Cryptography based Network Security Analysis using Secure Hashed Identity Message Authentication

Computer networks utilized to transfer data from one system to other system in the form of packets and while transferring hackers/attackers may attack packets to alter data and this alter data will be received at destination. To avoid such attacks many symmetric and asymmetric encryption algorithms were introduced which will allow sender to encrypt data before sending and if attacker alter the packets then it will not decrypted and steal data also cannot be decrypted by attacker without keys.

Encryption algorithms will provide security to network data but all those algorithms are very heavy in computation so author of this paper introducing Network encryption security with Message Authentication called SHIMA.

SHIMA will generate AERSA based prime number and then employ simple key substitution algorithm to encrypt message and then employ SHIMA message authentication algorithm using AERSA keys and message words. Authentication technique is a rule based technique which consist of 80 rule.

In rule1 will be applied if XOR operation between encoding word length and AERSA key is 0 to 19.

In rule2 will be applied if XOR operation between encoding word length and AERSA key is 20 to 39.

In rule3 will be applied if XOR operation between encoding word length and AERSA key is 40 to 59.

In rule4 will be applied if XOR operation between encoding word length and AERSA key is 60 to 79.

All the above rules will be encoded using predefined hash codes such as M1, M2, M3 and M4. All this codes are given in base paper. Destination will receive encrypted message and then apply reverse substitution technique to decrypt message and then apply SHIMA algorithm to generate Authentication code. If generate authentication and received code is same then message will be authenticated else authentication get failed.

Extension Algorithm

In propose SHIMA work Authentication code will be generated for each word by applying rules and XOR operations which will be heavy in computation and storage cost so as extension work instead of checking rules we have employed HMAC algorithm to generate Authentication code for SHIMA encrypted text. HMAC is light in computation and storage so we can save computation and storage cost.

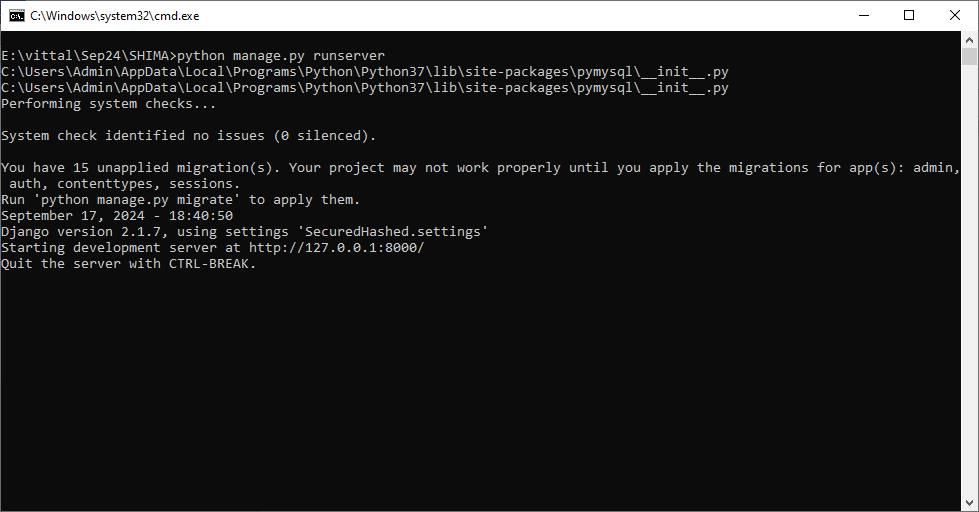
To implement above concept we have designed following modules

1. User Sign up: using this module new users can sign up with the application
2. User Login: after signup user can login to system
3. Upload File & Run SHIMA: using this module user can upload any text file which will get encrypted and authentication code will be generated using SHIMA and then compute storage and computation cost
4. Run SHIMA with HMAC: using this module file get encrypted using SHIMA encryption technique and then authentication will be generated using HMAC
5. Comparison Graph: using this module will plot storage cost graph between propose and modified algorithm
6. Download File: using this module user can view all his uploaded files and can download any file in decrypted format

SCREEN SHOTS

To run project install python 3.7.2 and then open console and copy all packages from requirements.txt file and then paste in console and press enter key to install all packages. Now install MYSQL database and then open MYSQL console and then copy content from ‘database.txt’ file and then paste in MYSQL console to create database.

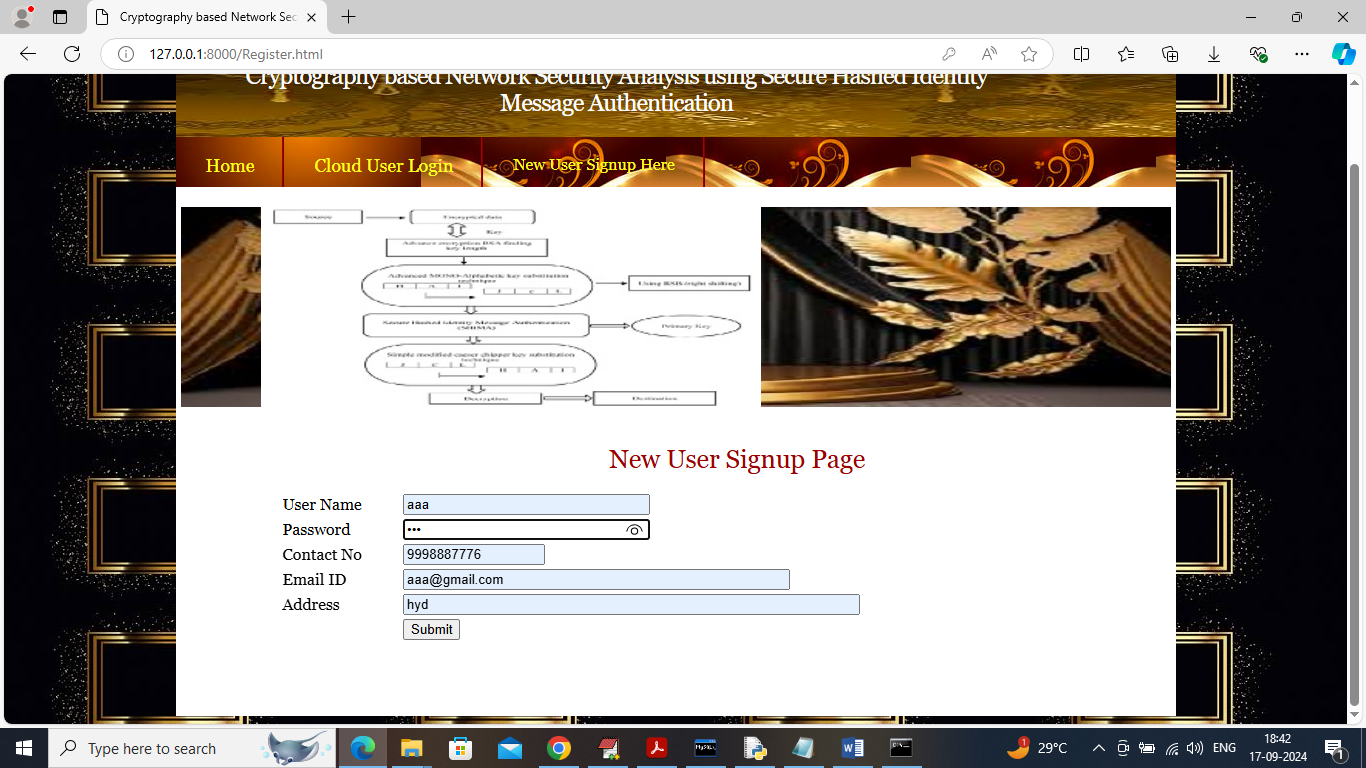
Now double click on ‘runServer.bat’ file to start python server and get below page



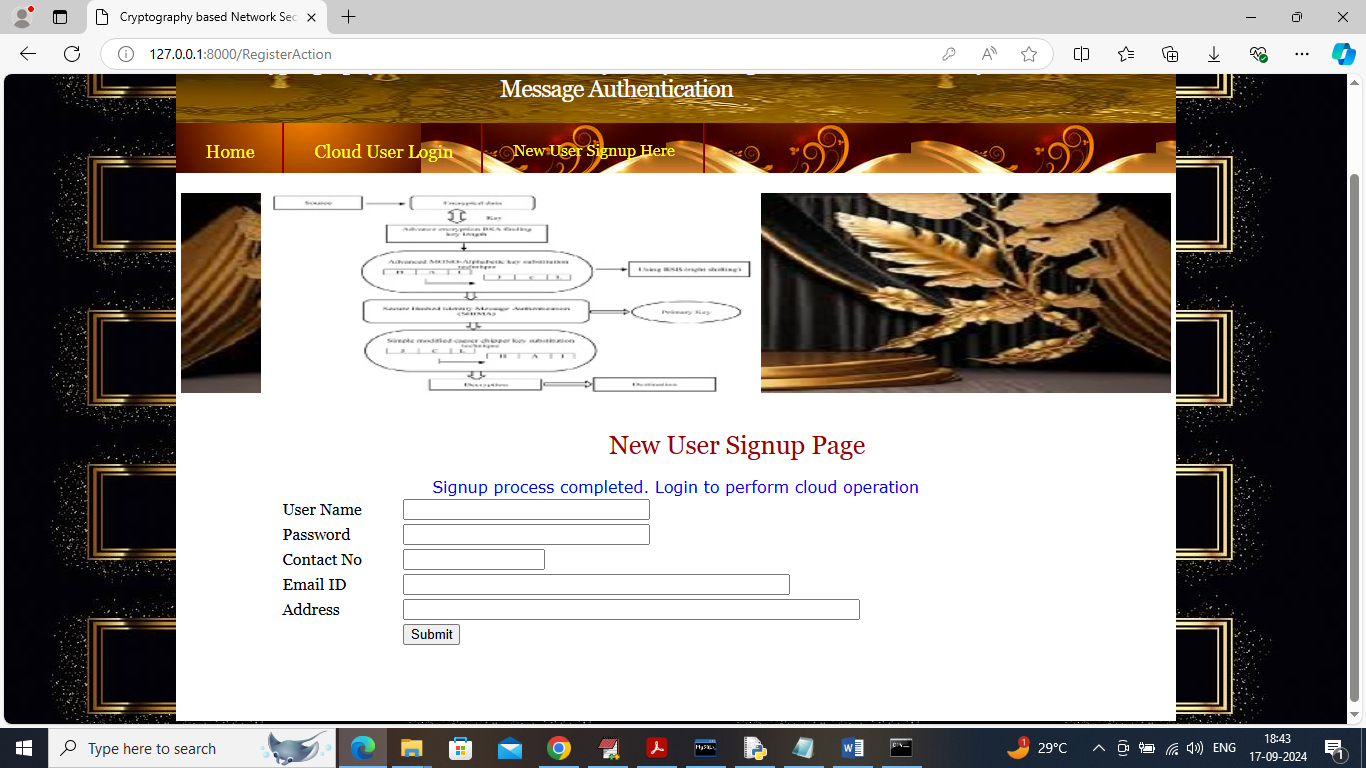
In above screen python server started and now open browser and enter URL as <http://127.0.0.1:8000/index.html> and press enter key to get below page



In above screen click on ‘New User Signup’ link to get below page



In above screen user is entering sign up details and then press button to get below page



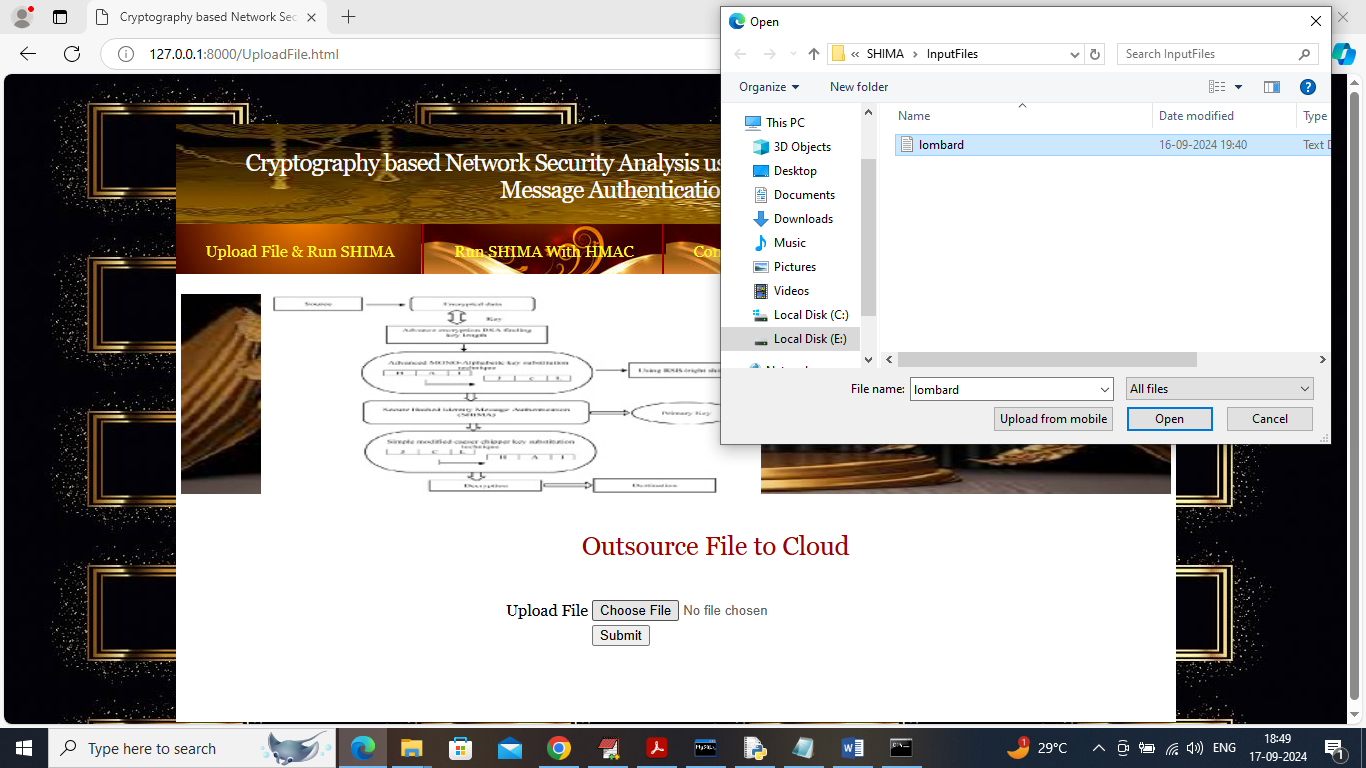
In above screen user sign up completed and now click on ‘User Login’ link to get below page



In above screen user is login and after login will get below page



In above screen click on ‘Upload File & Run SHIMA’ link to get below page



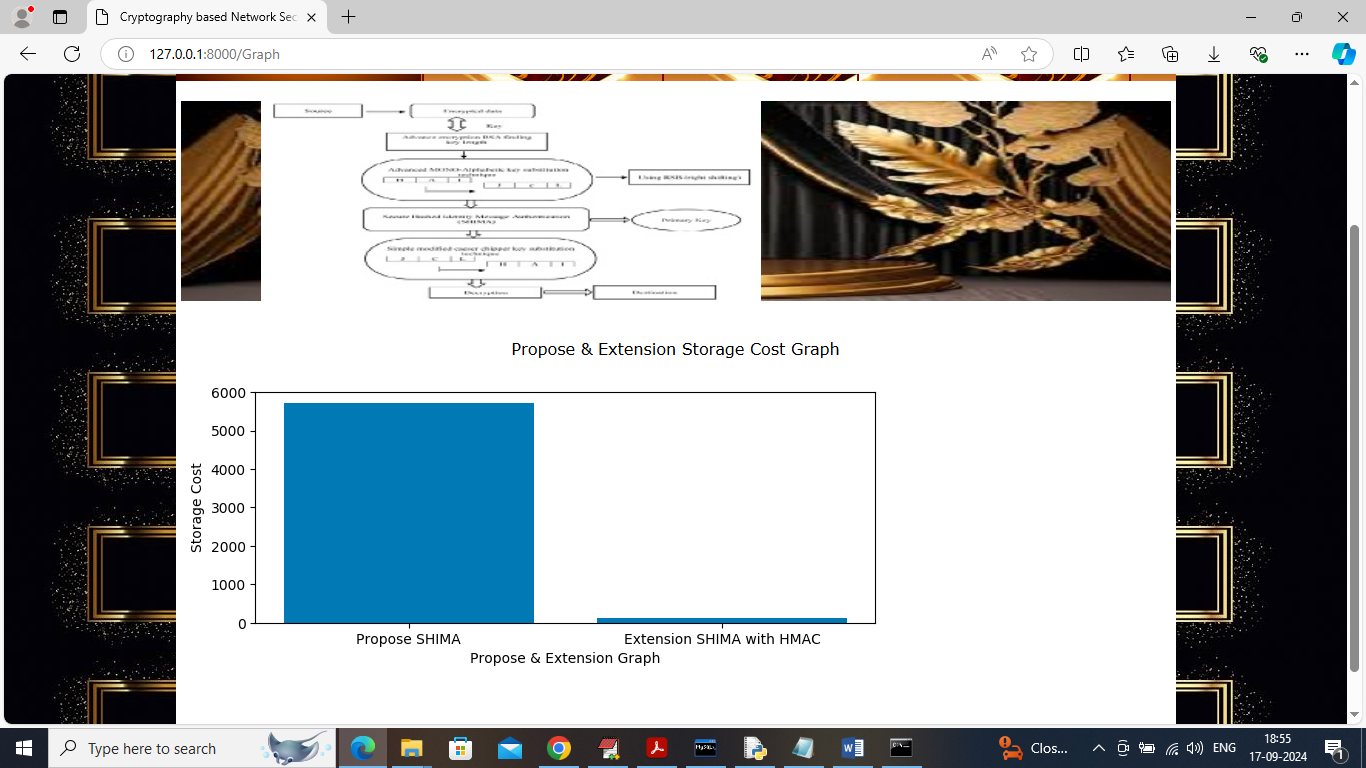
In above screen selecting and uploading input file and then click on ‘Open and submit’ button to encrypt file using propose SHIMA and extension SHIMA with HMAC and then compute storage and execution time and then will get below output



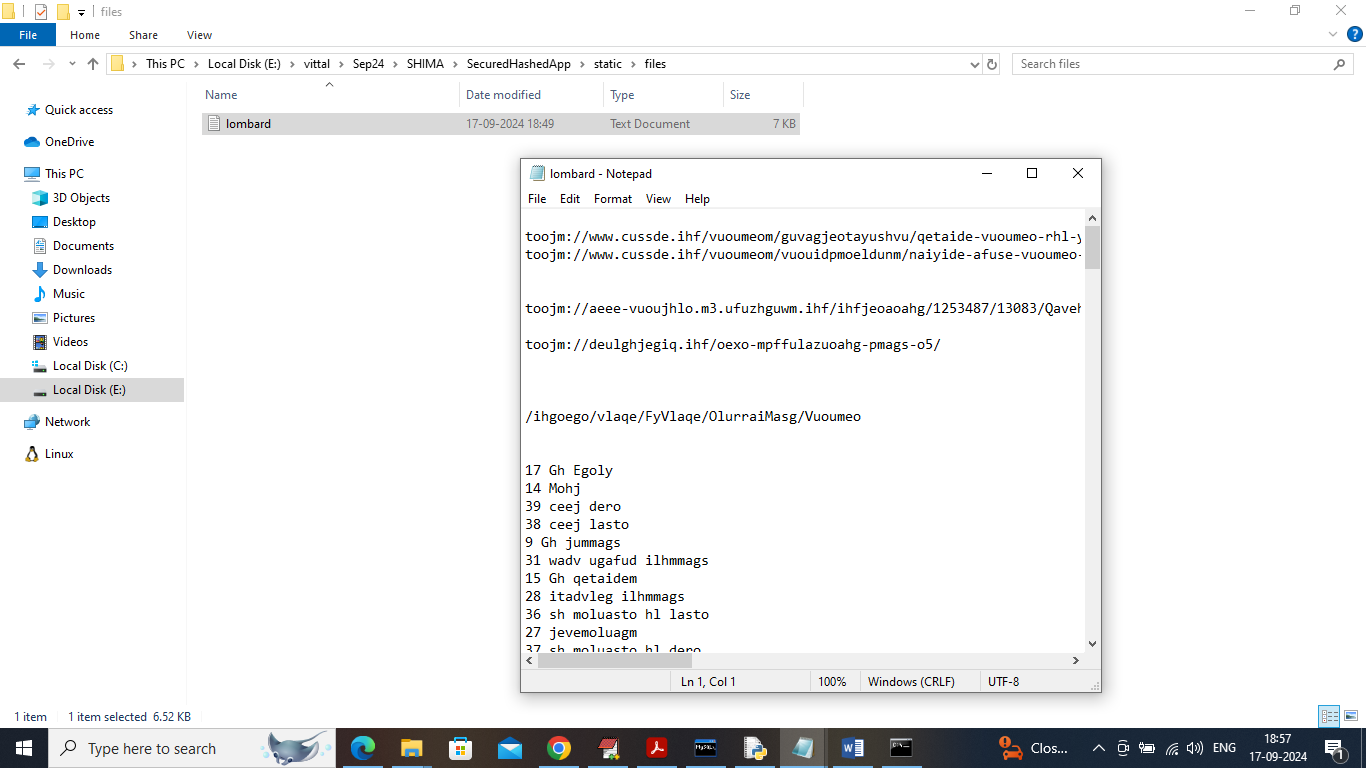
In above screen can see generated SHIMA hash code along with computation and storage cost in bytes and now click on ‘Run SHIMA with HMAC’ link to encrypt file with SHIMA and HMAC and get below output



In above screen can see file encrypted using SHIMA with HMAC and can see storage and computation time and then can see HMAC authentication code. HMAC will generate fixed size of authentication code. Now click on ‘Comparison Graph’ link to get below page



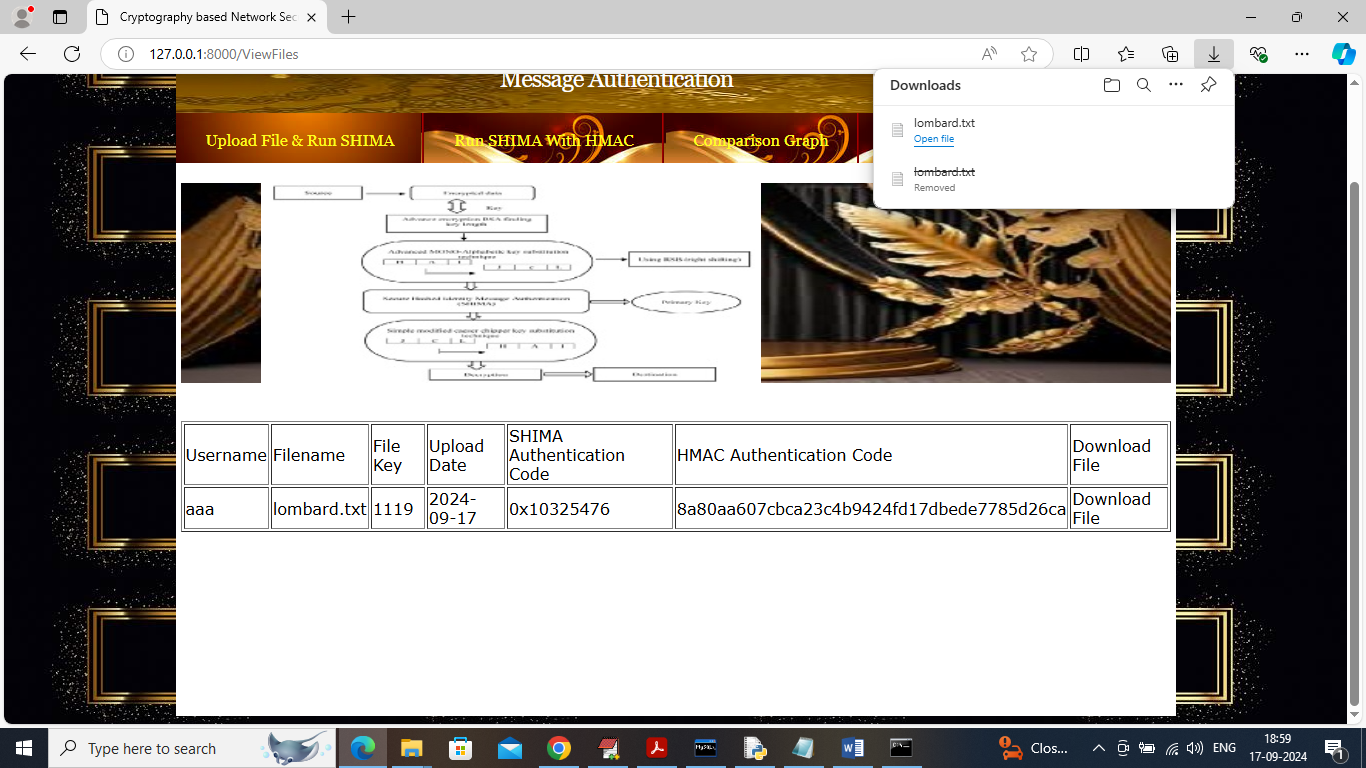
In above graph x-axis represents algorithm names and y-axis represents ‘Storage Cost’ and in both techniques SHIMA with HMAC got less storage. All the encrypted files you can see inside ‘SecuredHashedApp/static/files’ folder and in in below screen can see encrypted text



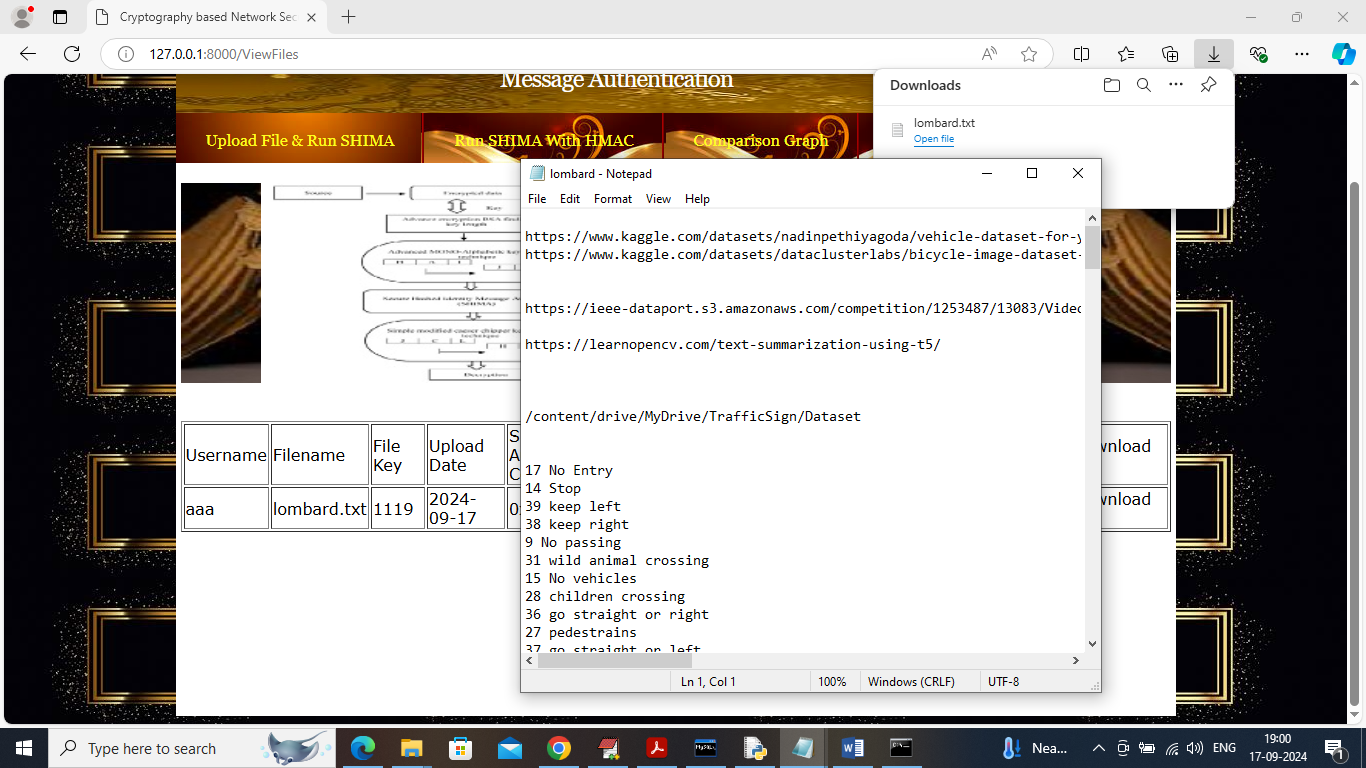
In above screen can see encrypted file store at server location and in above file we cannot read anything as all file data is in encrypted format. Now click on ‘Download File’ link to get below page



In above screen user can view list of all uploaded files along with SHIMA and HMAC authentication code and user can click on ‘Download File’ link to download file and get below page



In above screen in browser bar we can see file is downloaded and in below screen can see decrypted text.



In above screen file is decrypted and in readable mode.

Similarly by following above screens you can run entire code.