VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

CHAT CLIENT USING DIFFIE HELLMAN AND AES WITH KEY BASED DYNAMIC S-BOX

NAME : SUNDARAVELAN.
REG NO : 20BCT0248.

NAME : P.G.S.KOUSHIK.
REG NO : 19BCE0062.

NAME : A SREE HARSHA.
REG NO : 19BCE0221.

NAME : PRATEEK AGGARWAL.
REG NO : 20BCE0387.

NAME : VATTI DEEPAK.
REG NO : 20BCI0276.

NAME : SUDANSHU CHAUHAN.
REG NO : 20BCI0035.

A report submitted for the J component of

-- CSE1011- CRYPTOGRAPHY FUNDAMENTALS .

Supervisor

-- Madhu Viswanatham.

ABSTRACT

In this modern world everything is made available online (we are disclosing our data across the internet) so our data need to be safe and secure while sharing it using internet , so for ensuring that our data to be safe we are coming with a design of chat client using a hybrid cryptographic system in python with the help of AES (advanced encryption standard) and Diffie hellman algorithm for encryption. In this system Diffie hellman key exchange algorithm is utilized for transferring the AES key to the authorized or intended user. AES algorithm is used to encrypt the messages between the client and server after the key transfer. First the client transfers the AES key which is encrypted with the help of public key of the recipient(server) ,decrypt the AES key at the recipient (server) using it's private key using Diffie hellman key exchange algorithm .AES key which is used to encrypt and decrypt further messages.Diffie hellman key exchange and AES with 128 bit key. AES is implemented with dynamic Key based S Box. Using this method messages can be securely passed between the two users.

1. INTRODUCTION

In the current state of our world, encryption plays a critical role in data security. With more communication comes a greater understanding of the value of encryption. Encryption makes sense when data packets are sent across open channels that can be accessed by other devices or persons.

Encryption is knowledge of altering data with the key cipher by utilizing cipher algorithms that can export plain text from the ciphertext by someone with knowledge of the cipher key and chip algorithm. Encryption does not only conceal information, it also implies delivering information to a different form to guarantee data security.

An encryption system is made of transformations that turn plain text into encrypted text. Plain text is converted into blocks in the block cipher system, which is then subjected to a cipher algorithm to generate encrypted text.

Diffusion and Confusion are the two basic concepts that govern block cipher systems. Each bit of plain text turns into several bits according to the Diffusion Principle. However, the number of bits does not vary in the Confusion principle, and only modifications apply to plain text, therefore the size of plain text and ciphertext is the same in the Confusion principle. In both concepts, round repetition is used to generate encrypted text. Repeating a single round adds to the cipher's simplicity.

There are two basic kinds of cipher algorithms: private key algorithms and public key algorithms. Single key private key algorithms for encryption of plain text and decryption on sender and receiver side for the cipher. DES, 3DES, and Advanced Encryption Standard are private key algorithm examples (AES). S Boxes that are fixed and have no relation with a cipher key are determined by block cipher systems. Therefore, the cipher key only is changing the parameter. As the sole nonlinear component of AES is S Boxes, they constitute a significant cryptography source.

1.1. Theoretical Background

One of several standard models acknowledged for development of network applications is the Client Server Model. There is a client concept and a server idea in this architecture. As the name indicates, a server is a process (or a process computer) that offers various services to other entities called customers. A client, on the other hand, is a (running) process on the same computer or other machine requiring server services. But the server services in our project are little different, what happens here is The client is contacted by a server or chatted, vice versa.

A chat application is basically a combination of two applications :

- Server application .
- Client application .

The server app runs on the computer of the server while the client app runs on the client computer. A client can transmit data to the server and vice versa with this chat application.

The API provides classes for socket creation to make network programme communications easier. In this way, the Python API methods such as sockets form the endpoints of logical connections and may be utilised for transmitting and receiving data between two hosts. Python treats socket communications in a manner that is as easy to read or write to sockets as possible from input and output procedures.

A server connection must be formed and connected to a port where the server waits for connections in order to establish a server connection. For example, the client is connected to the port of 8002 at the Local-IP address (i.e. "local host") in our project and to the "local host" which is the result of communication between them.

1.2. Motivation

There are a lot of things nowadays that we can accomplish with the internet like delivering a simple message to an authorized person since we have been driven to choose this project out. And why did we consider the application Chat too, you could think? The development of internet technology helped individuals readily access the web. This internet offers more and more services Thanks to technology, all this can be virtualized. Communication among internet users is an integral aspect of their everyday lives. People have utilized the chat client system to connect to transmit messages. Here are the reasons why we have chosen Chat apps with great caution since messages are sent from server to client or client to server from some hackers due to security losses. In this project, we will examine key safety problems related to different storage formats and give a complete overview of security techniques, notably through the provision of confidentiality and data integrity.

We are BCI Students (Security), and we are delighted to make this procedure considerably safer and avoid intruders by offering privacy and data integrity.

1.3. Aim of the Proposed Work

The project is aimed at developing an encrypted conversation utilizing a python-based hybrid encryption system that uses AES and DIFFIE HELLMAN to encrypt and key exchange. The project begins with a quick description of how we are using the encryption and decryption RSA and AES algorithms. This section discusses how messages may be transmitted by connecting to the Local Host Port from client to server and server to client. The primary part of the project consists of a comprehensive breakdown and analysis of potential faults in our suggested design. The last section concludes how our solution addresses the issue of message hacking.

1.4. Objective(s) of the Proposed Work

(a) **ACCURACY:**

It is stated that if a message was conveyed without alteration, it would be correct. In this project, we want to build a chat customer with adequate security to prevent any individual other than the sender or receptor from altering the messages of a conversation.

(b) CONFIDENTIALITY:

Privacy means privacy or information confidentiality. It protects against unauthorized personal information. In our project secrecy, AES-key using a mathematical algorithm is encrypted. AES encrypts RSA and communications, and they are transmitted over sockets as text ciphers.

(c) INTEGRITY:

An integrity chat customer is stated to have only if that allowed-sender and recipient may view or modify messages.

(d) AUTHENTICATION:

Authentication ensures that the sender is who it is claimed that digital signature and public-key encryption may be maintained. We utilize an Diffie hellman key exchange technique in our project, a publicly available key cryptography, but we don't fully maintain Authentication if the key of the sender is anything that relates to it. The opposite means that AES encrypts the key with a private sender and a public and receiver key and decrypts the message with its private key and maintains public key senders that can improve our project.

(e) SPEED:

Messages should be delivered rapidly in any messaging program. Anyone using any texting app wants to send their messages fast. Messages are now sent within a fraction of a second in nearly all messaging systems. One of the aims of this project is to develop a chat customer for the message function at the pace of current messaging apps.

2 . LITERATURE SURVEY.

2.1. Survey of the Existing Models/Work

Literature Survey 1:

Hybrid Encryption/Decryption Technique Using Public And Symmetric Key Algorithm

Authors : Prakash Kuppuswamy , Saeed Q.Y. Al-Khalidi

Published on March 2018

Content :

A hybrid cryptosystem combines the characteristics of a publicly available cryptosystem with a symmetric key cryptosystem. Public key cryptosystem is convenient since the sender and recipient need not share a shared secret for safe communication (among other useful properties). However, they frequently require on complex maths and are often far more inefficient than equivalent symmetrical key cryptosystems. The high cost of encrypting lengthy communications in a public key crypto scheme might be prohibitive in many applications. This is dealt by utilising a mix of the two hybrid systems. A cryptosystem hybrid can be restricted.

- A key encapsulation scheme, which is a public-key cryptosystem, and
- A data encapsulation scheme, which is a symmetric-key cryptosystem

Literature Survey 2:

A Secure Electronic Messaging System in Client Server Cryptography-RSA Algorithm

Authors : G. Geeta Sai Sruthi , M. Raghupathi
Published on September 2019

Content :

Client/Server :

The client/server paradigm utilised outlines how a server serves one or more clients with resources and services. Examples include web servers, chat servers, and servers for files. Both servers supply customer devices with resources. Most servers feature one-to-many customer relationships, so that a

single server may offer computers with resources. Networks themselves become complicated numerous customers at once to satisfy the major needs of companies.

Chat service :

A secure chat service gives the opportunity for online, one-to-one or in-group interactions with users in real time. A public network slightly collects information, not on a single computer used to communicate with people. A secure chat between the client and the server, so that communication is secure and dependable

- Allows for instant communications between users.
- Uses real time chat over the network that can eliminate costly long distance charges.
- Allows for rapid query and rapid responses. While the negative points of chat service can be listed as following
- Security problems of instant messaging program.

Literature Survey 3:

A Survey on Secure Key Policy Attribute-Based Encryption Policy for Data Sharing Among Dynamic Groups in the Cloud

Authors : Soniya Thomas, Mr.J.Santhosh
Published on November 2018

Content :

RSA is safe because it factors big integers that are the product of two large numbers, according to Search Security. The key size is also huge, therefore improving the safety. The RSA keys usually have a length of 1024 and 2048 bits. The greater key size means, however, that it is slower than other encryption processes. While many more encryption methods are accessible, your private data is safe and is away from unwelcome eyes with the knowledge about and the most secure ways.

Literature Survey 4:

Using Cipher Key to Generate Dynamic S-Box in AES Cipher System

Authors : Razi Hosseinkhani, H. Haj Seyyed Javadi
Published on March 2012

Content :

We are attempting to introduce a new S-Box algorithm from the key chip. This method has been proven to produce new S-Boxes by altering only two pieces of chipset. We test the S-Box element at various intervals for this purpose. This technique will lead to secure block cyphers, address the problem of S-boxes in the fixed structure and enhance the AES block cypher system's safety level. The primary advantage of this technique is that you may use the Cipher key to produce multiple S-boxes.

Literature Survey 5:

Development of A Client / Server Cryptography-Based Secure Messaging System Using RSA Algorithm

Authors : Ferdi Sonmez , Mohammed Khudhair Abbas
Published on December 2017

Content :

Every day, individuals utilize a chat room, scan chat, or send messages to specific users through the users (clients). However, all customer information must be secured from hackers by the security

components in the chat area program. Without sufficient safety components, chat messages from users can be converted simply by skilled hackers. A strategy is required to safeguard a chat message from hackers via the chat area interface (CAI). It is important to maintain privacy in order to prevent unwanted access to private data. Basically, a communication channel between clients via a server with RSA encoding is intended to be provided by the proposed messaging/chat system inside the Client/Server context. The objective of this study was to utilize client/server architecture to secure chat between clients and not allow the server to decode the message using a single coding layer between the server and the clients and a second coding layer in the chat room between the clients The RSA algorithm is utilized for all the encryption procedures.

Literature Survey 6:

Scalable Virtual World Chat Client Server System

Authors : Dave Leahy , Judith Challinger , B. Thomas Adler, S. Mitra Ardon
Published on April 2001

Content :

A network of clients-servers is a network where one or many servers are connected via a communication channel to one or more clients. In general, an address is allocated to each server and client, so that every person may determine which Network communications are addressed. Although this system may just have one server, there are usually many customers. An object server is a request object: from and after the client object. Some services to answer the desire of the customer. A client is a requesting item. It is not set the designation of the particular item as a "server" object or a "client" object (ordinary hardware and/or software operation).

Literature Survey 7:

Client Server Chat Application

Authors : M. A. Mioc
Published on July 2016

Content :

There are two types of encryption used for messages: asymmetrical public-key encryption; private RSA key encryption and symmetrical AES encryption. The two modalities were employed since RSA is the most secure encryption method, but it doesn't work with lengthy communications. It can't be used. In order to transmit the messages both to the message recipient, AES encrypts the messages whereas RSA uses the encryption key and the AES VI.

Literature Survey 8:

Secure Data Encryption Through a Combination of AES, RSA and HMAC

Authors : Eman Salim, Ibrahim Harba
Published on March 2014

Content :

AES generally takes a fast speed and little RAM, which is extremely beneficial for data encryption, but since it's the same key for both encryption and decryption, a major difficulty exists in transferring key transfers from the sender side to the decryption side (receiver). For encryption key protection purposes, RSA is employed. We have created a stand alone RSA algorithm that produces both key algorithms instead of utilising a costly RSA supplier (private and public). The recipient has a private password and the sender has a public password. To encrypt the data, the sender uses this key. This is beneficial for attackers who are passive, but not active.

Literature Survey 9:

Design and Implementation of Client-Server Based Application using Socket Programming in a Distributed Computing Environment

Authors : Rolou Lyn R. Maata, Ronald Cordova, Balaji Sudramurthy, Alrence Halibas
Published on December 2017

Content :

In Java programming, connection classes between the client and the server are utilised. For this reason, Java offers two classes: socket and socket server. These ports link two programmes and create two programme sides (client and server). A user can execute customer software to generate a query via the client-server programme. The client connects and sends queries via socket to the specified server (client-side). Once the request is received, the server evaluates the request and transfers the result straight to the customer.

Literature Survey 10:

Implementation of Security Enhancement in AES by Inducting Dynamicity in AES S-Box

Authors : Gousia Nissar, Dinesh Kumar Garg, Burhan Ul Islam Khan
Published on August 2019

Content :
The dynamic S-Box put forward as recommended in the proposition in characterized as under :

1. According to the unique AES algorithm, Nr−1 round keys are driven from the cipher key.

2. In round j where 1 <= j < Nr a random string of 128 bits is created by utilizing the Kj as input to an arbitrary number generator (hash function).

3. This 128-bit string is then divided into 16-word, each word is of 8-bit length.

4. A state which consists of 16-word is what AES works on. In round j there are:
    a. A 16-word long random string which characterizes the state.
    b. A 16-word long random string.

5. If the word to be replaced is wji, the equivalent random word being Rji, the unique AES S Box being S, the unique AES inverse S-Box being IS, and the two are represented as one

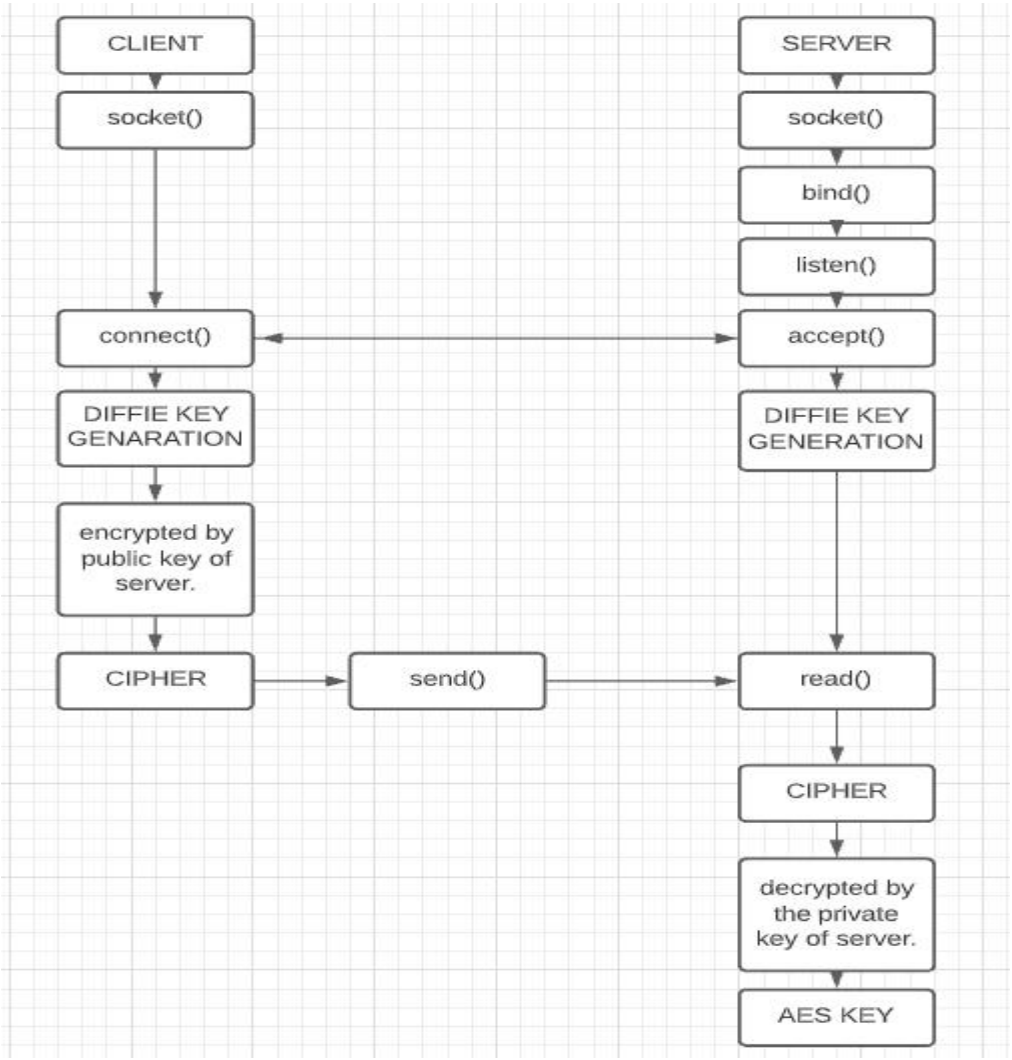
2.2. Summary/Gaps identified in the Survey

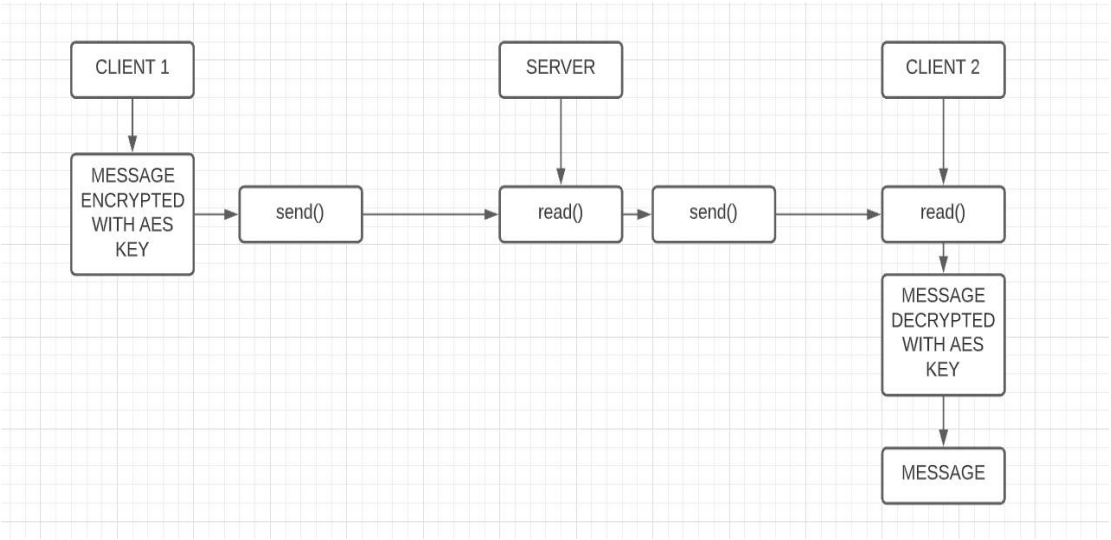
Every day, individuals utilize a chat room, scan chat, or send messages to specific users through the users (clients). However, all customer information must be secured from hackers by the security components in the chat area program. Without sufficient safety components, chat messages from users can be converted simply by skilled hackers. A strategy is required to safeguard a chat message from hackers via the chat area interface (CAI). It is important to maintain privacy in order to prevent unwanted access to private data.
In principle, a communication channel between clients via an AES-based encryption server is to be delivered by a suggested messaging/chat system in the client/ server context. Public key cryptosystems are handy since the sender and receiver do not have to share a shared secret to safely communicate.

We have created a novel S-Box dynamic algorithm from the AES cipher key. This method has been proven to produce new S-Boxes by altering only two pieces of the cipher key. This technique will result in a safe block cipher, address the problem of S-boxes, and enhance the AES block cipher system security level. The primary advantage of this technique is that you may use the Cipher key to produce multiple S-boxes.

3.Overview of the Proposed System :

3.1 Architecture/Model of Proposed System ：





3.2 Introduction and Related Concepts

We designed a secure chat client system for communicating in our design. If the communications being sent between the two parties are not encrypted at the sender's end and decoded at the receiver's end, there is no security. Because there is no protection, hackers may simply receive the communications and read them if they are not turned into encrypted text before transmitting. So, for encryption on the sender's side and decryption on the receiver's side, we employed the AES algorithm in our project. The AES key is utilised in the encryption and decryption process. As a result, both parties must be aware of the AES key. Communication takes place between the client and the server in our project. In this case, the client creates a random AES key. For encrypted communication, the server must now have this AES key. If this AES key is delivered directly from the client to the server, hackers can easily get to know the AES key and they can read the messages.

So, for transmitting AES keys from client to server, we employed the diffie hellman key exchange method in our project. The server generates the Diffie public and private key pairs. The client now transmits the AES key to the server using the server's diffie public. The AES key is now provided to the server as cypher text. Only the server's private key can retrieve the real AES key, and only the server knows its private key. As a result, no one (not even the client) has access to the server's diffie private key. After decrypting the cypher text given by the client using its diffie private key, the server now has the AES key. The AES key is now shared by both the server and the client (sender and receiver). As a result, they may now chat safely.

PROPOSITION FOR GENERATING NEW S-BOX DEPENDING OF MASTER KEY:

With the replacement matrix used in AES core as a starting point, it's time to find additional matrices that have the same or superior properties. The base is the main SBOX (shown in the diagram), which is employed in the cryptographic process.

The following technique computes (SBOXxor) the new plaintext substitution matrices based on the S-BOX proposed in AES (SBOX AES) and dependent on the key used to encrypt the new plaintext replacement matrices:
• First of all select one byte from master key (initial key) Key[i];
• Computting new SBOXxor, where each cell is equal to XOR with selected byte, SBOXxor[x,y] = SBOXAES[x,y] ⊕ Key[i];
• A substitution matrix newly calculated is used for plaintext encryption.

Decryption process the following approach is used:

• Select same byte from key - Key[i];
• Computing new SBOXxor, where each cell is equal to XOR with selecte byte, SBOXxor[x,y]=SBOXAES[x,y] ⊕ Key[i];
• Computing inverse matrices by using SBOXxor, SBOXxor INV=INV (SBOXxor)
• A recalculation inverse S-BOX is used to decrypt the plaintext.

By using the above described method 256 substitution matrices have been obtained. One of them SBOXAES ⊕ 0hex is equal to original SBOX suggered in AES core.

Encryption:

• Chose a key initial for AES;
• The first byte of the Key master is selected Key[1];
• Computing new SBOXkey[1] = SBOXAES ⊕ Key[1]
• Continue according to the algorithm set out in AES by using new calculated S-BOXkey[1].

Decryption:

• Chose a key initial for AES;
• The first byte of the Key is selected Key[1];
• Computing new SBOXkey[1] = SBOXAES ⊕ Key[1];

• Computing inverse SBOXkey[1]INV=INV (SBOXkey[1])=INV (SBOXAES ⊕ Key[1]);
• Continue as described in the AES algorithm with set out in AES by using new computed SBOXkey[1]INV.

Part of server code :

```
read_sockets, _, exception_sockets = select.select(sockets_list, [], sockets_list)


    # Iterate over notified sockets
    for notified_socket in read_sockets:

        # If notified socket is a server socket - new connection, accept it
        if notified_socket == server_socket:

            # Accept new connection
            # That gives us new socket - client socket, connected to this given client only,
it's unique for that client
            # The other returned object is ip/port set
            client_socket, client_address = server_socket.accept()

            # Client should send his name right away, receive it
            user = receive_message(client_socket)

            # If False - client disconnected before he sent his name
            if user is False:
                Continue

            # Add accepted socket to select.select() list
            sockets_list.append(client_socket)

            # Also save username and username header
            clients[client_socket] = user
            print('Accepted new connection from {}:{}, username: {}'.format(*client_address,
daes.decrypt(user['data'].decode('utf-8'),int_test_key)))

        # Else existing socket is sending a message
        else:

            # Receive message
            message = receive_message(notified_socket)

            # If False, client disconnected, cleanup
            if message is False:
                print('Closed connection from:
{}'.format(daes.decrypt(clients[notified_socket]['data'].decode('utf-8'),int_test_key)))

                # Remove from list for socket.socket()
                sockets_list.remove(notified_socket)

                # Remove from our list of users
                del clients[notified_socket]

                Continue

            # Get user by notified socket, so we will know who sent the message
            user = clients[notified_socket]

            print(f'Received message from {daes.decrypt(user["data"].decode("utf-
8"),int_test_key)}: {daes.decrypt(message["data"].decode("utf-8"),int_test_key)}')
```

Part of client code :

```
while True:

    # Wait for user to input a message
    message = input(f'{my_username} > ')
    message=daes.encrypt(message, int_test_key)

    # If message is not empty - send it
    if message:

        # Encode message to bytes, prepare header and convert to bytes, like for username
above, then send
        message = message.encode('utf-8')
        message_header = f"{len(message):<{HEADER_LENGTH}}".encode('utf-8')
        client_socket.send(message_header + message)

    try:
        # Now we want to loop over received messages (there might be more than one) and print
them
        while True:

            # Receive our "header" containing username length, it's size is defined and
constant
            username_header = client_socket.recv(HEADER_LENGTH)

            # If we received no data, server gracefully closed a connection, for example using
socket.close() or socket.shutdown(socket.SHUT_RDWR)
            if not len(username_header):
                print('Connection closed by the server')
                sys.exit()

            # Convert header to int value
            username_length = int(username_header.decode('utf-8').strip())

            # Receive and decode username
            username = daes.decrypt(client_socket.recv(username_length).decode('utf-
8'),int_test_key)

            # Now do the same for message (as we received username, we received whole message,
there's no need to check if it has any length)
            message_header = client_socket.recv(HEADER_LENGTH)
            message_length = int(message_header.decode('utf-8').strip())

            message = (daes.decrypt(client_socket.recv(message_length).decode('utf-
8'),int_test_key))

            # Print message
            print(f'{username} > {message}')
```

Part of dynamic s box code :

```
def sboxRound(key, newSbox):
    """ Runs a round of the S-Box block round."""
    shiftCount = getShift(key)
    usedRow = list(range(16))
    usedColumn = list(range(16))
    for i in key:
            coord = getIndex(i, usedRow, usedColumn)
            shiftRow(coord[0], shiftCount, newSbox)
            shiftColumn(coord[1], shiftCount, newSbox)
            swap(coord, newSbox)
```

OUTPUT SCREEN SHOTS :

One client sending the message to the other :

CLIENT 1 :

```
PS D:\PROJECT\proj1>
PS D:\PROJECT\proj1> python -u "d:\PROJECT\proj3\client1.py"
Username: user1
user1 >
user2 > hello
user1 >
```

CLIENT2 :

```
PS D:\PROJECT\proj2> python -u "d:\PROJECT\proj2\client1.py"
Username: user2
user2 > list
user2 >
user2 > |user1|user2|
user2 > user1-hello
user2 >
```

SERVER :

```
PS D:\PROJECT\proj> python -u "d:\PROJECT\proj\server1.py"
Listening for connections on 127.0.0.1:1234...
Accepted new connection from 127.0.0.1:31779, username: user1
Accepted new connection from 127.0.0.1:31781, username: user2
Received message from user2: list
Received message from user2:
Received message from user2: user1-hello
Received message from user1:
```

5. Results and Discussion

5.1. Experimental Results

S-Box is a number replacement table that takes one number and returns another. This is a nonlinear action. In S-Box, each of the n input bits is represented by one of two characters. After that, the set of 2n characters is transposed to one of the other characters in the set. After that, the character is transformed to a nbit output. It is simple to demonstrate that there are (2n)! distinct replacement or connection patterns. As a result, if n is big, the number of feasible S-Boxes is also huge. If a cryptanalyst wants to decode the AES algorithm, he should strive to produce as many S-Boxes as possible and employ them in the AES encryption system's Sub Bytes function. As n grows larger, say n = 128, the cryptanalyst's work becomes computationally impracticable; then 2n = 10^38, and (10^38)!! possible S-Box can be generate which is an astronomical number (1038)! = ∞ Possible S-Box can be generate when n = 128

For key_hex1={5e,d3,f1,b4,7c,18,51,9a,ae,81,42,57,42,78,dc,8f}

67	1E	77	7B	4B	4F	6F	46	1F	F8	6A	87	B9	71	60	CD
AD	99	85	AF	48	D5	81	0E	17	82	9C	C9	35	13	47	0C
E5	7C	B8	A4	E1	55	E3	DF	02	01	31	70	B0	B7	BD	C1
80	D4	8F	E2	BF	54	B2	68	3B	A1	18	0F	9A	03	DC	EC
D6	E6	AC	B3	3F	7A	22	C5	5E	1D	EB	CA	D2	A6	5A	A0
D0	EF	AA	FB	43	16	AE	07	7D	F9	0A	33	93	26	E9	44
9B	4D	53	D1	00	04	20	5B	5D	51	76	CB	DB	6E	4A	6B
25	AB	D9	E7	73	BA	9D	DA	59	12	74	C0	C4	F7	11	63
94	C7	ED	6C	27	61	CE	39	30	FA	08	C3	36	EA	7F	E8
19	A2	A8	05	52	D7	9F	8C	4C	BE	9E	96	3E	FE	8E	A7
49	D3	8B	38	F1	1C	B6	41	1A	E4	FD	F5	89	D8	24	CF
5F	72	37	6D	92	95	4E	FC	83	0B	65	45	0D	A5	14	3D
E0	2D	C2	3A	B1	06	B4	C6	2E	32	10	62	8D	34	DE	BC
CC	C8	B5	66	2F	58	F6	98	79	56	86	F0	2A	2B	57	3C
88	78	84	90	91	97	EE	5C	BB	DD	69	15	42	40	28	64
75	A3	F2	23	2C	50	7E	1B	29	F4	8A	21	A9	FF	F3	09

Only two bits of key hex1 are changed, and the key and S-Box2 are shown in Table 3. By altering only two bits of the key, we can detect 225 differences between S-Box1 and S-Box2, implying that almost 87 percent of the second S-Box is modified. The distinction between the S-Box1 and S-Box2 parts.

key_hex2={5d,d3,f1,b4,7c,18,51,9a,ae,81,42,57,42,78,dc,8f}

4D	CA	D7	59	3F	50	DE	5B	BC	21	16	2D	3A	55	28	A9
82	7D	EB	FA	9E	FE	F0	51	E2	0C	72	68	A2	B9	99	73
E3	C5	B6	01	46	AA	5D	DA	42	F2	BE	B5	77	B7	52	F9
0D	9C	27	B2	B3	86	C7	FD	29	C0	6D	9A	7B	81	DC	EC
BF	AD	17	22	45	89	83	31	5E	35	2F	E6	64	A4	3B	69
D0	A1	7C	FB	43	65	AE	38	47	D9	24	87	67	26	02	E7
80	E1	BD	D4	00	7F	20	8F	61	23	11	05	6F	6E	4A	6B
2C	94	40	88	75	2E	D1	10	07	A0	F3	C9	E5	F7	19	CF
6C	04	66	25	62	30	54	39	C3	5F	09	ED	1F	EA	18	CB
48	A3	7A	12	92	FC	9F	F5	4C	53	76	8B	B8	06	AC	A7
A8	91	0B	1E	F1	C6	79	08	5A	E4	15	95	44	F8	D3	96
BA	70	37	0F	D8	8A	57	CC	1B	98	33	AF	D5	A5	85	8E
C2	B1	E8	DD	CE	41	32	71	4E	1A	E0	78	8C	A6	49	E9
13	C8	C1	1D	34	58	3E	6A	B4	56	2B	0E	9D	84	FF	3C
DB	7E	4F	EE	9B	97	4B	5C	BB	D6	14	74	60	0A	1C	D2
B0	DF	93	90	63	C4	EF	03	36	8D	2A	F4	CD	F6	AB	3D

5.2. CONCLUSION :

Diffie hellman and modified AES are used to offer secure communication in this chat client. For AES, we developed a novel approach for generating dynamic S-Box from encryption key. To produce new S-Boxes, the quality of this technique was checked by modifying only two bits of the encryption key. This approach will construct more secure block cyphers, address the problem of fixed structure S-Boxes, and improve the AES block cypher system's security level. The primary advantage of this technique is that it can produce a large number of S-Boxes simply altering the Cipher key.

6 . References

Web links:

1. https://www.researchgate.net/publication/333802735_Development_Of_A_Client_Server_CrYptography-Based_Secure_Messaging_System_Using_RSA_Algorithm/link/5e2a3c914585150ee77df1e1/download

2. https://www.ijeat.org/wp-content/uploads/papers/v8i6S3/F13430986S319.pdf

3. https://patentimages.storage.googleapis.com/77/9d/99/651d6d51c187c2/US6219045.pdf

4. https://www.researchgate.net/publication/335528033_Implementation_of_Security_Enhancement_in_AES_by_Inducting_Dynamicity_in_AES_S-Box

5. https://www.inderscienceonline.com/doi/pdf/10.1504/IJICS.2014.068103

6. http://www.jmest.org/wp-content/uploads/JMESTN42351694.pdf

7. https://www.etasr.com/index.php/ETASR/article/download/1272/536

8. https://www.cscjournals.org/manuscript/Journals/IJCSS/Volume6/Issue1/IJCSS-630.pdf

9. http://www.iosrjournals.org/iosr-jce/papers/Vol20-issue6/Version-1/I2006013339.pdf

10. https://www.researchgate.net/profile/Rolou_Lyn_Maata/publication/322221902_Design_and_Implementation_of_Client Server_Based_Application_Using_Socket_Programming_in_a_Distributed_Computing_Environment/links/5a4c5500458515a6bc6c01d3/Design-and-Implementation-of-Client-Server Based-Application-Using-Socket-Programming-in-a-Distributed-Computing Environment.pdf?origin=publication_detail

Journal:

1. Razi Hosseinkhani & H. Haj Seyyed Javadi.” Using Cipher Key to Generate Dynamic S Box in AES Cipher System” IEEE Trans. on Circuits and Systems – I: Volume: 53 Issue: 6 – 2006

2. Federal Information Processing Standards, “Advanced Encryption Standard (AES)” Publication 197, November 26 – 2001

3. Kazys KAZLAUSKAS, Janunius KAZLAUSKAS,” Key-Dependent S-Box Generation in AES Block Cipher System” , Inoformatica Volume: 20 - 2009