
  - Store (b,bP)
- For a = 1, ..., m
  - Check if xP + (-amP) equals one of precomputed bP
  - Return am + b
- Works in time O($q^{0.5}$) and space O($q^{0.5}$)
- Pollard rho method: reduce space to constant

Summary

- Elliptic curves are specially constructed groups where DLP is conjectured to be hard
- These are faster than RSA or DLP over $\mathbf{Z}_p^*$
- Being used increasingly in practice
  - EC-DSA (bitcoin)
  - TLS EC-DHE (elliptic curve ephemeral DH)