① Preliminaries.

Public : trapdoor
Private : symmetric

- Order of an element in a group
- $\phi(n)$ : Euler totient function.
- generator of a group

* key one : existance of at least 1 generator
(And the proof)

② Discrete Logarithm Problem

○ private ○ public

③ Diffie-Hellman Key Exchange

A, B agreed.

1) g : generator, p : prime modulars (Field : $\mathbb{Z}/p\mathbb{Z}$)

2) A (a) (private) $1 < a < p-1$ avoid triviality. $1 = g^a \bmod p$ (send)

3) B (b) (private) $m = g^b \bmod p$ send

4) A takes $m$ and $m^a \bmod p$

5) B takes $l$ and $l^b \bmod p$.

6) $m^a \bmod p = (g^b)^a \bmod p = (g^a)^b \bmod p = l^b \bmod p = s$.

generator of multiplicatively written group.

$S$ is the shared key.

Secret.

what hackers have: $p, g, l, m$

④ RSA: 1) Given $N = pq$ Find $p, q$, integer factorization problem.

2) RSA: Find $m$ s.t. $c \equiv m^e \pmod n$

where $(n, e)$ is a public key and $c$ is an RSA ciphertext.

Both are easy from 1 direction and hard from another one.

RSA: Bob: $p, q$ large primes

compute $n = p \times q$ (public)

compute $\phi(n) = (p-1)(q-1)$

$= n - p - q + 1$

$e$ public $\gcd(e, \phi(n)) = 1$

Find $d$ $xe \equiv 1 \mod \phi(n)$

Send public key → Alice

$(n, e)$

Attackers don't have $d$.

Alice: message $m$

compute ciphertext $c$.

$c \equiv m^e \bmod n$

send $c$ to Bob.

Bob: compute the original $m \equiv c^d \bmod n$

$m \equiv c^d \bmod n = m \equiv m^{ed} \bmod n$

$m \equiv m^{1+k\phi(n)} \bmod n$.

Euler's theorem, $m^{\phi(n)} \equiv 1 \bmod n$ for $\gcd(m, n) = 1$

RSA improvement ① Authenticity Sender

② Make sure the message is not changed (attacked)

However not sustainable for modern mobile devices

Need new system!!!

① ECC : more security & smaller key size

1. Def: $y^2 = x^3 + ax + b$

2) requirements: 1) characteristic not equal to 2 or 3. 2) $\Delta = -16(4a^3 + 27b^2) \neq 0$ non-singularity

3) $x^3 + ax + b$ distinct roots.

For 1) char 2: $y^2 + xy = x^3 + ax^2 + b$

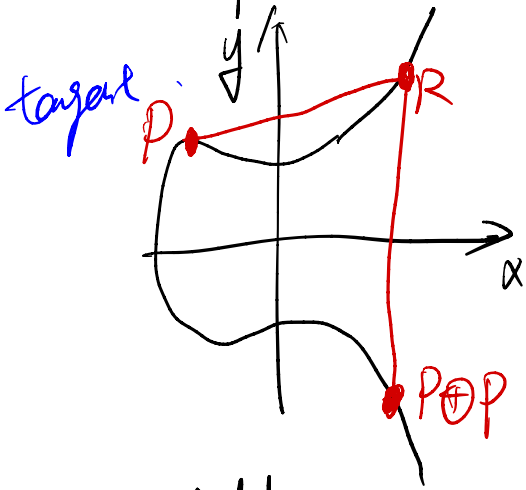
char 3: $y^2 = x^3 + ax^2 + b$.

## ② Elliptic Curve over the Reals.

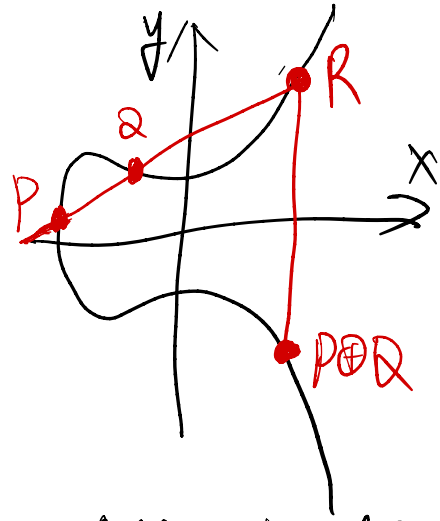
Graphic: ① Two Components (distinct): $\Delta > 0$

② One Component (connected): $\Delta < 0$

Group law:




Adding a point to itself.




Adding two distinct points.

O: points at infinity, all vertical lines in the space where the curve exists go through this point.

Form an abelian group:

① $(P \oplus Q) \oplus R = O$.

② $P \oplus O = P$ for all $P \in E$.

③ If $P \in E$, then there's a point $-P \in E$ s.t. $P \oplus (-P) = O$

④ $(P \oplus Q) \oplus R = P \oplus (Q \oplus R)$

Proof especially for ④

Associativity: Use the graphic construction to represent the two side into T and S.

And prove $T = S$.

# Homogenization Process

Suppose $F(x,y) = y^2 - (x^3 + ax + b)$ standard Weierstass equation.

$$f(x:y:z) = z^3 F(x/z, y/z) = z^3\left(\left(\frac{y}{z}\right)^2 - \left(\left(\frac{x}{z}\right)^3 + a\left(\frac{x}{z}\right) + b\right)\right)$$

$$= y^2 z - (x^3 + axz^2 + bz^3)$$

Use dimension to prove.

## ③ Elliptic Curves over Finite Fields.

Graph: discrete set of points.

$E(\mathbb{F}_q)$: finite abelian group in $\mathbb{F}_q$

Either cyclic or a product of 2 cyclic groups.

Why $\mathbb{F}_p$ useful here:

① All calculations performed mod p, so the results remain within the field.

② All non-zero element has multi inverse, division ✓

$\#E(\mathbb{F}_q)$ ⟶ difficulty of solving the discrete log problem ⟶ security of the system.

Hasse's theorem: bound for $|\#E(\mathbb{F}_q) - (q+1)| \leq 2\sqrt{q}$

⟶ indicates # grows approx as q (size of $\mathbb{F}_q$)

Determine the exact number, Schoof's algorithm

uses @ Hasse's theorem, Frobenius endomorphism $\phi$, Chinese remainder theorem and division polynomials.

① Computing Large Multiples of a Point.

$G$: abelian group on $E$.

$\oplus$: the operation we defined.

multi by scaler $t$ on $E$:

$$tP = P \oplus P \oplus \cdots \oplus P \quad (t \text{ terms})$$

How to compute $tP$? "Double and Add"

↓

① Binary Expansion:

$t = t_0 + 2t_1 + 2^2 t_2 + \cdots + 2^k t_k$

where $t_i \in \{0, 1\}$

② $tP = t_0 P + 2t_1 P + 2^2 t_2 P + \cdots + 2^k t_k P$.

We can compute $2^k P$ by $k$ doublings and $k = \lfloor \log_2 t \rfloor$

This algo: $m$ steps. at most 2 operations per step.

worst $2m$ operations.

computation time $O(m)$ ( $O(\text{bytes})$ )

exponentially better

⑤ Elliptic Curve Discrete Logarithm Problem

E: elliptic curve

G: abelian group.

If $P \in G$ and $R = kP$ is a multiple and $k$ is a scalar.

Def: DLP: Find an integer $k$ s.t. $kP = R$ where $P$ is a generator point on $E$ and $R$ belongs to the cyclic subgroup generated by $P$.

Hard to solve: Best known algorithms have exponential runtime.

⑥ Elliptic Curve Diffie-Hellman (ECDH). • public • private

(1) A.B agree on $(p, a, b, P, n, h)$

p: prime. a, b random values make up the equation of E.

P: random point on E n: order of P

h: a cofactor of the group G.

[2] A: pick private key $d_A$ ($d_A \in [1, n-1]$ and $d_A$ is integer)

compute $Q_A = d_A G$

B: pick $d_B$

compute $Q_B = d_B G$

[3] Exchange $Q_A$, $Q_B$

[4] Compute shared secret:

A: uses $Q_B$ and $d_A$

$S_A = d_A Q_B$

B: uses $Q_A$ and $d_B$

$S_B = d_B Q_A$

We see $S_A = S_B$ (shared key)

Because $d_A(d_B Q_A) = d_B(d_A Q_A)$

solve $d_A$ using $Q_A$ and $P$

shared secret:

x or y-coordinate of $d_A d_B P$. point.

Third party E: only has $P$, $Q_A$ and $Q_B$

unable to get $d_A d_B P$ without solving discrete log problem.

[5] Use this to derive a symmetric key (key derivation function (KDF))

And communicate via ciphers.

Remark: Lagrage's Theorem $h \cdot n = |G|$

$n$ should be a large number $\rightarrow$ harder to solve.

$h$ be small ($h \leq 4$), preferable $h = 1$.

Reason? Efficiency ↑ Attacks ↓ (specific)

NIST Recommendations

Usually participants don't derive the parameters themselves. (time consuming : computation of $|G|$ Schoof's algorithm.)

Organizations publish recommended curves with computed parameters. ← public knowledge

e.g. secp256k1 Bitcoin.

Alternatively, asymmetric cryptosystem can be used. usually slower

⑦ ElGamal System on Elliptic Curves:

$p$: prime  $E$: elliptic curve over $\mathbb{F}_p$

$P$: randomly chosen point on $E$

$n$: order of $P$.  $Q$: public key we just computed (of the intended recipient).

$m$: plaintext.

(1) A: comes up with (public) $f: m \to M$ (maps a message $m$ to a point $M$ on $E$).

pick $k \in_R [1, n-1)$ computes $C = kP$

$M = f(m)$  $D = M + kQ$

Send $(C, D)$ to B.

(2) B: use $d$.

First computes $M = D - dC$

Then performs $m = f^{-1}(M)$

Remark: $dC = d(kP) = k(dP) = kQ$

If a third party want to receive a value of $M$, needs to compute $kQ$. Computing $kQ$ given $C = kP$ and $Q$. The same discrete logarithm problem.

ECDH: establish a shared secret over an insecure channel.

ElGamal: encrypting messages using recipient's public key so that only the recipient can decrypt it using their private key.

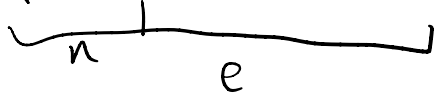
## ⑧ Elliptic Curve Digital Signature Algorithm.

Another advantage of ECC: provides opportunity for the parties to "sign" their messages. (receiver knows it's him)

Signature generation algo:

m: message  n: prime order of the subgroup generated by P.

1. SIGGEN(m, n, P)
2. Compute $e = hash(m)$
3. $z :=$ l leftmost bits of e where l is bit length of n.




4. Select $k \in_R [1, n-1]$ (uniformly chosen)
5. Compute $(x_1, y_1) = kP$
6. Compute $r = x_1 \bmod n$.
7. if $r = 0$ then
   select a new k back to 4.
8. Compute $s = k^{-1}(z + rd_A) \bmod n$ where $d_A$ is A's private key and $k^{-1}$ is multiplicative inverse of $k \bmod n$.

sender

9. if $s=0$ then:

Select a new $k$, back to step 4.

10. Return $(r, s)$.

---

Signature verification algo: Need $Q_A$ publicly of A.

Not fully understood.

Main drawback: need to compute domain parameters :C

1. SIGVER($E$, $n$, $Q_A$, $s$, $z$, $P$, $r$)
2. if $Q_A \neq O$ and $Q_A \in E$ and $nQ_A = O$ and $r, s \in R[1, n-1]$ then
3. Compute $u_1 = zs^{-1} \bmod n$.
4. compute $u_2 = rs^{-1} \bmod n$.
5. Compute the point $(x_1, y_1) = u_1 P + u_2 Q_A$

6. if $(x_1, y_1) \neq O$ and $r \equiv x_1 \bmod n$ then

7. "The signature is valid".

8. else,

9. "The sig not valid".

10. else.

11. "The sig not valid."

It's rather expensive to compute the domain parameters. In addition, ECC facing dangers:

Quantum attacks / classical attacks.

⑨ Attacks on ECC and Pollard's rho algorithm.

(1) Classical attacks: slow + require exponential running time to solve the ECDLP. Most effective:

(2) Quantum attacks: E.g. Shor's algorithm which finishes the attack in polynomial time.

Once quantum computers become practical, ECC will be in big danger.

Pollard's rho-algorithm.

Let E be the Ec and DLP on E as defined before. n be the order of the subgroup generated by P.

Running time roughly $O(\sqrt{n})$ and least $O(1)$ in space complexity. Best!

Idea: find distinct pairs of integers $(a_{j_1}, b_{j_1})$ and $(a_{j_2}, b_{j_2})$ s.t. $a_{j_1}P + b_{j_1}Q = a_{j_2}P + b_{j_2}Q$.

Floyd's cycle finding algorithm.

1. Partition the set of points on E into 3 subsets. Define a suitable iterating function $f$ and apply to them.

2. Iteration generate sequence:

$A_i = a_j P + b_j Q$

Once there's a match $A_{i_1} = A_{i_2}$

We get $a_{j_1}P + b_{j_1}Q = a_{j_2}P + b_{j_2}Q$.

Will eventually occur: O points finite on the curve

⊗ subgroup generated by P is cyclic.

3. $a_i P + b_i Q = a_j P + b_j Q$

$(a_i - a_j) P = (b_j - b_i) Q$

$k = (a_i - a_j)(b_j - b_i)^{-1} \mod n.$

Remark: relies on the birthday paradox — for a set of randomly chosen points, the probability of a collision is high → ensure the efficiency of the algo.

---

(10) Future of ECC.

D+H isogeny-based system may be proved to be quantum-resistant!

(1) Uses supersingular curves over $\mathbb{F}_{p^2}$ where p is a prime.

(2) A supersingular curve is defined as having no points of order p.

(3) An isogeny $\phi: E_1 \to E_2$ is a rational map s.t. the number of points on the two curves is the same.

Reason: non-abelian. Further research needed.

Classical algo: $\exists$ non-abelian groups resistant to quantum attacks.

① Hash-Based

② Lattice-Based

③ Multivariate Equations

④ Error Codes.