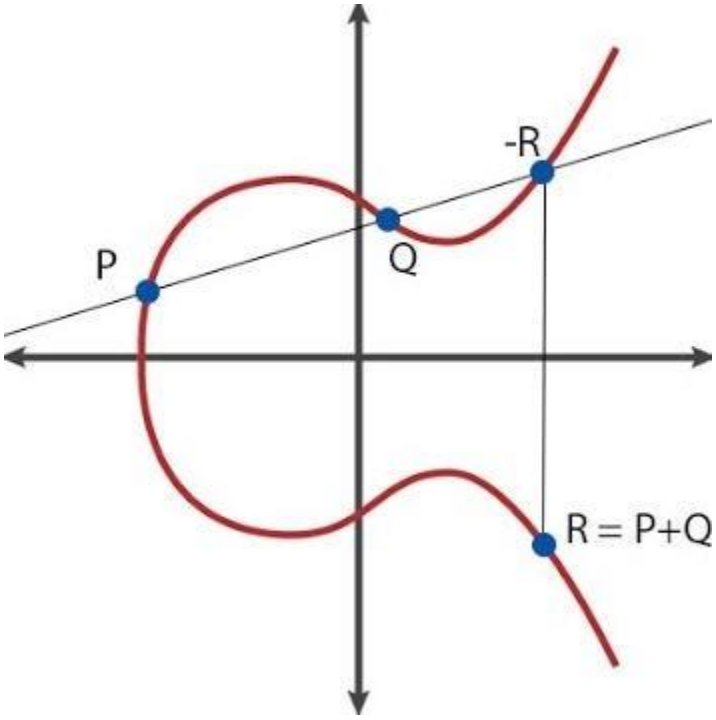
Elliptic curve cryptosystem

1. It is Asymmetri/public key cryptosystem.
2. It provides equal security with smallest key size (as compared to RSA) as compared to non ECC algo means key size & high security.
3. It makes use of Elliptic curves.
4. Elliptic curves are defined by some mathematical function->cubic function

e.g. $y^2=x^3+ax+b$ (eq of degree 3)(a,b constants)

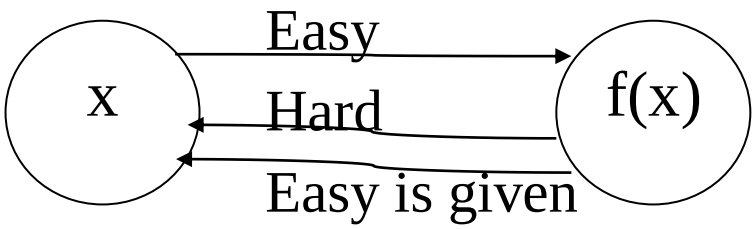


- Limit upto n
- Symmetric to x-axis.

- If we draw curve ,it will touch a max of 3 points.

A Trapdoor function:-

1.it is a function is easy to compute in one direction,yet difficult to compute in opposite direction(finding its inverse) without special info.,called the trapdoor.



"t" →trapdoor value

Let Ep(a,b)-elliptic curve

Consider the eq Q=KP

(Q,P are points on curve,K<n)

Note:-if K and P are given,not easy to find Q but if we know Q & P,it should be extremely difficult to find k. this is called the discrete log.

Problem for elliptic curve.

It is a one way function(Trapdoor fn)

A->B is easy but coming B to A is very difficult

ECC Algo:-

ECC key Exchange

1. **Global public elements:-**
   1. Eq(a,b) –elliptic curve with parameters (here a,b are constants
      q-prime no or integer of the form 2^m)

   2. Generator point(G):- points on the curve/elliptic curve whose order is large value of n

A----→B

Sender Receiver

User A key Generation:-

Select private key n_A then $n_A < n$

Calculate ,public key P_A then $P_A = n_A * G$

User B key Generation:-

Select private key n_B then $n_B < n$

Calculate ,public key P_B then $P_B=n_B*G$

Calculation of secret key by user A:-

$K_A=K=n_A*PB$

Calculation of secret key by user B:-

$K=n_B*P_A$

Encryption:-

1. Let the messagebe M
2. First encode this message m into a point on elliptic curve.
3. Let this point be P_m.(this point is encrypted)
4. For encryption ,choose a random positive integer K.
5. The cipher point will be
   $C_m=\{KG,P_m + K.P_B\}$ (public key of B used)

This point sent to receiver.

Decryption:-

1. For decryption multiply first point in the pair with receives secret key.

i.e. $KG*n_B$ (for decryption private key of B used)

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

i.e. $P_m+K.P_B-(KG*n_B)$ $[P_B=n_B*G]$

so $Pm+K.P_B-K.P_B$

so P_m (that is the original point)

so receives get the same point.

USES OF ECC:-

1. Elliptic curve cryptography encryption is one of the most generally used application techniques for digital signatures in various cryptocurrencies. Popular cryptocurrencies such as Bitcoin and Ethereum make use of the Elliptic Curve Digital Signature Algorithm (ECDSA key) particularly in signing transactions due to the security levels offered by ECC.
2. For digital signatures, ECC is applied in digital signatures through Elliptic Curve DSA (ECDSA

key) and in key exchange through Elliptic Curve Diffie-Hellman (ECDH). These algorithms are used in different parts of the SSL standard utilizing signing SSL certificates with ECDSA instead of RSA.

BENEFITS OF ECC:-

1.The usual ECC key size of 256-bit is equal to a 3072-bit RSA key, which is 10,000 times efficient than a 2048-bit RSA key. Therefore, to remain safe and to be ahead of a hacker's computing power, RSA keys must be long and requires keys that are 2048-bit or longer, which makes the process slower.

Elliptic Curve Cryptography vs RSA:-

The difference in size to security yield between RSA and ECC encryption keys is notable. The table below shows the sizes of keys needed to provide the same level of security. In other words, an elliptic curve cryptography key of 384 bit achieves the same level of security as an RSA of 7680 bit.

RSA Key Length (bit)
512

1024
2048
3072
7680
15360

ECC Key Length (bit)
112

160
224
256
384
512