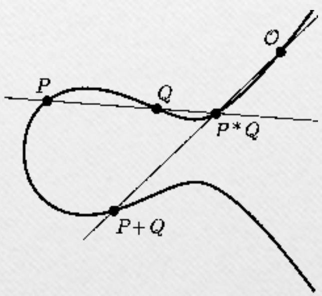
LOCATION BASED ELLIPTIC CURVE CRYPTOGRAPHY



By: Rupin Kejriwal

Abu Kaunen Nawaz

B.E (IT) 4th year

INTRODUCTION

- Location Based cryptography enhances security by integrating position and time into encryption and decryption processes.
- The cipher text can only be decrypted at a specified location.
- This can be used to ensure that data cannot be decrypted outside a particular facility, for example, the headquarters of a government agency or corporation, or an individual’s office or home.

ELLIPTIC CURVES

- Elliptic curves are cubic equations in two variables that are similar to the equations used to calculate the length of a curve in the circumference of an ellipse. The general equation of the elliptic curve is :

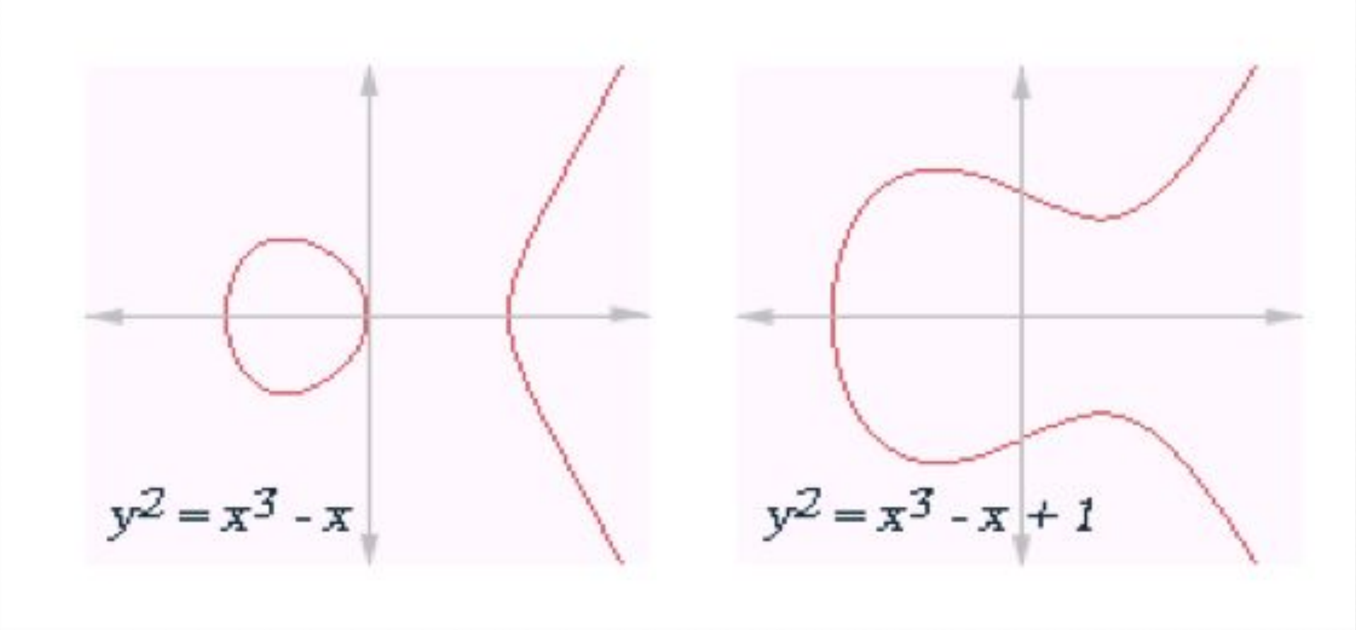
$$y^2 + b_1xy + b_2y = x^3 + a_1x^2 + a_2x + a_3$$

where a, b, x and y all belong to a field of say rational numbers, complex numbers, finite fields (Fp) or Galois Fields ($GF(2^n)$).

- Elliptic curves over real numbers use a special class of elliptic curves of the form :

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

Elliptic Curves (Examples)



Elliptic Curves Over GF(p)

- Since cryptography uses modular arithmetic, we use Galois fields.
- A Galois field, GF(p) is a finite field with p elements. This field can be the set Zp, {0,1,…,p-1}, with two arithmetic operations (addition & multiplication).
- In this set, each element has an additive inverse and non-zero elements have a multiplicative inverse.
- We define elliptic curve over a field as follows:

$$E(L) = \{\infty\} \cup \{(x,y) \in L \times L \mid y^2 + \ldots = x^3 + \ldots\}, \; L=GF(p)\text{: Galois Field}$$

Message mapping & reverse mapping in ECC

- Elliptic curve cryptography is used as a public-key cryptosystem for encryption and decryption in such a way that if one has to encrypt a message, they attempt to map the message to some distinct point on the elliptic curve by modifying the message using a mapping algorithm.
- In the receiver's end we need to get back the original message. To do this, a reverse mapping algorithm is used.

Location Key

- To implement location based ECC it is required that we form a location key which is another point on the elliptic curve.
- This location key is formed by the concatenation of location base point and the time key and then multiplying this concatenated value to the generator point on the curve.

Location Base point

- The map is divided into different grids of equal dimensions. Each grid has a base point marked with red dot. BP is located at the left bottom corner of each grid.
- Thus, we form the location BP. If the latitude is 22.789 and longitude is 88.126, then the coordinates are concatenated to form Location BP= 2278988126.

Time key

- Time Frame varies with time.
- For example, JU campus is open from 10:00:00 to 17:30:00 every day. So if the sender sends data on 04/05/2017 to the receiver, then the receiver will be able to decrypt that message if the receiver is physically located inside the campus & the receiver receives the message before 18:59:59 on that particular date.
- For each date, 10:00:00 is considered as the B.P of time frame. In this case the Time Key used by the sender is the concatenation date and BP of that particular frame = 04052017100000

Location Key

- Location Key = f(Time Key, Location BP)
- If we consider f() as concatenation then

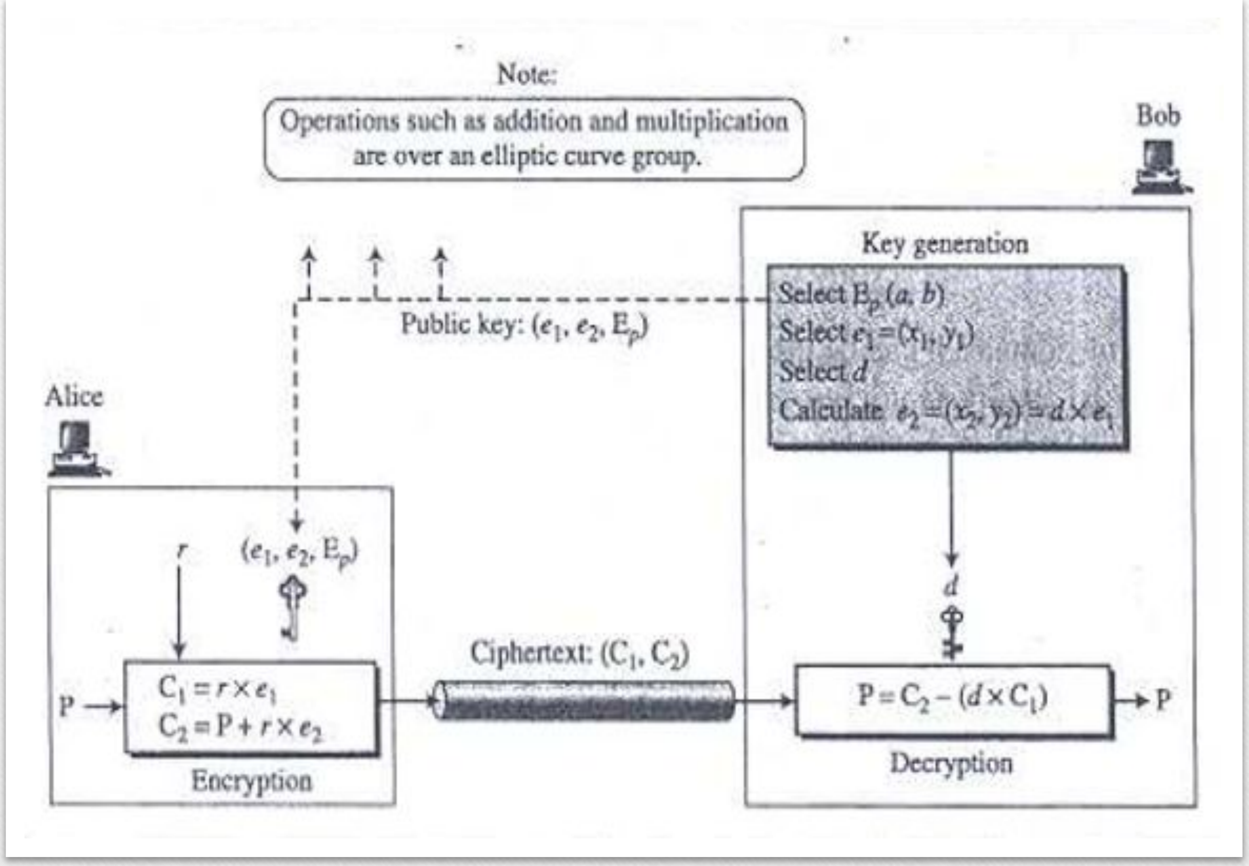
  Location Key= 040520171000000 || 2278988126

  L_k = 0405201710000002278988126

Elliptic Curve Cryptosystem

- This involves three steps :

1. Generating Public & Private keys.
2. Encryption by sender using location key & receiver's public key
3. Decryption by receiver using its own private key & its location key.


Note:
Operations such as addition and multiplication are over an elliptic curve group.
Bob
Key generation
Select $E_p(a, b)$
Select $e_1 = (x_1, y_1)$
Select d
Calculate $e_2 = (x_2, y_2) = d \times e_1$
Public key: (e_1, e_2, E_p)
Alice
r
(e_1, e_2, E_p)
P
$C_1 = r \times e_1$
$C_2 = P + r \times e_2$
Encryption
Ciphertext: (C_1, C_2)
d
$P = C_2 - (d \times C_1)$
P
Decryption

Generating Public & Secret keys

- Bob chooses **E_p(a,b)** with an elliptic curve **GF**(p).
- Bob chooses a point on the curve, $e_1(x_1, y_1)$.
- Bob chooses an integer **d**.
- Bob calculates $e_2(x_2, y_2) = d * e_1(x_1, y_1)$.
- Bob announces Ep(a,b), $e_1(x_1,y_1)$ and $e_2(x_2,y_2)$ as his public key; he kept d as his private or secret key.

Encryption

- Alice selects P, a point on the curve, as her plaintext, P (done by **message mapping algorithm**). She then calculates a pair of points on the text as cipher-texts :

  $C_1 = r * e_1$

  $C_2 = P + r * e_2 + L_k * e_1$ where r is a random integer.

- Alice sends these two cipher-texts to Bob.

Decryption

- Bob, after receiving C_1 and C_2, calculates P, the plaintext using the following formula :

  $P = C_2 - (d * C_1) - (L_k * e_1)$

  The minus sign here means adding with the inverse.
- This P is again a point on the elliptic curve which is converted into plain text message using the **reverse mapping algorithm**.

Security of ECC

- To decrypt the message, Eve needs to find the value of r or d.
- If Eve knows r, she can use **$P = C_2 – (r * e_2)$** to find the point P related to the plaintext. But to find r, Eve need to solve the equation $C_1 = r * e_1$. This means, given two points on the curve, C_1 and e_1, Eve must find the multiplier that creates C_1 starting from e_1. This is referred to as the *elliptic logarithm problem*, and the only method available to solve it is the Pollard rho algorithm, which is infeasible if r is large and p is large.

Security of ECC (Cont.)

- If Eve knows d, she can use **$P = C_2 – (d * C_1)$** to find the point P related to the plaintext. Because $e_2 = d * e_1$, this is the same type of problem.

- *The security of ECC depends on the difficulty of solving the elliptic curve logarithm problem.*

THANK YOU