GOPI
Page No.
Date: / /

Elliptic Curve Cryptography

→ It is asymmetric / public key cryptography

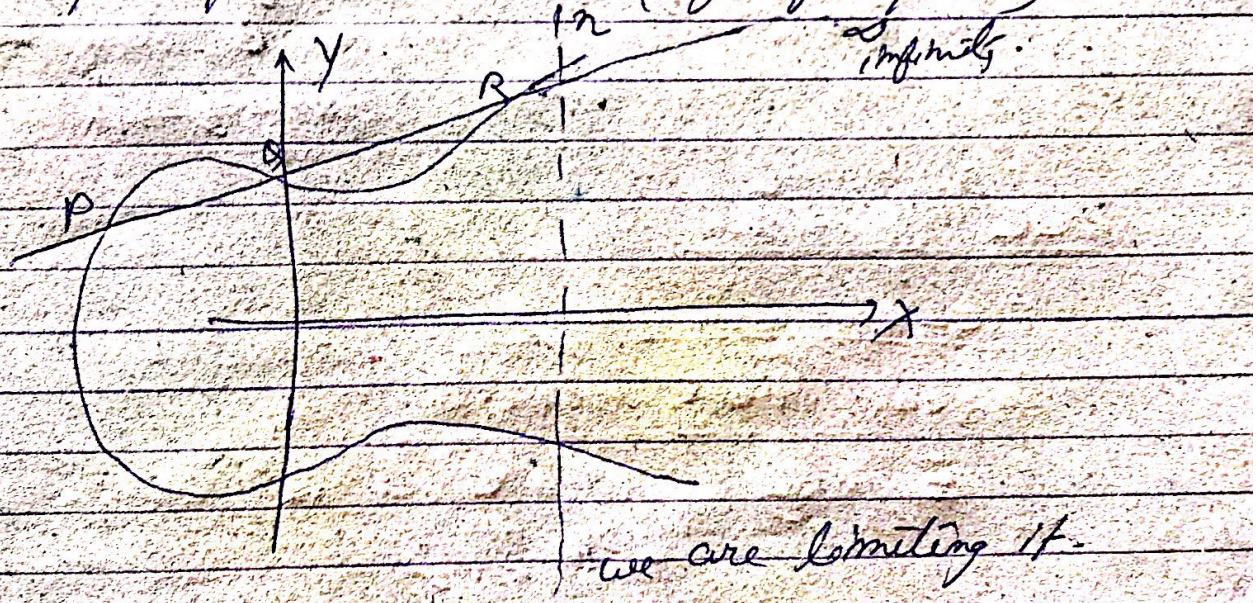
→ It provides equal security with smaller key size (eg: as compared to RSA) as compared to non-ECC algos.

i.e small key size & high security.

→ It makes use of Elliptic curves.

→ Elliptic curves are defined by cubic functions

eg: $y^2 = x^3 + ax + b$ (eqn of degree 3)



→ Symmetric to x-axis

→ If we draw a line, it will touch a max of 3 pts.

* A Trapdoor fn is a f^n that is easy to compute in one direction, yet difficult to compute in the opposite direction (finding its inverse) without special information called the trapdoor.

$(x) \xrightarrow{\text{easy}} \xleftarrow{\text{hard}} (f(x))$ easy if given 't' → trapdoor value

→ Let $E_p(a,b)$ be the elliptic curve

Consider the eqn $Q = kP$.

where $Q, P \rightarrow$ points on curve & $k < n$.

[If k & P are given, it should be easy to find Q but if we know Q & P, it should be extremely difficult to find.] → This is called the discrete logarithm problem for elliptic curves.

It is a one way f^n → Trapdoor f^n.

∵ $A \rightarrow B$ easy but

converse $B \rightarrow A$ is difficult.

ECC - Algorithm

ECC Key ~~Agreement~~ exchange

Global Public Elements

$E_q(a,b)$ → elliptic curve with parameters a, b & q

↓

prime no. or an integer of the form 2^m.

G: point on the curve / elliptic curve whose order is large value of n.

User A key generation

Select private key n_A $\quad n_A < n$

Calculate public key P_A $\quad \boxed{P_A = n_A \times G}$

GOPI
Page No.
Date: / /

User B key generation

Select private key n_B $\quad n_B < n$.

Calculate public key P_B $\quad \boxed{P_B = n_B \times G}$

Calculation of secret key by user A:

$$k_A = \boxed{k = n_A \times P_B}$$

Calculation of secret key by user B

$$\boxed{k = n_B \times P_A}$$

ECC Encryption

→ Let the message be M.
→ First encode this message M into a point on elliptic curve.
→ Let this point be $\boxed{P_M}$

Now this pt. is encrypted.

For encryption choose a random positive integer k

The cipher point will be

$$\boxed{C_M = \{kG, P_M + kP_B\}}$$ ← for encryption public key of B used.

This pt will be sent to the receiver.

Decryption

For decryption, multiply 1st point in the pair with receiver's secret key

i.e. $kG \times n_B$ // for decryption private key of B used.

Then subtract it from 2nd point / coordinate in the pair.

$$\left[\because P_M + kP_B - (kG + n_B)\right]$$

But we know $P_B = n_B * G$.

$$DO. = P_M + kP_B - kP_B$$

$$= \boxed{P_M} \text{ (original pt.)}$$

→ so receiver gets the same point.

ECC Based Scheme (size of n in bits)	RSA/DSA (modulus size in bits)
112	512
160	1024
256	3072
512	15360