***ELLIPTIC CURVE CRYPTOGRAPHY ALGORITHM***

***Computing the width-w NAF***

We use window-NAF technique to compute scaler multiplication that is required during the multiplication of the private key with the public key.

***INPUT:***

w where w is the window size

k where k is a positive integer in base-10

OUTPUT:

dn-1,…,d1,d0 where -2w-1 ≤ di < 2w-1 , di is in base-10,

and n is the length of k in base-2

STEP

i 🡨 0

while k ≥ 1 do

if (k modulo 2 = 1) then

ki 🡨 k modulo 2w

if (ki > 2w-1 – 1) then

ki 🡨 ki – 2w

end if

k 🡨 k - ki

else

ki 🡨 0

end if

k 🡨 k / 2

i 🡨 i + 1

end while

We will use this window for scaler multiplication. It is an efficient algorithm to compute scaler multiplication than other algorithms.

***Scaler multiplications operations with window-NAF technique***

***INPUT:***

P where P is a valid point in the curve (P belongs to E(Fq))

w where w is the window size

dn-1,…,d1,d0 where -2w-1 ≤ di < 2w-1 , di is in base-10,

and n is the length of k in base-2

***OUTPUT:***

R where R is a valid point in the curve

(R=[k].P)

STEP:

for all d in {1, 3, …, 2w-1-1} do

Ad 🡨 [d].P // pre-compute

end for

Initialize R with O (R 🡨 O)

// O is point at Infinity

for i = n-1 downto 0 do

d 🡨 |di|

R 🡨2R

if (di > 0) then

R 🡨 R + Ad

else if (di < 0) then

R 🡨 R - Ad

end if

end for

Return(R)

The algorithm uses deffi-hellman technique for key exchange. In this technique both the receiver and the sender chooses any number between 1 to n-1 where n is the prime order of G. G is the generator point on which the point operations will be performed. The sender and the receiver will generate their own key by using the private key that they has chosen. On performing the scaler multiplication on the Generator point using their private key, they will exchange it . They will multiply the exchanged key with their own private key and will generate the same value because the multiplication operation is commutative . Scaler multiplication is discrete logarithmic problem therefore the hacker will not be able to obtain the private key easily.

The first party will calculate

R=Pi\*Ps\*G

While the second party will calculate

R=Ps\*Pi\*G

Both R is the same.

The below algorithm is performing the key exchange.

***INPUT:***

Q where Q is the public key of other party

Pk where Pk is the private key of receiver

***OUTPUT:***

x where x is the shared key between two parties

STEP:

R🡨 [Pk].Q (The function that is written above is called here to compute scalar multiplication in ECC)

Return(Rx) (Rx is the key generated)

**We can use Elliptic curve cryptography algorithm for encryption of the data in VPN in place of RSA**. Elliptic curve cryptography has a discrete logarithmic problem which states that given a elliptic curve and a generator point G on the curve find the value of n where:

R=[n].G where R is the Resultant point after the point multiplication of n with G. It is very difficult to find out n.( more difficult to solve the integer factorization problem of RSA).

They both use asymmetric public key encryption technique. ECC uses smaller key size than RSA to provide the same level of security because the discrete logarithmic problem of ECC is harder. 256 bit ECC key is equal to 3072 bit keys in RSA. RSA security is based on integer factorization problem. This problem can be stated as:- Given num which the product of two prime number a and b, find a and b such that a.b=n. This problem is however much more easier than discrete logarithmic problem of ECC.

**Key Size (bits) Generation time (seconds)**

**ECC RSA ECC RSA**

163 1024 0.08 0.16

233 2240 0.18 7.47

283 3072 0.27 9.89

409 7680 0.64 133.90

571 15360 1.44 679.06

The above table represents the comparison between RSA encryption and ECC encryption. We can also find out that the generation time required by ECC is lower than RSA. Generation time is the time algorithm takes to generate the key.

A variety of attacks have been devised against Elliptic Curve Cryptosystems like Pollard rho. The safety of message depends on the type of elliptic curve that we use for encryption. To ensure best security, curves recommended by standard organizations like NIST should be used.

***ECC encryption***

***INPUT:***

Q where Q is the public key of other party

m where m is the intended message

k is the private key.

***OUTPUT:***

e where e is the encrypted message

C(x,y) where C is the chosen point

STEP:

Choose random value k [1,n-1]

C 🡨 [k].G

R 🡨 [k].Q

e 🡨 (Rx\*DECIMAL(m)) modulo p

Return (e,C)

Decimal(m) represents the ASCII value of the message. This ASCII value is encrypted by performing point multiplication with R.

***ECC Decryption***

***INPUT:***

e where e is the encrypted message

C where C is the chosen point

Pk where Pk is the private key of receiver

***OUTPUT:***

d where d is the decoded message

STEP:

R 🡨 [Pk].C

d 🡨 (e\*(Rx)-1) modulo p

Return(TEXT(d))

Therefore we can use Elliptic Curve cryptography algorithm in place of RSA for better encryption and efficiency.