Secure Browser-to-Browser Communication using Cryptographic Algorithms and Socket Programming

Smit Shukla
18BCI0198
SCOPE Department
Vellore Institute Of Technolology
smitshukla8@gmail.com

Vaidit Patel
18BCI0122
SCOPE Department
Vellore Institute Of Technology
vaiditpatel@gmail.com

Abstract

The project proposes to design a web-based interface to communicate from one computer to another using TCP client-server paradigm through browser. The communication will be two-way communication which will be secured by any of the selected cryptographic algorithm such as Affine, Hill, Playfair, ElGamal, AES, DES etc. by the users on both the ends. The sending and receiving of the encrypted text on the web will be developed using socket programming based on TCP client-server model and Web Sockets. The interface is designed for multiple users i.e. any number of users can connect in a room and chat with each other.

In the interface the user will be asked for login username and password

i.e. the information submitted during signup on the web page. After successful login the user will be asked if he/she wants to create a room or enter a room. After submitting the required inputs, the user types the message after clicking on the send button the message is encrypted with a access key of the room and the encrypted text will be sent to the intended computers where the other users can decrypt the message using the correct access key of the room. In the same way the other users can reply and both users can communicate with one another. The whole communication is secured using the room access key which is used to encrypt and decrypt the messages.

I. INTRODUCTION

Communication has become an important part in our day to day life. It can be either for business purposes, personal chatting or even covert communication between military personnel. Today there are many chat applications available on the app stores which can be downloaded on personal devices and used to communicate. Many of them offer end-to end encryption like WhatsApp. But there may be situations when you are using an unsecure non-personal device. You cannot download an app or any software on this device, or signup and use the device for communication.

This is where this proposed web-based chat application comes to rescue. The proposed model allows any no of users to create a chat room with an access code. After creating the chat room any user having the room id and corresponding correct access code can enter the room and engage in the chat. This chat room is secured using the AES encryption i.e. all the incoming and outgoing messages are secured using the AES algorithm and the symmetric key for that algorithm is the access key of the room.

The web-based chat application is a one-time use application i.e. the messages are not stored on any of the devices but in a socket buffer. Once all the users log out from the chat room the messages cannot be retrieved by any means. This is one of the main advantages of this chat application.

II. LITERATURE REVIEW

1. Security mechanisms in a web serve

This paper deals with session processing module for a server that are adapted to communicate across the Internet with a plurality of clients. The processing module runs within a servlet and allocates a session identifier in response to a first input stream of a session between a client and the server; negotiates communication characteristics for the session; and instantiates, according to the communication characteristics, routines for processing subsequent session input streams containing request data and routines for generating session output streams containing response data.

2. SQLrand: Preventing SQL Injection Attacks

This paper deals with practical protection mechanism against SQL injection attacks. Such attacks target databases that are accessible through a web front-end, and take advantage of flaws in the input validation logic of Web components such as CGI scripts. It applies the concept of instruction-set randomization to SQL, creating instances of the language that are unpredictable to the attacker.

3. Web and Database Security

Security in web applications is the most important concern when it comes to processing transactions in the web. One of the major issues is the security and privacy of data and information transferred, stored and processed through at real time. These days, many online transactions between client and server are executed at the cloud data centers, where such sensitive data run on virtual resources. This paper introduces the types of attacks that target web applications. In addition, several examples on many attack scenarios are introduced.

4. Automated Initialization of Web Software Projects

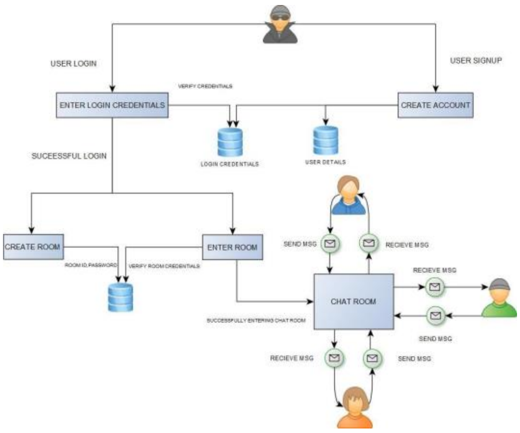
This thesis researches methods to design such a template and describes the implementation of a working tool for initializing a new project. The project template contains a set of predefined components for different aspects of the continuous development pipeline. These components include configuration files for a continuous integration pipeline, static analysis and a project skeleton for a front-end and back-end application, among many others**.**

5. Getting Started with WebSockets

WebSockets are a useful tool that allow for two-way communication between a web server and a browser. This is an introduction to the basics of WebSockets using Express, HTML, and vanilla JavaScript.

III. Methodology

A. Architecture



B. Security Aspects:

1. **Confidentiality:** Confidentiality refers to protecting the information from being accessed by unauthorized parties. AES is being used for encrypting data. Even when the transmission or storage medium has been compromised, the encrypted information is practically useless to unauthorized persons without the proper keys for decryption. If intercepted interceptor will not be able to crack unless they know the key. Advanced Encryption Standards (AES) is used for encrypting and decrypting the data using room access key.

2. **Access Control:** Users are provided with passcodes to grant access for chat group/room that they have the passcode to access among the different chat groups. It ensures that only the people eligible to be the part of a particular chat group are granted permission and able to log in to the chat room created.

3. **Integrity:** Digital signatures being used for ensuring integrity and user authentication. Public and private key pair being used for digital signature. Sender creates a message digest (md1) using a hash function. The message and message digest are encrypted. Transmitted to receiver. Receiver decrypts message and message digest (md1). Receiver applies same hash function to message to create message digest (md2). The two message digests must match to ensure data integrity. RSA alongside message digest is also used for digital signature to verify the user’s authentication.

4. **Non-Repudiation:** Non repudiation ensures that the sender cannot deny sending something. Chat once posted cannot be deleted and sender cannot deny sending something to the receiver and chats will be stored once the session between the user’s in a group is terminated/ended. In the chat application non-repudiation is ensured so that the user cannot delete his messages.

5. **Authentication:** User ID and password are being used for authenticating that the user belongs to registered set of people who can access the page. All the passwords are encrypted using SHA-256 algorithm. So in case of any attack on databases, passwords are not compromised.

C. Algorithms Used

1. Advanced Encryption Standards (AES):

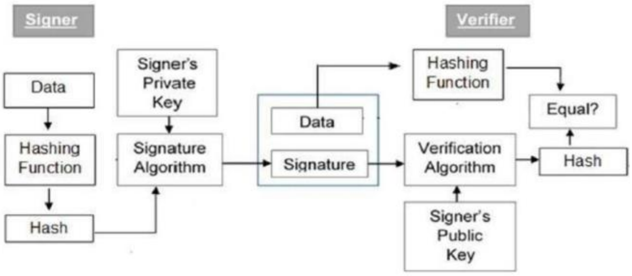
The Advanced Encryption Standard, or AES, is a symmetric block cipher chosen by the U.S. government to protect classified information and is implemented in software and hardware throughout the world to encrypt sensitive data.

AES is an iterative rather than Feistel cipher. It is based on ‘substitution–permutation network’. It comprises of a series of linked operations, some of which involve replacing inputs by specific outputs (substitutions) and others involve shuffling bits around (permutations).

Interestingly, AES performs all its computations on bytes rather than bits. Hence, AES treats the 128 bits of a plaintext block as 16 bytes. These 16 bytes are arranged in four columns and four rows for processing as a matrix.

Unlike DES, the number of rounds in AES is variable and depends on the length of the key. AES uses 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys. Each of these rounds uses a different 128-bit round key, which is calculated from the original AES key.

The schematic of AES structure is given in the following illustration –



2. SHA-256:

SHA-256 is a member of the SHA-2 cryptographic hash functions designed by the NSA. SHA stands for Secure Hash Algorithm. Cryptographic hash functions are mathematical operations run on digital data; by comparing the computed "hash" (the output from execution of the algorithm) to a known and expected hash value, a person can determine the data's integrity. A one-way hash can be generated from any piece of data, but the data cannot be generated from the hash.

SHA-256 has quite good technical parameters:

block size indicator (byte): 64.

maximum allowed message length (bytes): 33.

characteristics of the message digest size (bytes): 32.

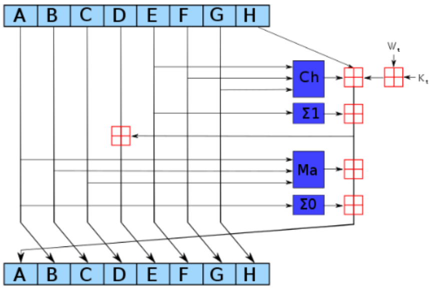
the standard word size (bytes): 4.

internal position length parameter (bytes): 32.

the number of iterations in one cycle: 64.

the speed achieved by the Protocol (MB/s): approximately 140.

The Sha-256 algorithm is based on the Merkle-Damgard construction method, according to which the initial index is divided into blocks immediately after the change is made, and those, in turn, into 16 words



One iteration in a SHA-2 family compression function. The blue components perform the following operations:

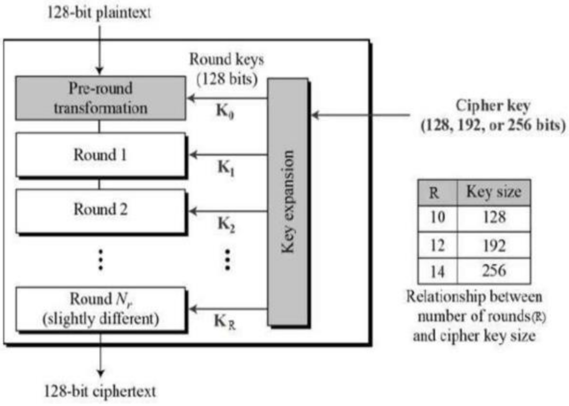
$$\mathrm{Ch}(E,F,G) = (E \land F) \oplus (\neg E \land G)$$
$$\mathrm{Ma}(A,B,C) = (A \land B) \oplus (A \land C) \oplus (B \land C)$$
$$\Sigma_0(A) = (A \ggg 2) \oplus (A \ggg 13) \oplus (A \ggg 22)$$
$$\Sigma_1(E) = (E \ggg 6) \oplus (E \ggg 11) \oplus (E \ggg 25)$$

The bitwise rotation uses different constants for SHA-512. The given numbers are for SHA-256.

The red ⊞ is addition modulo 2^{32} for SHA-256, or 2^{64} for SHA-512.

3. RSA Digital Signatures:

Digital signatures are the public-key primitives of message authentication. In the physical world, it is common to use handwritten signatures on handwritten or typed messages. They are used to bind signatory to the message. Similarly, a digital signature is a technique that binds a person/entity to the digital data. This binding can be independently verified by receiver as well as any third party.



Digital signature is a cryptographic value that is calculated from the data and a secret key known only by the signer.

By adding public-key encryption to digital signature scheme, we can create a cryptosystem that can provide the four essential elements of security namely – Privacy, Authentication, Integrity, and Non- repudiation.

IV. Implementation

A. *Technical Overview*

The project has used a variety of front and back end frameworks for implementations such as:

- HTML: For front-end development
- CSS: For front-end development
- JS: For animations and socket programming.
- Php: For front and back end connections, session creation and user authentication.
- NodeJs: implementation of socket programming and server handling.

B. *Hardware & Software Requirements*

- **Hardware:**

  (a) Client/User Side: Any device with internet connection.

  (b) Server Side: Apache server to host the site.

- **Software:**

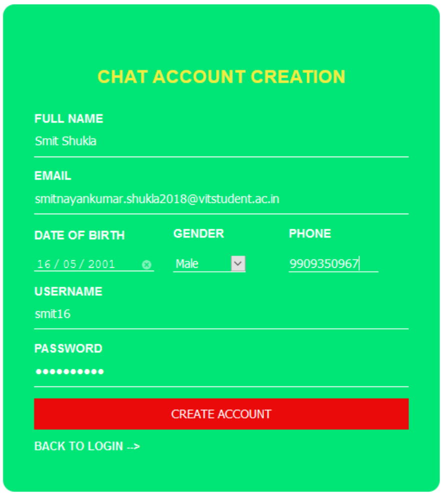
  (a) Client/User Side: Internet Browser.

  (b) Server Side: Apache tomcat, Php , Socket.io.

V. MODULE DESCRIPTION

1. Sign-Up Page: -

This page is for the first-time user to sign-up with their personal details. This page ensures that no 2 users have same username ande-mails.


CHAT ACCOUNT CREATION

FULL NAME
Smit Shukla

EMAIL
smitnayankumar.shukla2018@vitstudent.ac.in

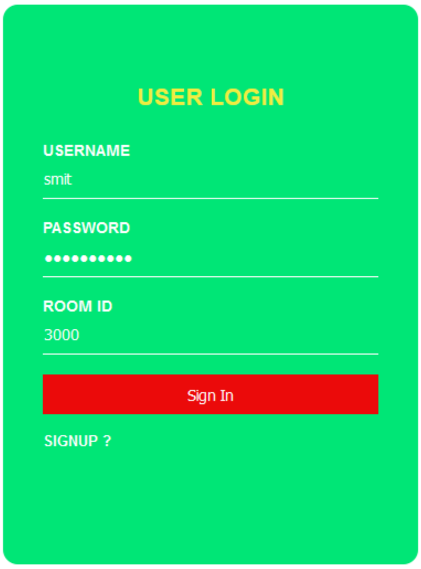
DATE OF BIRTH: 16 / 05 / 2001
GENDER: Male
PHONE: 9909350967

USERNAME
smit16

PASSWORD
••••••••••

2. Login Page: -

This is a user authentication page to verify the registered users and prevent invalid users. This also takes an input by the users to enter a predefined chat room by room administrator.


USER LOGIN

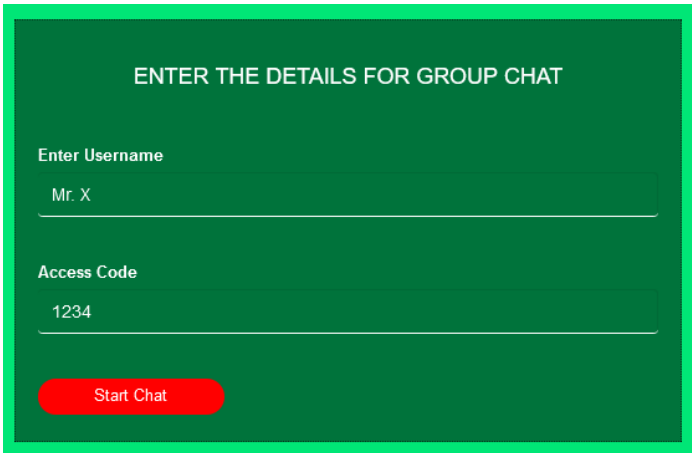
USERNAME
smit

PASSWORD
••••••••••

ROOM ID
3000

3. Chat Room Login Page:-

This is the chat room login credential i.e. the access code which allows the user to enter into the requested chat room. Incorrect access code will result in failure to receive and send messages.

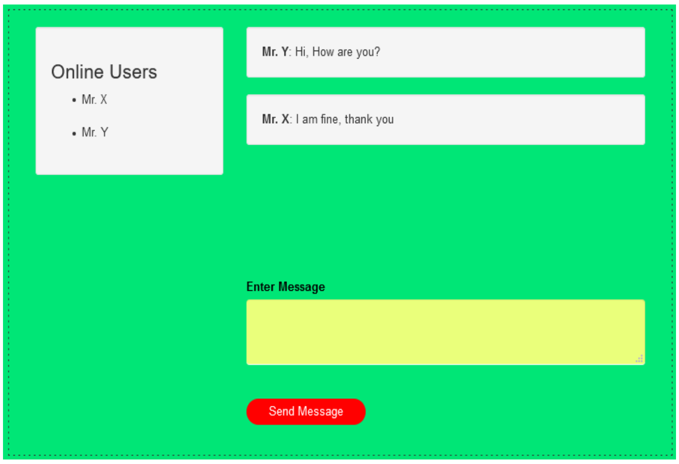

ENTER THE DETAILS FOR GROUP CHAT

Enter Username
Mr. X

Access Code
1234

4. Chat Page:-

This is the main chat page which is two way encrypted and where the users can chat anonymously by using the username they entered in previous screen. Each user can see the online users in their room. But they cannot see users in the other room.



Online Users

- Mr. X
- Mr. Y

Mr. Y: Hi, How are you?

Mr. X: I am fine, thank you

Enter Message

VI. VULNERABILITY ANALYSIS

After developing the application, the application was tested against vulnerabilities. After careful planning and applying ethical hacking techniques, following vulnerabilities were found.

1. SQL injection attack (SQLi):

SQL Injection is an attack that poisons dynamic SQL statements to comment out certain parts of the statement or appending a condition that will always be true. It takes advantage of the design flaws in poorly designed web applications to exploit SQL statements to execute malicious SQL code. The types of attacks that can be performed using SQL injection vary depending on the type of database engine. The attack works on dynamic SQL statements. A dynamic statement is a statement that is generated at run time using parameters password from a web form or URI query string.

In our application, in the login page we were authenticating the users by verifying credentials from the database. We were using the following query for authentication.

```
$res=mysqli_query($conn,"SELECT username, password,
email FROM login WHERE username='$usrid'");
$row=mysqli_fetch_array($res);
$count = mysqli_num_rows($res);


if( $count == 1 && $row['password']==$pass &&
$row['is_staff']==0) {
$_SESSION['chat'] = $row['email']; header("Location:
http://localhost:3000");
} else {
$errMSG1 = "Incorrect Credentials, Try again...";
}
```

Now, the problem with the query was that any hacker intelligently can supply an input which can bypass the user authentication and allows the user to proceed even without correct credentials.

For e.g.

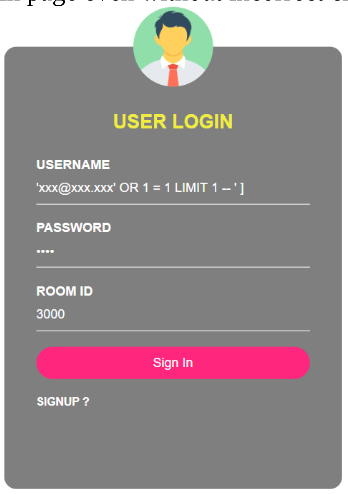
The above code can be exploited by commenting out the password part and appending a condition that will always be true.

SELECT * FROM login WHERE username = 'xxx@xxx.xxx' OR 1 = 1 LIMIT 1 -- '] AND password = md5('1234');

HERE,

- xxx@xxx.xxx ends with a single quote which completes the string quote
- OR 1 = 1 LIMIT 1 is a condition that will always be true and limits the returned results to only one record.
- -- ' AND … is a SQL comment that eliminates the password part.

Even when the username xxx@xxx.xxx is not in the database, it will show a result which will allow the hacker to bypass the login page even without incorrect credentials.

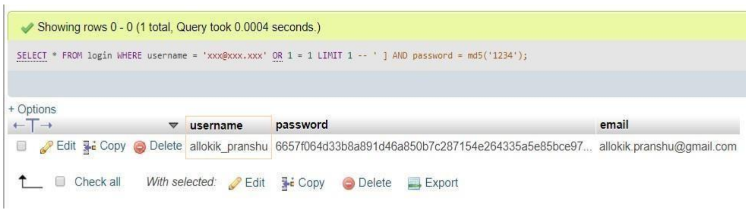


USER LOGIN

USERNAME
'xxx@xxx.xxx' OR 1 = 1 LIMIT 1 -- ']

PASSWORD
••••

ROOM ID
3000



✓ Showing rows 0 - 0 (1 total, Query took 0.0004 seconds.)

```
SELECT * FROM login WHERE username = 'xxx@xxx.xxx' OR 1 = 1 LIMIT 1 -- ' ] AND password = md5('1234');
```

	username	password	email
□ Edit Copy Delete	allokik_pranshu	6657f064d33b8a891d46a850b7c287154e264335a5e85bce97...	allokik.pranshu@gmail.com

2.Password transmission and storage attacks:

Another potential threat to the application was in case of an attack on the server and user authentication database. In case of such attack the user passwords and information may be at a risk from the hackers. Not only from the database the passwords are at risk during transmission and authentication using Php.

VII. PREVENTIVE MEASURES

After exploiting the possible vulnerabilities present in the application, security and other preventive measures were devised to cop-up with these vulnerabilities.

1. Prevent SQLi attack:

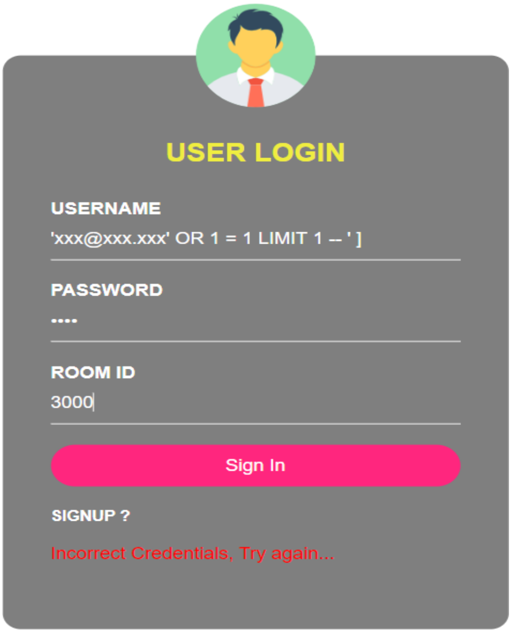
With user input channels being the main vector for SQL injection attacks, most of the defensive methods involve controlling and vetting user input for attack patterns.
Validation is the process of making sure the right type of input is provided by users and to neutralize any potential maliciouscommands that might be embedded in input string. For instance, in PHP, you can use the mysql_real_escape_string() to escape characters that might change the nature of the SQL command.

Previous SQL commands:

```
$usrid = trim($_POST['usrid']);
$usrid = strip_tags($usrid);
$usrid = htmlspecialchars($usrid);

$pass = trim($_POST['pass']);
$pass = strip_tags($pass);
$pass = htmlspecialchars($pass); if (!$error) {
$res=mysqli_query($conn,"SELECT username, password,
email FROM login WHERE username='$usrid'");
$row=mysqli_fetch_array($res);
$count = mysqli_num_rows($res);


if( $count == 1 && $row['password']==$) {
$_SESSION['chat'] = $row['email']; header("Location:
http://localhost:3000");
} else {
$errMSG1 = "Incorrect Credentials, Try again...";
}}
```



USER LOGIN

USERNAME
'xxx@xxx.xxx' OR 1 = 1 LIMIT 1 -- ']

PASSWORD
••••

ROOM ID
3000

Incorrect Credentials, Try again...

Modified SQL commands:

```
$usrid = mysqli_real_escape_string($con, $_POST['usrid']);
$pass = mysqli_real_escape_string($con, $_POST['pass']);
$sql_command = "select * from login where username=
'".$useid;
$sql_command .="'AND password = '" . $password . "'";
```

2. SHA-256 hash to store and authenticate password:

For storage during sign-up:

```
$pass = trim($_POST['pass']);
$pass = strip_tags($pass);
$pass = htmlspecialchars($pass);
$password = hash('sha256', $pass);
$query2 = "INSERT INTO
login(username,password,email) VALUES('$usr',
'$password', '$email')";
```

For authentication during login:

```
$pass = trim($_POST['pass']);
$pass = strip_tags($pass);
$pass = htmlspecialchars($pass);

if(empty($pass)){
$error = true;
$errMSG1 = "Please enter your password.";
}
else{
$pass = hash('sha256', $pass);
}
```

The hashing ensures that the passwords are stored as hash values and cannot be exploited by any suspicious user.

3. DIGITAL SIGNATURES:

The previous version did not have the provision of a digital signature therefore it was vulnerable to a man in the middle attack. Now this problem was tackled by using digital signatures for both the users to verify each other and authenticate each other therby making it impossible for a middleman to intercept and change anything in the message or change the message therby ensuring integrity.

VII. Conclusion

The developed application is very much secure in terms of integrity, confidentiality, authentication and non-repudiation. Security measures have been taken to prevent any type of database exploitation, SQL injection and cross-site scripting. The application is in its initial stages. There is wide scope of improvement in the application in terms of user interface, features and security point of view. Still some vulnerabilities may exist which may be hard to find after first analysis of the application. But with trials and errors and with repeating security analysis the vulnerabilities can be found and with proper knowledge can be rectified with appropriate security measures.

ACKNOWLEDGMENT

We would like to express our special thanks of gratitude to Arivoli A ma’am for guiding us and helping us in throughout the project progress. We would also like to thank VIT University for giving us the opportunity to carry out our research and finish our project. Last but not the least, we thank our parents, friends and relatives for their continuous support.

REFERENCES

[1] Lambert, H. S., & Wright, S. (2017). U.S. Patent No. 6,363,478. Washington, DC: U.S. Patent and Trademark Office.

[2] Liu, Muyang, Ke Li, and Tao Chen. "Security testing of web applications: a search-based approach for detecting SQL injection vulnerabilities." Proceedings of the Genetic and Evolutionary Computation Conference Companion. 2019.

[3] A research Paper on Cryptography Encryption and Compression Techniques, Sarita Kumari,IJECS Volume 6 Issue 4 April, 2017 Page No. 20915-20919

[4] Boyd, S. W., & Keromytis, A. D. (2018, June). SQLrand: Preventing SQL injection attacks. In International Conference on Applied Cryptography and Network Security (pp. 292-302). Springer, Berlin, Heidelberg.

[5] https://www.guru99.com/learn-sql-injection-with-practical example.html

[6] https://medium.com/@martin.sikora/node-js-websocket-simple-chat- tutorial-2def3a841b61

[7] https://spin.atomicobject.com/2018/10/01/websockets-getting-started/