LOCATION BASED ELLIPTIC CURVE CRYPTOGRAPHY

By: SAURABH RAJ (01)
KISHAN KUMAR (25)
B.E. I.T. (4^{th} YEAR)

INTRODUCTION

- Location based cryptography technique enhances level of security by integrating both position and time frame into encryption and decryption processes.
- The cipher text can only be decrypted at a specified location and time by the receiver.
- This can be used to ensure that data cannot be decrypted outside a particular surrounding area and after a certain time, for example, the headquarters of a government agency , or an individual's office or home.

THEORY USED

ELLIPTIC CURVES

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

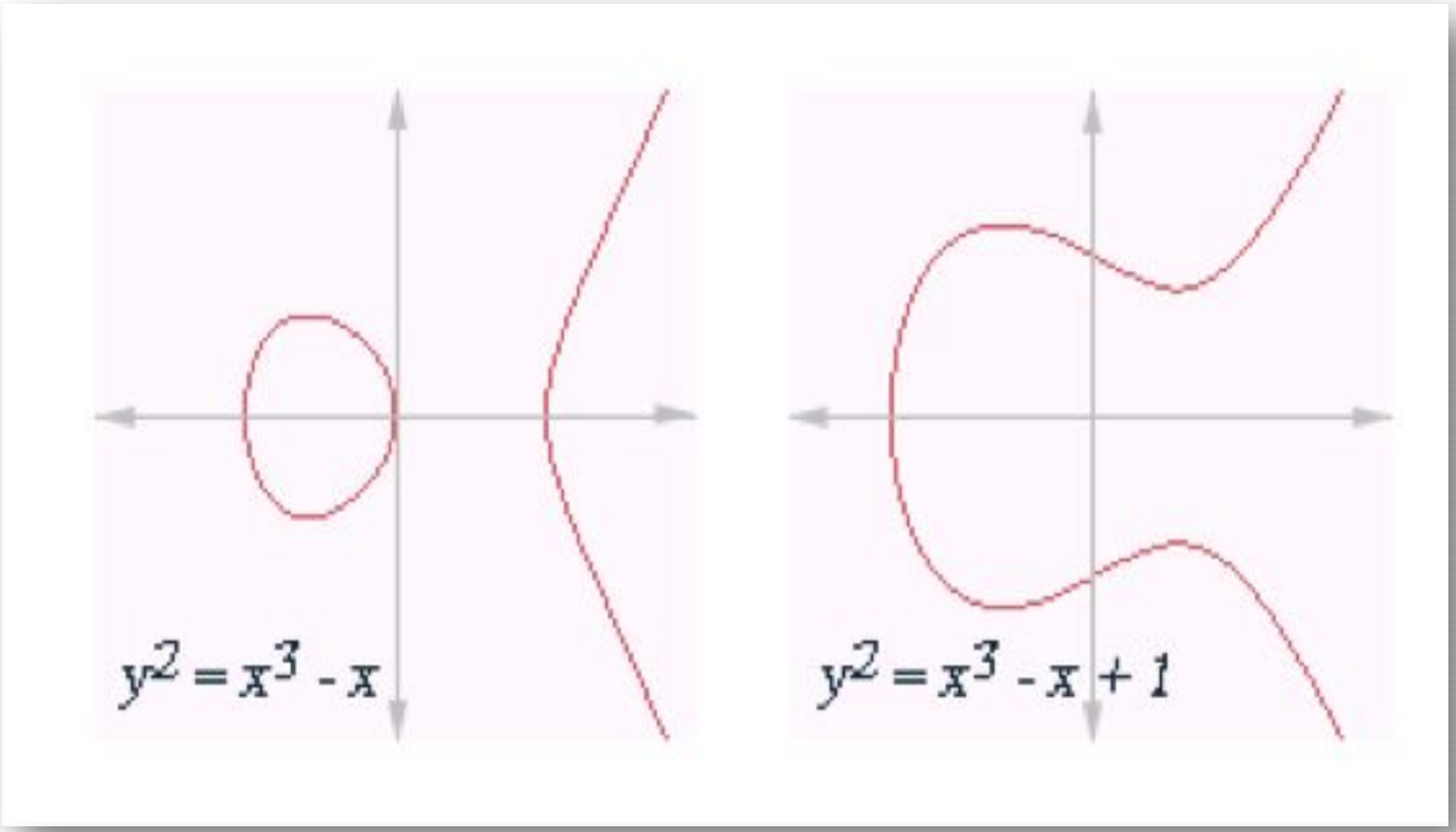
$$y^2 + b_1 x y + b_2 y = x^3 + a_1 x^2 + a_2 x + a_3$$

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 curve of the form:

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

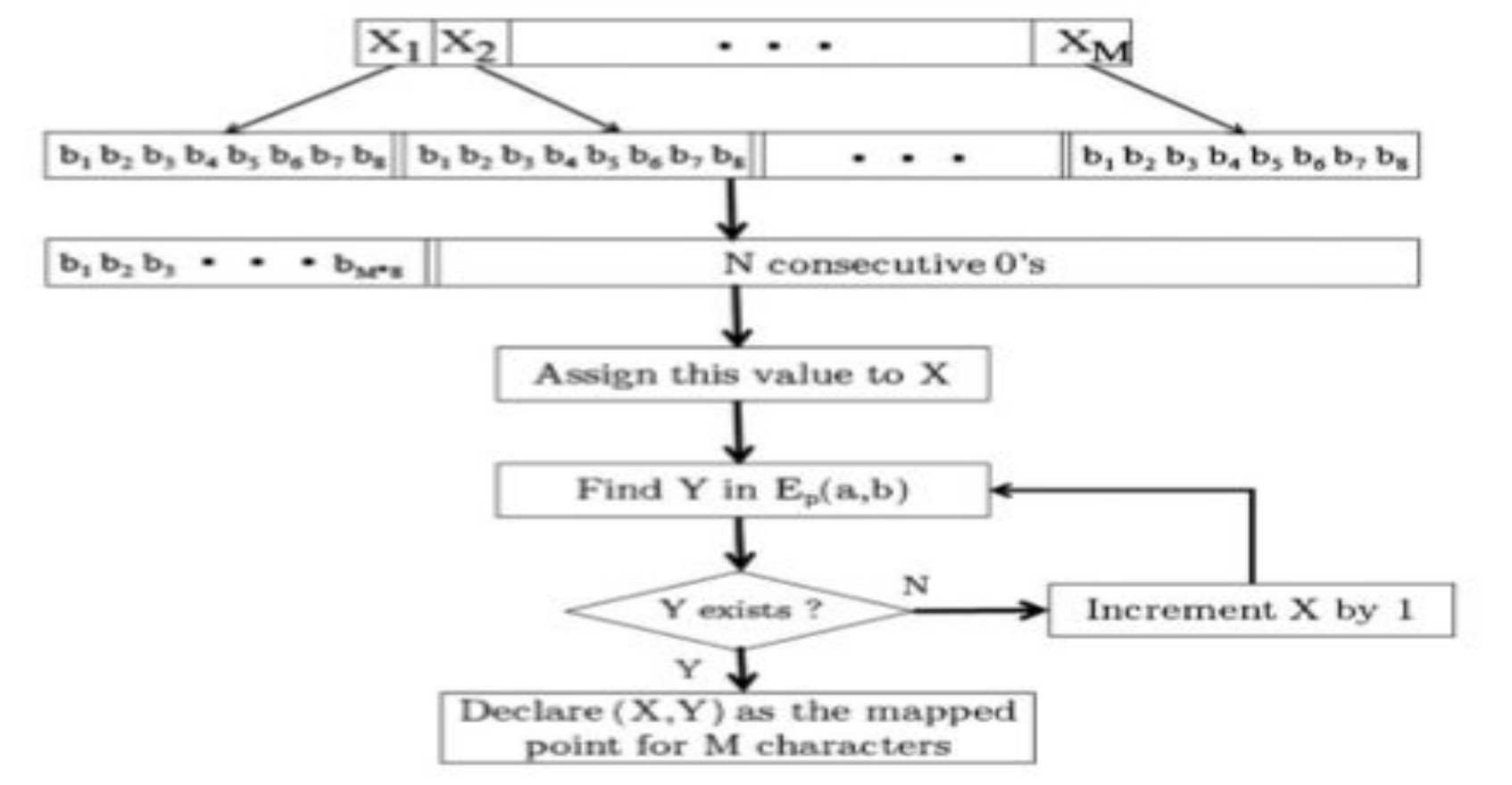
ELLIPTIC CURVES (Examples)



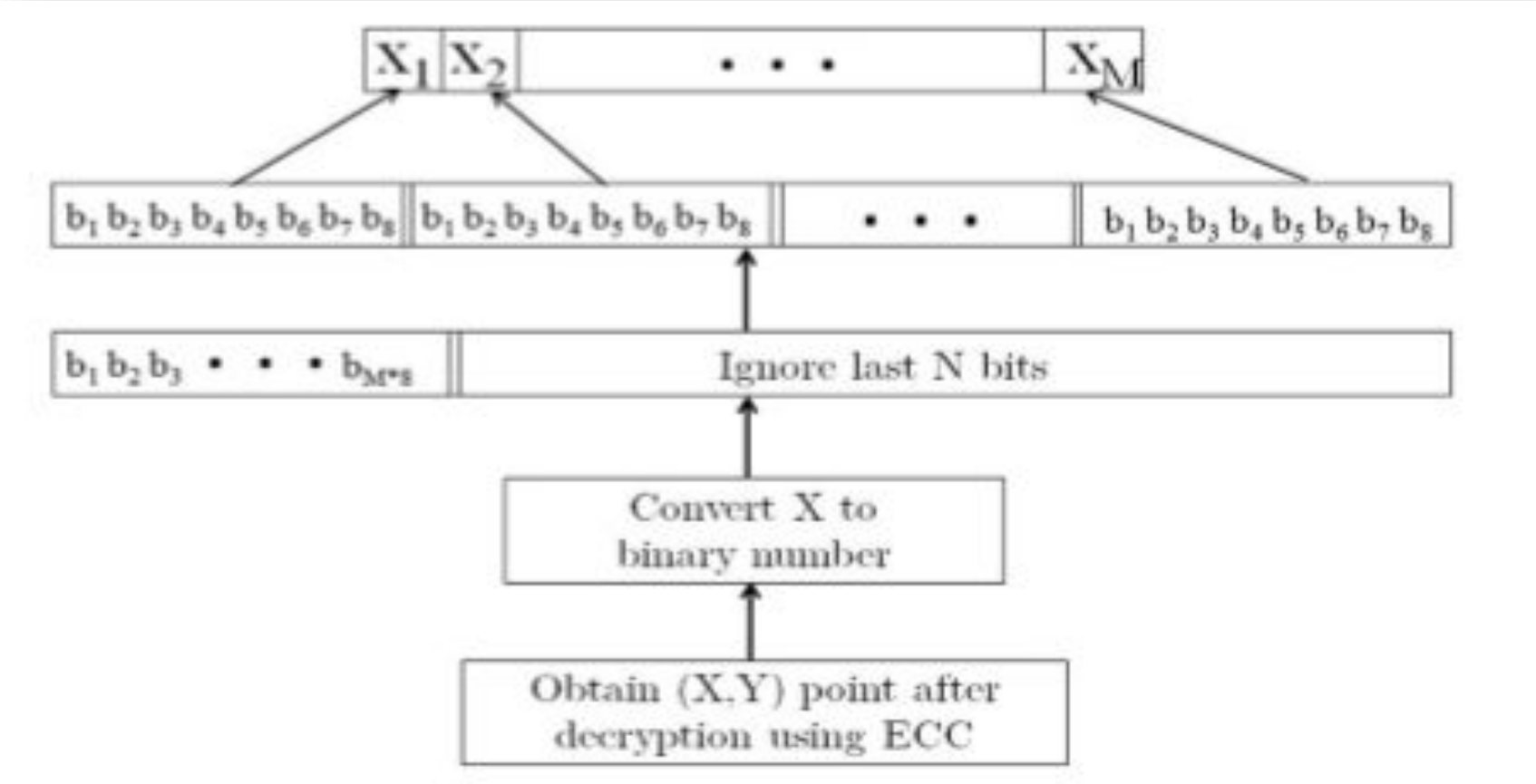
Message mapping and reverse mapping algorithms in ECC

- 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, he attempt to map the message to some distinct point on the elliptic curve by modifying the message using a mapping algorithm.
- At the receiver's end, it is needed to get back the original message using reverse mapping algorithm.

Mapping Algorithm:



Reverse Mapping Algorithm:



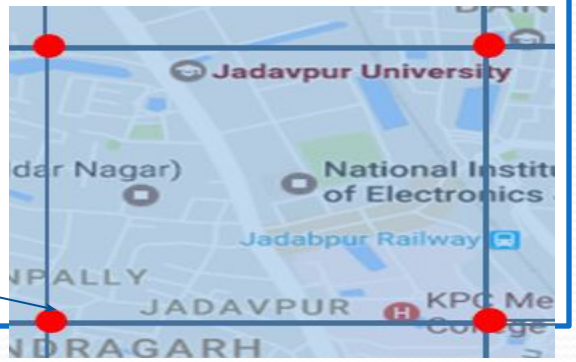
LOCATION KEY

- For implementation of Location based ECC, it is required to 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 time key and multiplying this concatenated value with the generator point on the elliptic curve.

LOCATION BASE POINT

- The map is divided into different grids of equal dimensions. Each grid has a base point marked with red dot. B.P. is located at the left-bottom corner of each grid.
- If the latitude is 22.789 and longitude is 44.126, then the coordinates are concatenated to form Location B.P. as Location BP = 2278944126.

BP of JU Main Campus



TIME KEY

- Time frame varies with time.
- For example, JU Main Campus is open from 10:00:00 to 17:30:00 everyday. So, if sender sends message on 12/01/2018 to the receiver, then the receiver will be able to decrypt that message only if he is physically located inside the campus and receives the message before 17:30:00 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 concatenation of date and B.P. of that particular frame given by 12012018100000.

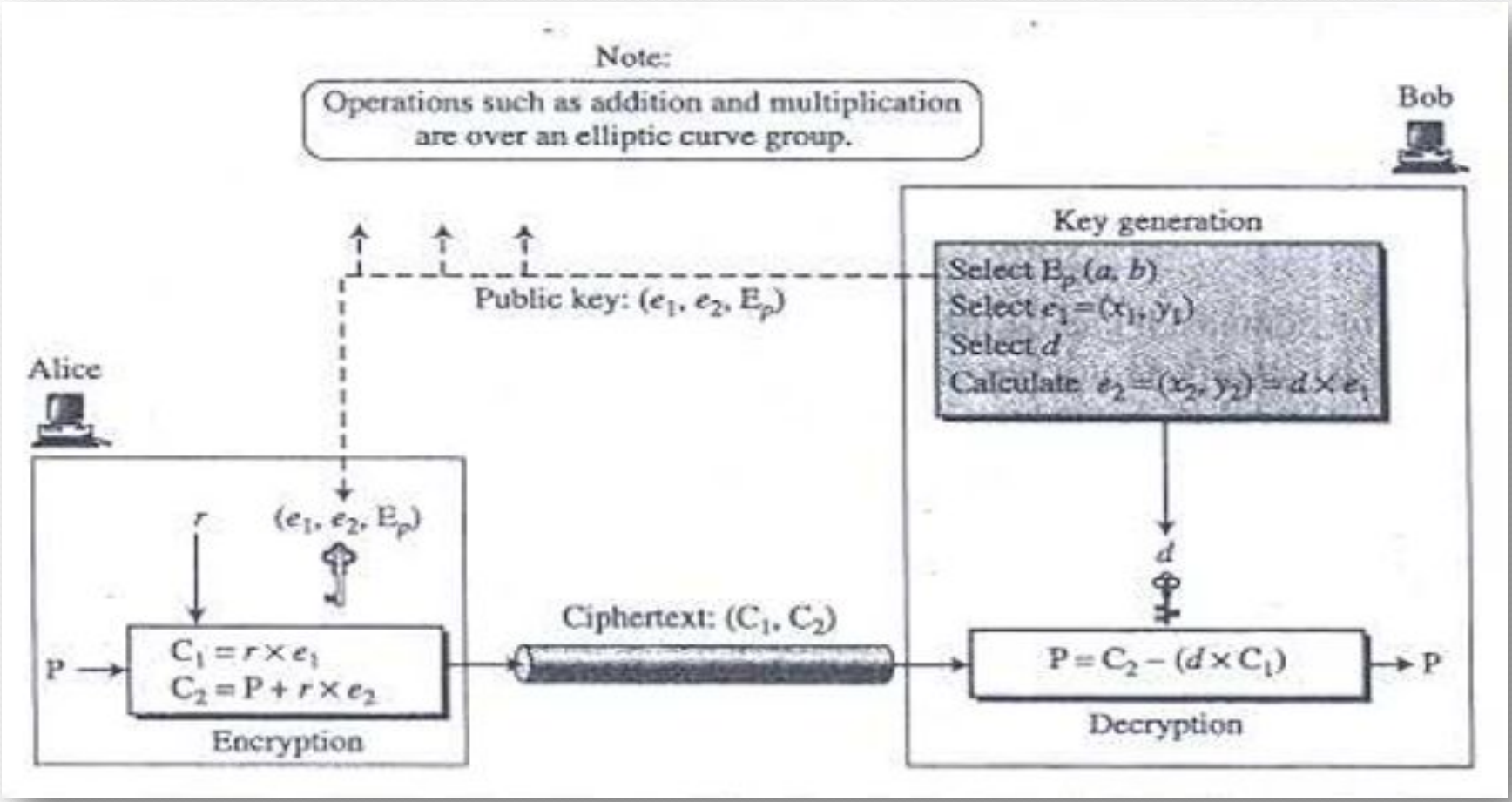
LOCATION KEY FORMATION:

- Location Key = f(Location BP, Time Key)
- Here f() is considered as concatenation operation then Location Key is given by:

  L_k = 2278944126 || 120120181000000

  i.e. L_k = 227894412612012018100000.

ELLIPTIC CURVE CRYPTOSYSTEM



Generation of Public & Private 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 point, $e_2(x_2,y_2) = d * e_1(x_1,y_1)$.
- Bob announces $E_p(a,b)$, $e_1(x_1,y_1)$ and $e_2(x_2,y_2)$ as his public key and d as his private key or secret key.

ENCRYPTION

- Alice selects a point P on the curve, as his plain-text message P (done by message mapping algorithm).
- He then calculates a pair of points on the curve 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 , C_1 and C_2 to Bob.

DECRYPTION

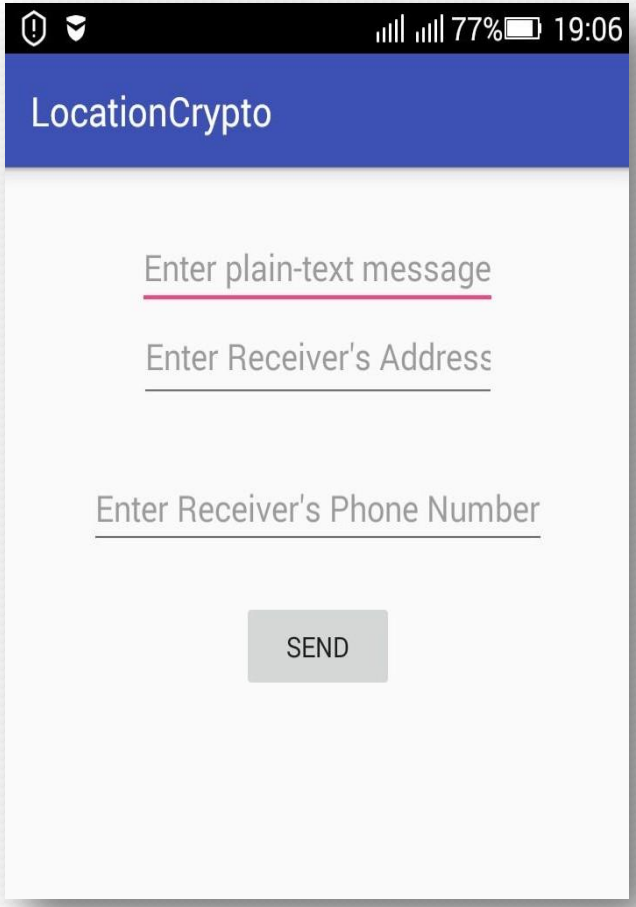
- Bob, after receiving C_1 and C_2, calculates plain-text message ,P using the following formula:

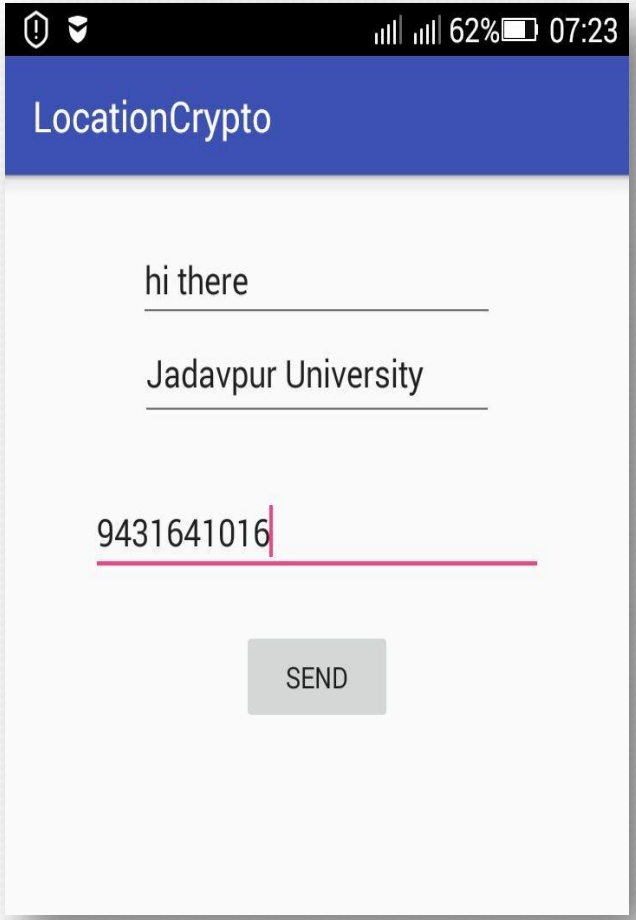
  $$P = C_2 - (d * C_1) - (L_k * e_1)$$

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

IMPLEMENTATION

ANDROID APP BASED





Decryption


77% 19:06
LocationCrypto
DECRYPTED MESSAGE
HELLO THERE!
LATTITUDE
32.5839°
LONGITUDE
88.3434°


77% 19:06
LocationCrypto
DECRYPT

Detecting Location of Receiver

SENDER'S END:

- The sender has to physically enter address at which the receiver is currently residing.
- The android.location.Geocoder classs of java is used and its method getFromLocationName(address,n) is called to get a list of all the n addresses close enough to this entered address.
- The methods getLatitude() and getLongitude() of Address class are used to get location coordinates of the receiver.

Detecting Location of Receiver (cont.)

RECEIVER'S END:

- The android.location.LocationManager class of java is used and its method getLastKnownLocation(LocationManager.GPS_PROVIDER) is called to get Location object.It means that this will access the GPS of receiver's mobile to get his location.
- The methods getLatitude() and getLongitude() of the Location class are used to get current location coordinates of the receiver.

TIME KEY GENERATION

- The java.util.Calendar class is used and its method getTime() is called to return a date object (of java.util.Date class).
- This date object is formatted using format(date) method of SimpleDateFormat class to bring it in the form “yyyyMMddHHmmss” .

MESSAGE SENDING

- After encrypting plain-text message, the sender has to then send the generated cipher-texts to the receiver.
- The android.telephone.SmsManager class of java is used and its method sendMultipartTextMessage(phoneNumber,SMSC address,text-message,pending-intent(SMS sent),pending-intent(SMS delivered)) is used to send cipher-text to the receiver.

MESSAGE RECEIVING

- BroadcastReceiver class is created to listen for any incoming message at the receiver's end.
- After that the phone number of the sender and the text message is fetched. This is done using methods like getOriginatingAddress() and getMessageBody() of SmsMessage class.
- The cipher text fetched is decrypted to obtain plain-text message, if possible.

SECURITY OF ECC

- To decrypt the message, Eve needs to find the value of r or d.
- If Eve knows r, he can use $P = C_2 - (r * e_2)$ to find the point P related to the plain-text. 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 and p have large values.

SECURITY OF ECC (cont.)

- If Eve knows d, he can use $P = C_2 - (d * C_1)$ to find the point P related to the plain-text. Because $e_2 = d * e_1$, this is the same type of problem.
- Hence, the security of ECC depends solely on the difficulty of solving the 'elliptic curve logarithmic problem'.

THANK YOU !!!