Cryptography

classmate
Date 02/04/24
Page Tuesday

Alice (K) — Bob (K)

$C = Enc(m, K)$ —C→ $m = Dec(C, K)$

→ Sharekey : Meet in confidential room

→ **Diffie & Hellman** (1976)

Key exchange (Public Key Cryptography).

$(G, *)$ → group (cyclic)

Alice — G = cyclic group $= \langle g \rangle$ — Bob

$|G| = n$

Secret: $0 \le a \le n-1$

$K_a = g^a$

$K_a = g^a$ →

Secret: $0 \le b \le n-1$

$K_b = g^b$

$K_b = g^b$ ←

He has: a, $K_b = g^b$

Secret key of Alice

Compute $K_b = (g^b)^a = g^{ba}$

He has: b, $K_a = g^a$

Public key of Alice

Compute $(K_a)^b = (g^a)^b = g^{ab}$

$g^{ba} = g^{ab}$

Shared Secret Key

$= g^{ab}$

classmate
Date
Page

Alice

Secret key : a
Public key : g^a

Bob

Secret key : b
Public key : g^b

good cyclic grp $\rightarrow g^x \rightarrow$ Public
$x \rightarrow$ Secret

Based on the property of group $G = \langle g \rangle$ finding x from g^x will be computationally difficult
$\Rightarrow$ This hard problem is known as Discrete Log Problem.

$\Rightarrow y = g^x$

$\Rightarrow \log_g y = x$

↓

will give decimal value

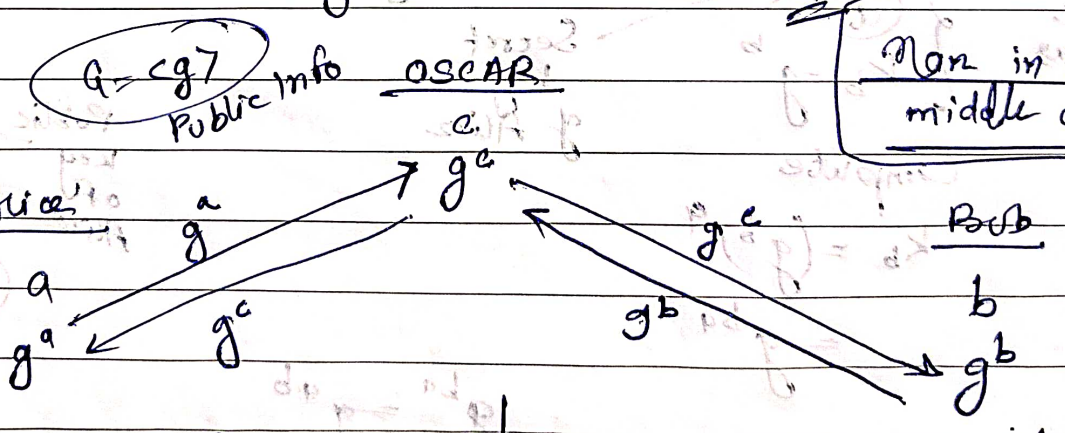
Brute force

$g, g^2, g^3, \ldots$ then find x.

$G = \langle g \rangle$ Public info

OSCAR

Man in the middle attack



Comm is happening but it is seen by Oscar!

$c_1 = Enc(m, g^{ac})$ → c_1 → $m = Dec(c_1, g^{ac})$
$c_2 = Enc(m, g^{bc})$ → c_2 → $Dec(c_2, g^{bc}) = m$

can add Mac to eliminate attack

classmate Date Page

$p = 2^{255} - 19$ → fixed.

Alice: $0 < a < n$, g^a

$|G| = n$ → large

$\subset Z_p^*$, mod p

$= \langle g \rangle$

Bob: $0 < b < n$, g^b

x^c is very hard, we need efficient algo when $c \approx 255$ e

$$c = \sum_{i=0}^{l-1} c_i 2^i$$

$$x^c = x^{\sum_{i=0}^{l-1} c_i 2^i} = \prod_{i=0}^{l-1} x^{c_i 2^i}$$

$$= x^{c_0 2^0} x^{c_1 2^1} x^{c_2 2^2} x^{c_{l-1} 2^{l-1}}$$

l times multiply

$\therefore \log(n)$ time mult

$\hookrightarrow \log(c)$ ok $\to x^c$.

Want to compute x^c

Algorithm

```
z = 1
for i = l-1 to 0
    z = z^2
    if c_i = 1
        z = z x x
return (z)
```

leftshift = square

add 1 to last in binary = multiply by x

Example

$3^5 = x^c$

$5 \to (101)$

$z = 1$

$i = 2$

$z = 1$

$c_2 = 1$

$z = 3$ ← [$i = 1$, $z = 3^2$] for 0

[$i = 0$, $z = z^2 = (3^2)^2$] for 1

for 1

in $\log(5)$ time

Square & Multiply Algo (2)

→ Three famous cryptographers name.

⊛ RSA —

$\phi(m)$: Number of +ve integers $< m$ which are co-prime to m

$$\downarrow$$

$$\gcd(x, m) = 1$$

Euler Totient function

$\phi(8) = 4$ $\quad \{1, 3, 5, 7\}$

$\phi(p) = p-1$ $\quad p$: prime

$\phi(p^k) = p^k - p^{k-1}$

$= p^k\left(1 - \frac{1}{p}\right)$

$n = pq$

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

$\gcd(a, m) = 1$ → set of all integers.

$S = \{x \bmod m\}$

$= \{r_1, r_2, r_3, \ldots r_m\}$

$S_1 = \{ar_1, ar_2, \ldots ar_m\}$

$ar_i = ar_j \quad r_i \neq r_j$

$ar_i \equiv ar_j \pmod m \quad r_i \not\equiv r_j \bmod m$

$\gcd(a, m) = 1$

$\Rightarrow 1 = a.b + ms$

$\Rightarrow \exists\, b$, s.t. $a.b \equiv 1 \pmod m$

$ar_i \equiv ar_j \bmod m$

$b\,ar_i \equiv b\,ar_j \bmod m$

$r_i \equiv r_j \pmod m$

$\Rightarrow \boxed{ar_i \not\equiv ar_j \pmod m}$

Euler's Theorem —

If $\gcd(a, m) = 1$ then

$$a^{\phi(m)} \equiv 1 \pmod m$$

$\Rightarrow S = \{x \mid \gcd(x, m) = 1\}$

$= \{s_1, s_2, \ldots, s_{\phi(m)}\}$

$\gcd(s_i, m) = 1$

$S_1 = \{as_1, as_2, \ldots, as_{\phi(m)}\}$

$as_i \not\equiv as_j \pmod m$

$|S| = \phi(m)$

$|S_1| = \phi(m)$

$s_i \equiv as_j \pmod m$

$$\prod_{i=1}^{\phi(m)} s_i \equiv \prod_{j=1}^{\phi(m)} as_j \pmod m$$

$$\prod_{i=1}^{\phi(m)} s_i \equiv a^{\phi(m)} \prod_{j=1}^{\phi(m)} s_j \pmod m$$

$\gcd(s_i, m) = 1$

$$\boxed{a^{\phi(m)} \equiv 1 \pmod m}$$

⊛ Fermat's Theorem:-

If p is a prime number and $p \nmid a$ (not dividing) then

$$a^{p-1} \equiv 1 \pmod{p}$$

$a^{\phi(p)}$

$\phi(p) = p-1$

$p \nmid a \Rightarrow \gcd(a, p) = 1$

$\Rightarrow a^{\phi(p)} \equiv 1 \pmod{p}$

$\Rightarrow \boxed{a^{p-1} \equiv 1 \pmod{p}}$

$\Rightarrow \boxed{a^p \equiv a \pmod{p}}$ → if a is divisible by p

⊛ RSA Cryptosystem

Fact

1) $g(a, m) = 1$ then:
$a^{\phi(m)} \equiv 1 \pmod{m}$

2) $a^{p-1} \equiv 1 \pmod{p}$

(p = No. of int. less than m & co-prime to m)

1) $n = pq$ here $p, q \rightarrow$ primes (random)

2) plaintext space $= Z_n$
ciphertext space $= Z_n$

3) Key Space $= \{k = (n, p, q, e, d) \mid ed \equiv 1 \pmod{\phi(n)}\}$

Key is Tuple : $k = (n, p, q, e, d)$.

4) Encryption:- $E(x, K) = C$

$$C = E(x, K) = x^e \pmod n \qquad x \in Z_n$$

5) Decryption:- $Dec(C, K) = x$

$$x = Dec(C, K) = C^d \pmod n$$

$\Rightarrow$ $ed \equiv 1 \bmod \phi(n)$

$ed - 1 \equiv t \cdot \phi(n)$ → $-t$ basically

$1 = ed + (t)\phi(n)$

$1 = \gcd(e, \phi(n)) = ed + t_1 \phi(n)$

e is coprime to $\phi(n)$

d is mul. inverse of e. (can find inv in poly time by ext. eucl. algo)

Enc $C = x^e \pmod n$

$x = C^d \pmod n$

Dec $C^d = (x^e)^d \bmod n$

$= x^{(ed)} \bmod n$

$= x^{1 + t\phi(n)} \bmod n$

$= x \cdot x^{t\phi(n)} \bmod n$

$= x \cdot x^{t[(p-1)(q-1)]} \bmod (p \cdot q)$ ← subs $\phi(n)$ and n

we have to prove that this is $= 1$

$e \cdot d \equiv 1 \pmod{\phi(n)}$

$ed - 1 = t \cdot \phi(n)$

$ed = 1 + t\phi(n)$

Date ______
Page ______

$x^{t(p-1)(q-1)} \bmod (pq)$ $\quad x \in (Z_n)$; $n = pq$

$x \in 0$ to $n-1$

since see rules.

~~we check.~~

~~$x^{t(p-1)(q-1)} \bmod (p)$~~

Fermat

divides $x \mid n$	$x \nmid n$
$\Rightarrow x$ is either p or q	$\Rightarrow \gcd(x, p) = 1 \rightarrow x^{p-1} = 1 \bmod p$
	$\gcd(x, q) = 1 \rightarrow x^{q-1} = 1 \bmod q$
(C I)	(C II)

Take (C II) we check

$\rightarrow x^{t(p-1)(q-1)} \bmod (p)$

$\left(x^{p-1}\right)^{t(q-1)} \bmod (p)$

$1 \bmod (p) \rightarrow$ As $x^{p-1} \equiv 1 \bmod p$

Now $\rightarrow x^{t(p-1)(q-1)} \bmod q$

$= 1 \bmod (q) \rightarrow$ As $x^{q-1} \equiv 1 \bmod q$

finally, $x^{t(p-1)(q-1)} \equiv 1 \pmod{pq}$

from above two results

$\gcd(x, p) \neq 1$
$\gcd(x, q) \neq 1$

Toh direct hoga

any one gcd = 1 have challenge

$x = 10, n = 15$ works!

since once gcd = 1

Public key ← Anyone can encrypt but only he can decrypt it

Take (CI) $x \mid n$

$x \equiv 0 \bmod n \equiv x \bmod n$

$x^{ed} = 0 \bmod n$

$\equiv x \bmod n$

Alice

$n = pq$

$ed \equiv 1 \bmod \phi(n)$

Public key of Alice

$= (n, e)$

Secret key of Alice

$= (p, q, d)$

$\{n, e\} \rightarrow$ Public

$p, q \rightarrow$ large usually

done equal no of bits

so that factors

difficult.

Bob

$x \rightarrow$ msg,

$y = x^e \bmod n$

$p, q \rightarrow$ large so finding p, q from n is computationally hard problem.

$n = pq \rightarrow \phi(n) \rightarrow d$

Solve factorization → Break RSA

takes $2^{\sqrt{n}}$ computations !!

Source algo can do it using quantum Algo

Quantum can break entire Public key crypto

∴ Public key crypto based on comp. hard.

* **Factorization problem** → find p, q → then d

* **RSA problem** –

$$\left\{\begin{array}{l}(n, e) \to \text{Public} \\ C = x^e\end{array}\right\} \Rightarrow \text{find } x, \ \text{find } d \text{ directly}$$

Solve Factor —✓→ Solve RSA
Solve RSA —✗→ Solve Factor

Alice – **Sign on m** — **Bob**

KG
$$\left\{\begin{array}{l} n = pq \\ \phi(n) = (p-1)(q-1) \\ ed \equiv 1 \pmod{\phi(n)} \end{array}\right.$$

PK : $\{e, n\}$

SK : $\{d, p, q\}$

Used in digital signature of document

Sign Algo

$S = m^d \bmod n$

msg.

→ S →

Verification Algo

$V = S^e \bmod n$

if $V = m$ then output 1
else output 0

Secure is factorization prob is hard

* **DSA** $(P, S, K, \text{Sign}, V)$ ——→ Used in Authentication

Dig. sign. Algo.

- P → plaintext space
- S → signature space
- K → Key tuple
- Sign → algo
- V → algo

RSA

$C_1 = m_1^e \bmod n$

$C_2 = m_2^e \bmod n$

$C_1 \times C_2 = (m_1 m_2)^e \bmod n$

} Computation on Encrypted Data

$S_1 = m_1^d \bmod n$

$S_2 = m_2^d \bmod n$

$S_1 * S_2 = (m_1 * m_2)^d \bmod n$

} Computation on Authenticated Data

little bit issue since $(m_1 * m_2)$ ka sign bhi generate kar sakte without you actually you signing it

So, $S_1 = [h(m_1)]^d \bmod n$

$S_2 = [h(m_2)]^d \bmod n$

$S_1 S_2 \neq [h(m_1 m_2)]^d \bmod n.$

So

Maths

To solve,

$ax \equiv b \pmod m$ — (1)

Equivalent ↓

$ax - my = b$ — (2)

↓ → Solve using Ext. Eucl. Algo

$ax_0 + my_0 = \gcd(a, m)$

divide ←

If $\gcd(a,m) \mid b$ then (2) have soln

Using Extended Euclidean Algo: $\gcd(a,m) \mid b \Rightarrow t \cdot \gcd(a,m) = b$

$ax_0 + my_0 = \gcd(a,m)$

$\Rightarrow ax + my = b \Rightarrow a(tx_0) + m(ty_0) = t \cdot \gcd(a,m)$

$ax \equiv b \pmod m$

$ax - my = b$ ——— $ax_0 - my_0 = b$

generalized soln for our eqn

$x = x_0 + \frac{m}{\gcd(a,m)} n$

$y = y_0 + \frac{a}{\gcd(a,m)} n$

$n \in Z$

$x \equiv a_1 \pmod{m_1}$ ——— (1)

$x \equiv a_2 \pmod{m_2}$ ——— (2)

Given: Here $\gcd(m_1, m_2) = 1$

$\Rightarrow$ Eqn (1) has soln $\xrightarrow{\text{means}} m_1 \mid (x - a_1)$

$x \equiv a_1 \pmod{m_1}$

$x = a_1 + m_1 y$ ——— (*)

$\Rightarrow$ If x is also a soln of (2) —

$x \equiv a_2 \pmod{m_2}$ — Now put (*) in (2)

$\Rightarrow a_1 + m_1 y \equiv a_2 \pmod{m_2}$

$\Rightarrow m_1 y \equiv (a_2 - a_1) \bmod m_2$

$\gcd(m_1, m_2) = 1$ ∴ will divide anything always

This eqn will have soln of the form $y = y_0 + \frac{m_2}{\gcd(m_1, m_2)} n$ ——— $(= 1)$

→ from extended eucl. algo.

Solve this using Extended eucl. to get y_0

$y = y_0 + m_2 n$

From (*) $\longrightarrow x = a_1 + m_1 y$

$x = a_1 + m_1 (y_0 + m_2 n)$

$$x = (a_1 + m_1 y_0) + n m_1 m_2$$

$$\boxed{x = x_0 + n \cdot m_1 m_2}$$ → unique under modulo $m_1 m_2$.

$$\boxed{x \equiv x_0 \pmod{(m_1 m_2)}}$$ *

Chinese Remainder Theorem

$$\left.\begin{aligned} x &\equiv a_1 \pmod{m_1} \\ x &\equiv a_2 \pmod{m_2} \\ &\vdots \\ x &\equiv a_r \pmod{m_r} \end{aligned}\right\}$$ where $m_1, m_2, \ldots m_r$ are pairwise co-primes.

$\left[\gcd(m_i, m_j) = 1 \quad i \neq j\right]$

then the above system has a unique soln modulo $(m_1 m_2 \ldots m_r)$

Proof using above thing:

Let for each $j = 1, 2, \ldots r$

define $\delta_j \longrightarrow \delta_j = \begin{cases} 1 \pmod{m_j} \\ 0 \pmod{m_i} \text{ if } i \neq j \end{cases}$

$$\boxed{x = \sum_{j=1}^{r} \delta_j \cdot a_j}$$ Satisfies all the eqns.

$$M = m_1 \cdot m_2 \cdot \ldots m_r$$

$$\gcd\left(\frac{M}{m_j}, m_j\right) = 1$$ → [all m_j's without m_j].

As $\gcd(m_i, m_j) = 1$ (if $i \neq j$)

Example:-

$\frac{M}{m_1} = m_2 m_3 \cdots m_r$

$\frac{M}{m_1} \cdot b_1 \equiv m_2 \cdots m_r b_1 \pmod{m_2}$

$$\left[\frac{M}{m_j} \cdot b_j \equiv 1 \pmod{m_j}\right] \longrightarrow \boxed{\delta_j = \frac{M}{m_j} b_j}$$ *

b_j: mul. inv. of $\frac{M}{m_j}$ under modulo m_j

1 for that m_j else 0

⊕ Uniqueness—

Assume x' is another soln of the above system then $x' \equiv x \pmod{m_1 \cdots m_r}$

$x' \equiv x \pmod{m_1}$

$x' \equiv x \pmod{m_2}$

$\vdots$

$x' \equiv x \pmod{m_r}$

$\therefore$ $x' \equiv a_1 \pmod{m_1}$

$x \equiv a_1 \pmod{m_1}$

$x' - x \equiv 0 \pmod{m_1}$

$x' = x \pmod{m_1}$

$x' \equiv x \pmod{m_1 m_2 \cdots m_r}$

$x = \sum_{j=1}^{r} \delta_j q_j$ is the unique soln under modulo $(m_1 m_2 \cdots m_r)$

x' and x are same under ↑

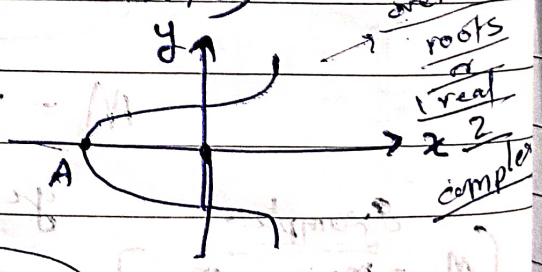
⊕ Elliptic Curve :—

1) $a, b \in R$ → makes sure roots are not overlapping.

2) $4a^3 + 27b^2 \neq 0$

3) Eqn of the curve $(y^2 = x^3 + ax + b)$ → symm around x axis

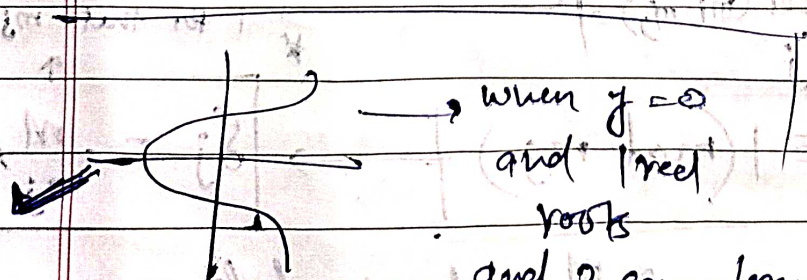
where $(x, y) \in R^2$

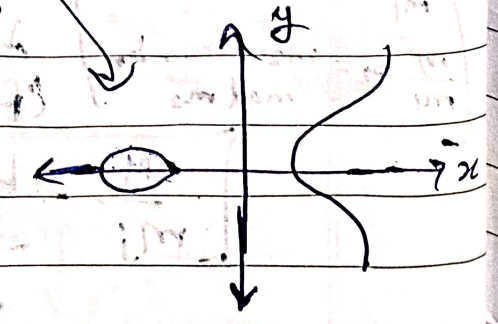


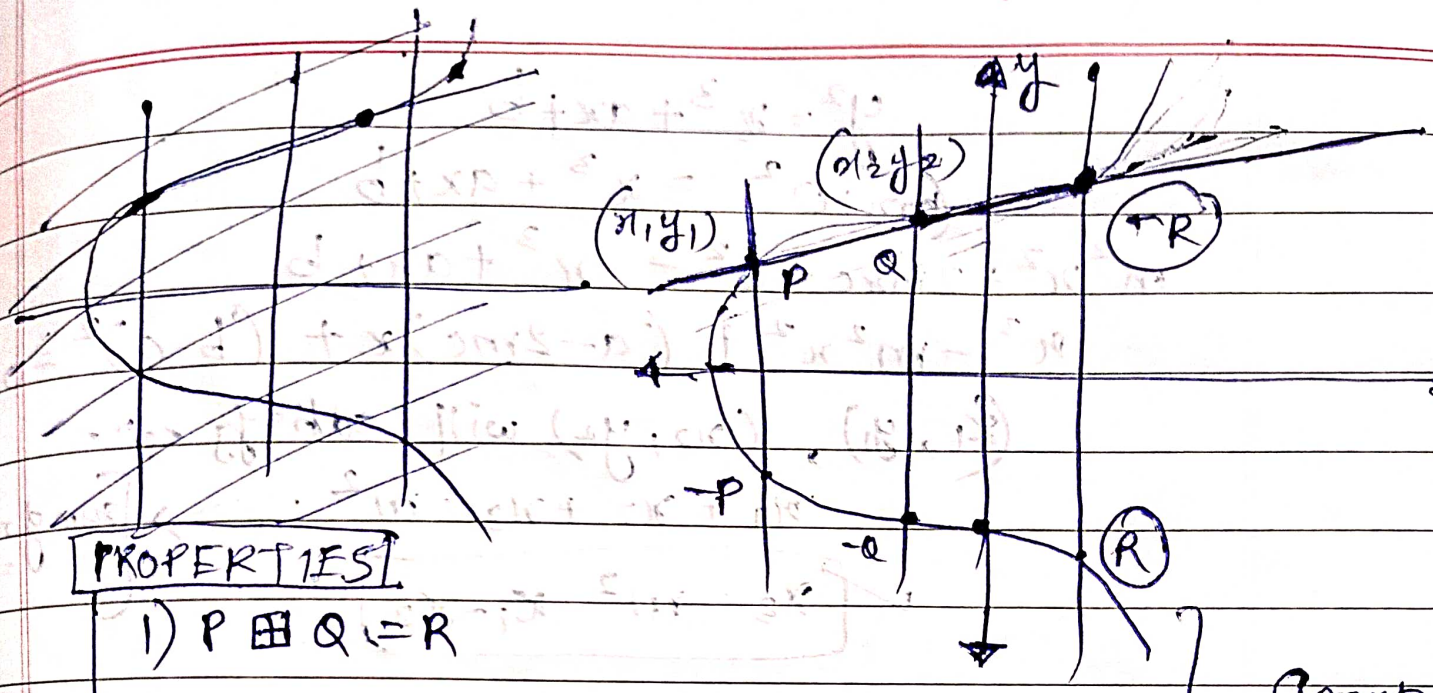
Now 2 cond's —

1) Three real distinct roots

2) one real, two complex roots







PROPERTIES

1) $P \boxplus Q = R$

2) $\theta \rightarrow$ point at infinity.

3) $P \boxplus (-P) = \theta \rightarrow$ identity element
$(x_1, y) \quad (x_1, -y)$

4) $P \boxplus \theta = P$.

5) $[P \boxplus Q] \boxplus R = P \boxplus [Q \boxplus R]$ → hard graphically fact.

group, ↓ commutative grp.

- $P \boxplus P = R \Rightarrow 2P = R$
  tangent at P
- $3P = 2P \boxplus P$
- $nP = (n-1)P \boxplus P$

(*) ⇒ Case 1: P & Q normal :- $(x_1 \neq x_2, y_1 \neq y_2)$

$y = mx + c$ ——— (a)

$m = \frac{y_2 - y_1}{x_2 - x_1}$

$y_1 = mx_1 + c$

$c = y_1 - mx_1, \quad c = y_2 - mx_2$

Eqn of the st. line at (a) will cut the curve (1)

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

$(mx+c)^2 = x^3 + ax + b$

$m^2x^2 + 2mxc + c^2 = x^3 + ax + b$

$x^3 - m^2x^2 + (a-2mc)x + (b-c)^2 = 0$

(x_1, y_1), (x_2, y_2) will satisfy eqn.

$x_1 + x_2 + x_3 = m^2$ → sum of roots $= -\frac{b}{a}$

$\boxed{x_3 = m^2 - x_1 - x_2}$

$m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{y_3 - y_1}{x_3 - x_1}$

$\boxed{y_3 = y_1 + m(x_3 - x_1)}$

$\boxed{P \oplus Q = R} \Rightarrow \boxed{R \rightarrow (x_3, -y_3)} \Rightarrow$ ans

(#) Case 2:- $x_1 = x_2$, $y_1 = -y_2$ (one below other)

$\boxed{P \oplus Q = O}$,

(#) Case 3:- $x_1 = x_2$; $y_1 = y_2$ (both pts. same)

$y = mx + c$

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

$2y\frac{dy}{dx} = 3x^2 + a$

$\frac{dy}{dx} = \frac{3x^2 + a}{2y}$

did this bcs in case 1 $m = \frac{0}{0}$ see q

Now, slope at (x_1, y_1) → $\left.\frac{dy}{dx}\right|_{(x_1, y_1)} = \frac{3x_1^2 + a}{2y_1} = m$

$c = y_1 - mx_1$

same as above $y^2 = x^3 + ax + b$

$(mx+c)^2 = x^3 + ax + b$

$\boxed{x_3 = m^2 - x_1 - x_2}$

$\boxed{y_3 = y_1 + m(x_3 - x_1)}$

$\boxed{R = (x_3, -y_3)}$

Made it discrete

Date .....

$E: y^2 = x^3 + ax + b$ $\quad 4a^3 + 27b^2 \neq 0$ $\Big\} \in R^2 = R \times R$

We will be considering the same curve in $Z_p \times Z_p$ where p is a prime number.

$y^2 = x^3 + ax + b$; $(x, y) \in Z_p \times Z_p$

& $a, b \in Z_p$

$4a^3 + 27b^2 \neq 0 \bmod p$ (everything mod p)

[CI] $x_3 = m^2 - x_1 - x_2 \in Z_p$ $\quad \{x_1 \neq x_2$ under mod p, $y_1 \neq y_2$ — " $\}$

$m = \frac{y_2 \ominus y_1}{x_2 - x_1}$ ⊖ → additive inverse $[+(p - y_1) = -y_1]$, $-$ → mul. inv.

$= (y_2 - y_1) \times (x_2 - x_1)^{-1} \bmod p$

$\therefore y_3 = y_1 + m(x_3 - x_1) \in Z_p$ [All under mod p]

(same for x_3)

② ECDH

E, P → Public

Curve, Point on E

Alice		Bob
Sec. K → a		b ← Sec K
Pub K → aP	aP →	bP ← Public K for Bob
for Alice	← bP	
a(bP)		b(aP)
abP	abP = baP (commutative, same meaning)	baP

[Shared Secret Key = abP].

~~The fact that~~ Security of ECDH relies on the fact that finding x from xP & P is comp. difficult.

→ This hard problem is known as Discrete log. problem on ECDH.

#

RSA Signature

RSA Enc [Enc : $C = x^e \bmod n$
Dec : $x = c^d \bmod n$

Signs [Sign: $S = x^d \bmod n$
Veri : $x = S^e \bmod n$

Date .....

ECDSA :-

$(E, P) \rightarrow$ Public

Secret Key : q

Public Key : qP

Alice - E C, G - Point (Base) on the curve there exists a large prime number n s.t. $nG = \mathcal{O}$

$(n+1)G \boxplus G = nG$

$(n \neq p)$ → p was curve like ye prime. n is here new prime.

$d_A \rightarrow$ Secret Key.

$Q_A \rightarrow d_A G \rightarrow$ Public Key.

$m \rightarrow$ message.

1) $e = \text{Hash}(m)$

2) $Z \rightarrow L_n$ leftmost bits of e when L_n is the bit length of n.

3) $K \rightarrow$ randomly from $[1, n-1]$

4) $(x_1, y_1) = K \cdot G$

5) $r = x_1 \bmod n$ if $r = 0$ then go to step (3)

6) $S = K^{-1}[Z + rd_A] \bmod n$ if $S = 0$ go to step (3)

7) Signature (r, S) on message → this tuple is Sign. by Alice

not pt. on curve.

inv of K is guaranteed bcs $\in [1, n]$

(B) Verification of ECDS performed by Bob.

1) Q_A is ~~equal~~ not equal to $\mathcal{O}$.

2) Q_A lies on the curve E_c or not

n known to Bob (Public) →

3) $nQ_A = n(d_A G)$
$= d_A(nG)$
$= \mathcal{O}$.

To verify G of Alice = G of Bob

1) verify $r, S \in [1, n-1]$ or not

2) $e = \text{Hash}(m)$

3) $Z \rightarrow L_n$ leftmost bits of e

4) $u_1 = Z \cdot S^{-1} \bmod n$
$u_2 = r \cdot S^{-1} \bmod n$

5) $(x_2, y_2) = u_1 G + u_2 Q_A$
if $(x_2, y_2) = \mathcal{O}$ then sign ~~valid~~ is invalid

6) if $r \equiv x_2 \bmod n$ then sign is valid otherwise invalid

PROOF

$C = u_1 G + u_2 Q_A \rightarrow$ Bob
$= u_1 G + u_2 d_A G$
$= (u_1 + u_2 d_A) G$
$= (Z \cdot S^{-1} + r S^{-1} d_A) G$
$= (Z + r d_A) S^{-1} G$

Spiral

$s = 2$

$(z + r \cdot dA) \cdot (k^{-1}(z + r dA))^{-1} G$

$= (z + r dA)(z + r \cdot dA)^{-1} KG$

$= KG = (x_1, y_1) \rightarrow$ Alice side.

$r \equiv x_2 \pmod n$

If x_1 matching then y_1 also matching ∵ point on curve.

(*) Discrete Logarithm Problem –

Given: finite cyclic group G of order n generator α of G element $\beta \in G$.

Find: Integer $x \ni 0 \le x \le n-1$ such that $\alpha^x = \beta$.

$(\mathbb{Z}_n, +) \rightarrow$ not hard bcs α^x is $x\alpha$

$x\alpha = \beta \Rightarrow x = \beta(\alpha^{-1}) \rightarrow$ Use extended eucl.

(Polynomial time complexity)

But $(\mathbb{Z}_n, *) \rightarrow$ hard.

(*) Exhaustive search $\rightarrow O(n) \rightarrow$ try α^x and check $= \beta$ or not for all x.

(*) Baby-Step Giant-Step Algo

$m = \lceil \sqrt{n} \rceil$; $\alpha^n = I$

If $\beta = \alpha^x$ then we can write

$x = im + j \rightarrow 0 \le i, j < m \ (\sqrt{n})$

$\alpha^x = \alpha^{im} \alpha^j$

$\beta = \alpha^{im} \alpha^j$

$\alpha^j = \beta (\alpha^{im})^{-1}$

$\alpha^j = \beta (\alpha^{-m})^i$

If you find i and $j \rightarrow$ can find x.

j and $\alpha^j \rightarrow$ Table

j to (0 to m)

Sort wrt α^j

Date .....

Now search $(\beta \alpha^{-m})$ in α^j value table.

you got i & j, now x.

Storage $= O(\sqrt{n})$

No. of mul $= O(\sqrt{n})$

Sort $= O(\sqrt{n} \log \sqrt{n})$

$= O(n^{1/2} \log n^{1/2})$ negligible.

Searching $= \log \sqrt{n}$ binary search replace

∴ Total $\sqrt{n}$

Algo from book

Example –

$G = \mathbb{Z}_{113}^*$ $\alpha = 3$

$|G| = 112$

$\beta = 57$ find $x \ni$

$3^x = 57$

$0 \le x \le 111$

$\approx \sqrt{112}$

① $m \leftarrow \lceil \sqrt{n} \rceil = 11$

② $(j, \alpha^j \bmod p)$ $0 \le j < 11$

j	0	1	2	3	4	5
$3^j \bmod 113$	1	3	9	27	81	17

6	7	8	9	10
51	40	7	21	63

Spiral

Date .....

Sorted Table-

$3^j \bmod 113$	j=0	1	8	2	5	9	3	7	6	10	4
	1	3	7	9	17	21	27	40	51	63	8

(3) $3^{-11} = ? \rightarrow (3^{-1})^{11} = (38)^{11} \bmod 113 = \boxed{58 = 3^{-11}}$

$3^{-1} \bmod 113 = 38$

↓

extended euclidean

(4) $Y = \beta \alpha^{-mi} \bmod 113$

$i = 0, 1, \ldots 10$

i	0	1	2	3	4	5	6	7	8	9	10
$Y = 57.(58)^i \bmod 113$	97	29	100	37	112	55	26	31	2	3	

GOT match

$\beta \cdot \alpha^{-9m} = 3 = \alpha^1$

$\beta \cdot \alpha^{-9 \times 11} = \alpha^{11}$

$\boxed{i=9, \; j=11}$

$x = im + j$

$= 9(11) + 1 = 100$

$\boxed{x = 100}$.

(*) ElGamal Public Key Cryptosystem-

1) Select a prime p
2) Consider the group $(Z_p^*, \cdot \bmod p)$
3) Select a primitive element

$\alpha \in Z_p^*$ → generator.

Z_p^* → Cyclic group

α can generate all

CRYPTO NOTES 2021/22

Date .....

4) Plaintext Space = Z_p^*

Key Space = $\{ (p, \alpha, a, \beta) \mid \beta = \alpha^a \bmod p \}$ ($\alpha \in Z_p^*$: cyclic; a → 1 to $p-1$)

↳ Public Key → p, α, β

↳ Secret key → a

$a = \log_\alpha \beta$ → Discrete log problem

5) $K = (p, \alpha, a, \beta)$

Select a random no. $x \in Z_{p-1}$, x is kept secret

Enc → by anyone.

Dec → only by authorized.

6) Enc:- $e_K(m, x) = (y_1, y_2)$

$y_1 = \alpha^x \bmod p$

$y_2 = m\beta^x \bmod p$

(x not shared.)

7) Dec:- $d_K(y_1, y_2) = y_2 (y_1^a)^{-1} \bmod p$

$y_1^a = (\alpha^x)^a \bmod p$

$= (\alpha^a)^x \bmod p$

$= \beta^x \bmod p$

$y_2 (y_1^a)^{-1} = (m\beta^x) \cdot (\beta^x)^{-1} \bmod p$

$= m \bmod p$

Jo jo Bob ko msg bhejna chahta hai wo ye p, α, β leke Bob ko msg bhejo, uske pass a hai, wo decrypt kar legi.

Ex Alice

$m = 5$

$x = 2$ → Random.

$y_1 = \alpha^x = 4^2 = 2$

$y_2 = m\beta^x = 5 \pmod 7$ hai

$(y_1, y_2) = (2, 5)$ →

Bob (public)

$p = 7$

$\alpha = 4$ → check karlo hora hai

$\beta = 4^3 = 1$

$a = 3$ → Secret

$(y_2)(y_1^a)^{-1}$

$= 5 \times (2^3)^{-1} = 5 \times (8)^{-1} = 5 \times (1)^{-1}$

$= 5 = m$

use ext. euc. algo to find inv

Kerberos (Version 4) : User authentication. → symm key

what is used for encr.

Date ..... is used for dec.

William Stallings Book

TGS : Ticket granting system
C : Client
AS : Authentication Server
V : Verifier.

client verified to TGS, AS, V

① $C \rightarrow AS : ID_c \| ID_{TGS} \| TS_1$

② $AS \rightarrow C : E(SK_c, [SK_{c,TGS} \| ID_{TGS} \| TS_2 \| Lifetime_2 \| Ticket_{TGS}])$

$Ticket_{TGS} = E(SK_{TGS}, [SK_{c,TGS} \| ID_c \| AD_c \| ID_{TGS} \| Lifetime_2])$

$SK_{c,TGS}$ → Session key b/w C & TGS.

SK_{TGS} → ~~Secret~~ key b/w TGS and AS

SK_c → " " C and AS

C is recieving

$Cipher_1 = E(SK_c, [SK_{c,TGS} \| ID_{TGS} \| TS_2 \| Lifetime_2 \| Ticket_{TGS}])$

$D(SK_c, Cipher_1) = [SK_{c,TGS} \| ID_{TGS} \| TS_2 \| Lifetime_2 \| Ticket_{TGS}]$

$Ticket_{TGS} = E(SK_{TGS}, [SK_{c,TGS} \| ID_c \| AD_c \| ID_{TGS} \| Lifetime_2])$

↳ can't be dec at client's side

③ $C \rightarrow TGS : ID_V \| Ticket_{TGS} \| Authenticator_c$

$Authenticator_c = E(SK_{c,TGS}, [ID_c \| AD_c \| TS_3])$

④ $TGS \rightarrow C : E(SK_{c,TGS}, [SK_{c,V} \| ID_V \| TS_4 \| Ticket_V])$

$Ticket_V = E(SK_V, [SK_{c,V} \| ID_c \| AD_c \| ID_V \| TS_4 \| Lifetime_4])$

$SK_{c,V}$ → ~~Secret~~ Session Key b/w Verifier and client → short time (has lifetime)

SK_V → Secret key b/w ~~client & V~~ V and TGS.

⑤ $C \rightarrow V : Ticket_V \| Authenticator_c$

$Authenticator_c = E(SK_{c,V}, [ID_c \| AD_c \| TS_5])$

⑥ $V \rightarrow C : E(SK_{c,V}, [TS_5 + 1])$ → bcs TS_5 was sent by client.

Spiral

Protocol :- Signal.org

: C = Enc(K, N, AD, M)

Sender —N, AP, C→ Reciever. (signal protocol)

AD : associated data

N : random number

K : Key (shared)

→ Forward Secrecy : If one key revealed then only 1 msg leak rest msgs are secure bcs every msg is encrypted using diff secret key.

---

① X3DH (Extended Triple DH)

② ECDSA

③ Double Ratchet

X25519 or X448 curve used

is $(2^{255} - 19)$ → prime no → some prime no.

① REGISTRATION

Alice generates the following packet P_A

P_A :

- IK_A : Identity Key $(n_1^A P, n_1^A)$ → long term
- SPK_A : Signed prekey $(n_2^A P, n_2^A)$ → weekly / monthly
- Prekey Sign : (Sign (IK_A, Encode(SPK_A)) → weekly / monthly — Sign on $n_2^A P$ using $n_1^A P$. Verification can be done using $n_1^A P$.
- Set of one-time prekeys : (OPK_A^1, OPK_A^2 ...) → optional / when server requests

Alice —Packet A→ Server ($n_1^A P$, $n_2^A P$, Sign)

Bob —Packet B→ Server ($n_1^B P$, $n_2^B P$, Sign)

This while user is doing registration first time.

Alice ← Packet B — Server (4)

Alice verify? sign of Packet B

Alice —— Server —— Bob

P_A P_B P_A P_B P_B

Alice generates ($n_3^A P$, n_3^A)

DH_1, DH_2, DH_3

SK_A = KDF ($DH_1 || DH_2 || DH_3$) ← SHA function.

AD = Encode (IK_A) || Encode (IK_B) → Acts as IVE

STUPY from Signal website and Sir's Slides

② Zero knowledge Proof using DLP :-

Prover — Verifier

Z_p^* → Group. Prover wants to prove the knowledge of x satisfying $y = g^x$ without revealing x. g → generator of group. p → prime. (SHARED)

Compute $y = g^x \bmod p$

Select r randomly from [0, p-1]

Compute $m = g^r \bmod p$

y →

m →

Chose e random [0, p-1]

← e

Compute $s = ex + r$

s → Checks $g^s \stackrel{?}{=} y^e m$

$s = ex + r$

$g^s = g^{ex+r} = g^{xe} g^r = y^e m$

if so, then ↑ Proved

Can't get x and r directly bcs DLP hard

⑧ RC4 Stream Cipher —

Array of S of N length ($N \approx 2^8$)

$S[i] = i$

$l \longrightarrow 5$ to 16

K = array → size = l bytes

$K[y] = k[y \bmod l]$

↓

Fixed secret key.

$y \to 0$ to $N-1$

$S[]$ → array → $S[0, \ldots N-1]$ } already filled with Scrambled permutation above array logic

$K[]$ → array → $K[0, \ldots N-1]$ }

for $i = 0$ to $N-1$ {

Initialize: $S[i] = i$, $j = 0$

}

for $i = 0$ to $N-1$ {

Scram-bling: $j = (j + S[i] + K[i]) \,(\%\, N)$

Swap ($S[i]$, $S[j]$)

}

→ Secret Key array

Key-Scheduling Algo

→ after scrambling,

$Pr[S[0] = 0] = 1/N$

$S[0, \ldots N-1]$ → Key dependent scrambled permutation array.

$i = j = 0$ → Initialization

$i = i + 1$

$j = j + S[i]$

Swap ($S[i]$, $S[j]$)

$t = S[i] + S[j]$

output $z = S[t]$

} output Keystream Generation Loop

$P = 1/N$ (∵ Pseudo random)

Key Generation Algo

$Z_2 = 0$

$Z_2 = 0$ & $S[2] = 0$; $Z_2 = 0$ & $S[2] \neq 0$

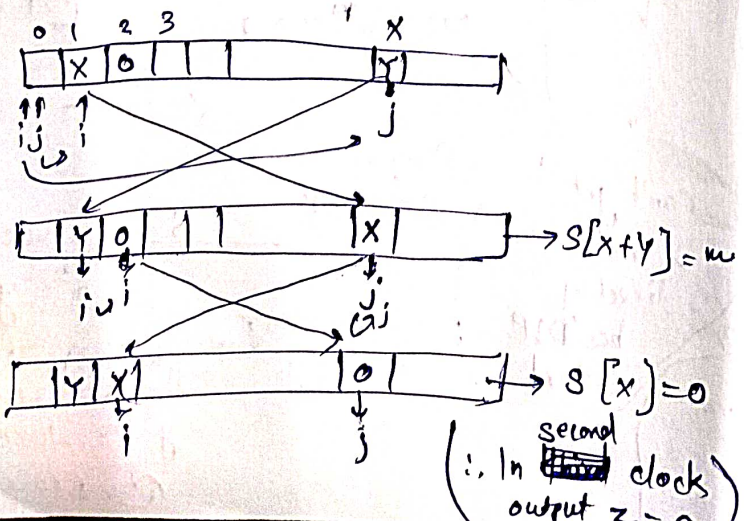
$P(Z_2 = 0) = P(Z_2 = 0 \mid S[2] = 0) \cdot P(S[2] = 0) + P(Z_2 = 0 \mid S[2] \neq 0) \cdot P(S[2] \neq 0)$

$= 1 \cdot 1/N + 1/N \cdot \frac{N-1}{N}$

$= 2/N$

$\therefore P(Z_2 = 0) = 2/N$

which is twice the expected prob.



only assumption if 2nd byte is 0 then 2nd output is 0

Result 1

⑨ SSL

(Secure Sockets Layer)

PDF — Sir shared

William Stallings