**Tools Utilised:**

In this project we will be looking to create 3 C++ files namely

1. Server.cpp- This file contains all the ECC algorithm and generation of OTP. It also contains the source code of creating a TCP socket and binding it. It is responsible for sending the message buffer to the client.
2. Client.cpp- This file contains all the ECC algorithm along with the listening socket for the client. When the server is sending some message to the socket, the client is listening the socket and message buffer from the server and decode the message and show the output.
3. FiniteFieldElement.hpp- This file is the header file. Since Elliptical curve cryptography is a part of the finite field. So, all the operations that happened in the ECC are the abelian group operations of the Finite Field Element. All the element operations are defined in this file along with the utils function and point representation of the ECC point.

First, we made a directory with the name “Preserved” and save all the files Server.cpp, Client.cpp and FiniteFieldElement.hpp in the folder.

=> Open up two terminal –

* + For Server
  + For Client

=> Move into the directory “Preserved” in both the terminal.

=> Enter the following command in terminals

* + g++ server.cpp -o server
  + g++ client.cpp -o client

The above command will compile both the server.cpp and client.cpp files and will make the execution file for them.

=> After that enter the following commands in the terminals

* + For server terminal - ./server
  + For client terminal- ./client

The above commands will start executing both the C++ files.

=> Enter the OTP in the server terminal.

=> Start the client terminal and start listening from the server socket.

=> The main working of the system is that the bank will enter the OTP. It will encrypt the OTP using its private key and shared key between the bank and the user and using the ECC algorithm. After that it will create a socket and will send the encrypted message to the client i.e. the user. After that the user will read the message and decrypt the message using its private key and shared key between the bank and the user. After that for the authentication the user has to enter the OTP in the ATM. If he entered wrong OTP authentication is failed else authentication success. Also, if an eavesdropper is trying to get the encrypted message, he will expose himself and the bank will be alerted as the data is being accessed by the unauthorized person.