Elliptic Curve Cryptography

Erica Dong

Background 01

Abstract Algebra Crash Course

Group: a set G and binary operation on G, $\cdot$, denoted (G, $\cdot$)

- Associativity
- Identity
- Inverses
- Closure

Abelian group: a group that is also commutative

E.g. ($\mathbf{Z}$, +)

Field: a set F and binary operations +, $\cdot$, denoted (F, +, $\cdot$)

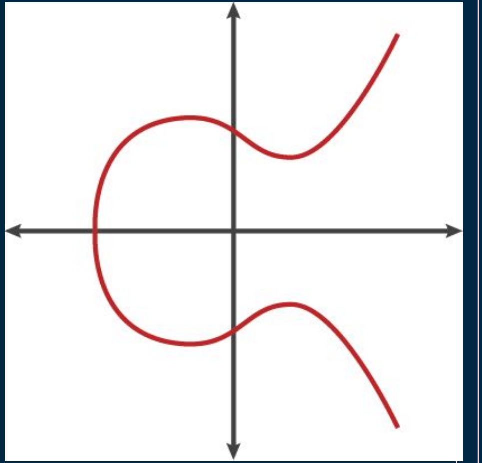
- Associativity
- Commutativity
- Identities
- Additive (+) inverses
- Multiplicative ($\cdot$) inverses (all nonzero elements)
- Distributivity of $\cdot$ over +
- Closure

E.g. ($\mathbf{Z}_5$, +, $\cdot$)

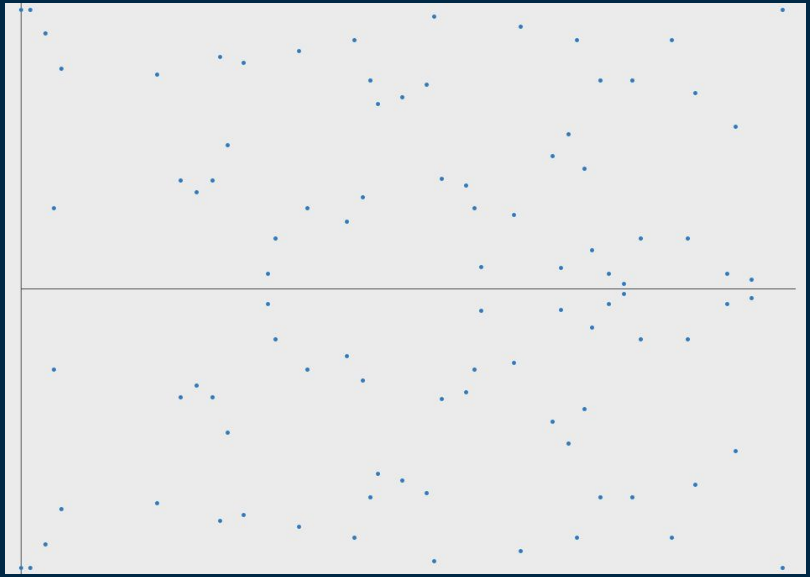
Elliptic Curves

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

- Curve over a finite field
  - Finite for cryptographic purposes
- Set of solutions and point at infinity forms an abelian group



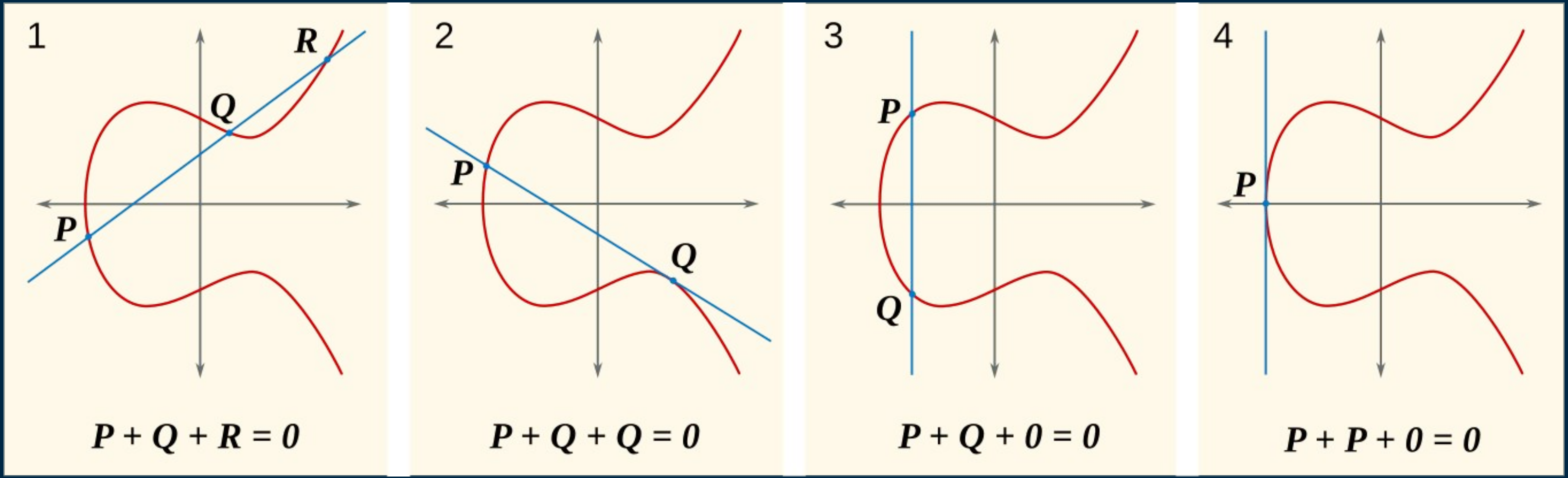
Elliptic Curves



What Operation?

- Ex. (0, 1) + (2, 1) on $y^2 = x^3 + x + 1$ over $\mathbf{Z}_5$
  - Line between (0, 1) and (2, 1): y = 1
  - $(1)^2 = x^3 + x + 1$
    - $0 = x^3 + x = x(x + 3)(x+2)$
    - (3, 1) is another solution
  - Reflect (3,1) over x-axis
  - (0, 1) + (2, 1) = (3, -1) = (3, 4)
- If not vertical, there is always another solution
- Line construction is abelian => + is abelian
- P = (a, b). The line between P and infinity is x = a, which intersects the curve at (a, -b). Then P + infinity = P, and infinity is the identity
- -P is the reflection of P across the x-axis
- Closed

What Operation?



Cryptographic Applications

02

Elliptic Curve Discrete Logarithm Problem

- **Discrete logarithm problem** (DLP): in a group G with a, b $\in$ G, find k $\in$ G s.t. k*a = a + … + a (k times) = b
  - Used in RSA and Diffie-Hellman key exchange
- **Elliptic curve DLP** (ECDLP): special case of DLP where the group is the group of points on an elliptic curve over some finite field
- Computational hardness is unsolved, so security of ECC is based on the computational Diffie-Hellman assumption
- Like RSA, broken by Shor's algorithm 😱

A Little More Abstract Algebra

Cyclic subgroup: for any element g in group G, $\langle g \rangle = \{ k*g \mid k \in \mathbf{Z} \}$

- g is called the **generator** of $\langle g \rangle$
- **order(g)** is the number of elements in $\langle g \rangle$

Elliptic Curve Diffie–Hellman Key Exchange

1. Alice and Bob publicly agree on domain parameters, including the generator g from the elliptic curve and order(g) = n
2. Alice and Bob each have a secret key s in [1, n-1] and a public key $K = s^*g = g + \ldots + g$ (s times) - secure unless Eve can solve ECDLP
3. The shared secret $(x_k, y_k) = s_A K_B = s_A s_B g = s_B s_A g = s_B K_A$
4. x_k is used in a key derivation function to obtain encryption key(s)

Dual EC DBRG

Dual Elliptic Curve Deterministic Random Bit Generator

1. Take an elliptic curve over field F, where F has prime size
2. Take some seed from F, and let the initial state be s_0 = seed
3. Choose two random points, P and Q, over the curve
   a. $X(x, y) = x$ and $t(x) = x \bmod (p / 2^{16})$ - utility functions
4. Let $f(x) = X(xP)$ and $h(x) = t(X(xQ))$
5. Then $s_k = f(s_{k-1})$ and $r_k = h(s_k)$

Dual EC DBRG

- Snowden documents indicate plans by NSA to install backdoor in Dual EC DBRG ?!
  - Could be used to decrypt SSL/TLS communications, etc.
- One-way trapdoor:
  - Say the NSA knows that P = jQ on the curve
    - Determining if the backdoor exists = ECDLP
  - $r_k = X(s_k Q)$ is known to the attacker
  - $s_{k+1} = X(s_k P) = X(s_k jQ) = X(j s_k Q) = X(j X^{-1}(r_k))$
  - Elliptic curves are symmetric across the x-axis, so X^{-1} has only two possible values
  - Truncation can be brute-force-reversed - outputs way too many bits
- No security reduction published

Extra

- Requires much smaller keys than factoring-based algos like Diffie-Hellman and RSA
  - 256 bit key in ECC => 3072 bits in RSA
  - Index calculus doesn’t work
- Real uses: digital signatures for cryptocurrencies (ECDSA), key-agreement for SSL/TLS, CSPRNGs
  - iMessage, US government internal communications, Tor, Bitcoin, etc.

THANKS!

Questions?