# Web-IO for All

S.A Krishnaa
B-Tech
Information Technology
SRM University
kattankuluthur,Chennai-603203
Email: krishna201294@gmail.com

Burujupalli Sarmishta
B-Tech
Information Technology
SRM University
kattankuluthur,Chennai-603203
Email: bsarmishta@gmail.com

***Abstract*—In this realtime environment theres an ever-growing need for communication between people or groups in all forms. In order to establish the communication between people the world has emerged through digitalised devices enhanced to send messages and make calls to the person we desire. So in a scenario when a user is connected to internet, security plays a major role. Implementing a web based chat system for sending messages to users, groups and making video calls, which ensures security in a manner that third parties cannot get into the system easily through encryption methodologies, ensures that every detail of the user is encrypted. The paper focusses on two different security aspects. The first part ensures that all the users data is modified and stored using an XOR operation with a key and securing the conversation between two users through AES-256 algorithm.The second part ensures integrating the possibilities of socket.io and node.js with WebRTC to make video calls work only in a secure manner. This is because WebRTC works only on SSL certified sites.**

***Keywords*—Socket.io,Laravel,node.js,WebRTC**

## I. Introduction

The internet has become a huge social platform for communication. People are accustomed to sharing there valuable information. The internet is also known to be more prone for vulnerability. Chat applications and chat based web pages are something very common to find these days. But all these lack the ability to secure a users data using encryption standards, and hence are prone to be easily hacked or modified by the admin in certain scenarios. The project is both an implementation of secure messaging and secure information storage of the user. The project has implemented various security measures to see to that third parties cannot perform Man in the Middle attacks easily.

## II. Literature Review

### *A. LAN Chat Messenger (LCM) using Java programming with VOIP*

A Chat system or chat application is important for easy communication and how possibly it can be done within an IP network. The paper declares that such a chat based system is working well using java and LCM .It is also truly known fact that communication through devices is the mode of transfer for the next generation. The paper also adds a fact that such a system of communication over IP network will be the key solution in industries. It reduces also the human need to transmit messages physically. The paper is also a birds eye view of how voice chat can be made possible[1].

For a chat based system its easy to write in nodejs using socket.io which has features of adding require() functionality which extends all new possible features that one may want to use.

### *B. 128-Bit Versus 256-Bit AES Encryption*

The paper above is well said clean description about 128-AES and 256-AES. It tells us that an attackers path to data is the encryption module and the encryption engine , and hence this reason conveys us how important it is to use encryption to stop or prevent attacks from a third party. The paper starts with a note on the preferred length for encryption keys, but continues to convey that the length of a key is not a major concern.And so once the these encryption engines are set properly the next major location for security is with the authentication portion of the system . So a strong authentication system and a proper encryption engine would ensure for a safe working of our web based chat system.

The paper also suggests some reasons as to why AES :

1) To encrypt relatively short messages
2) To compute digital signatures
3) To establish or verify cryptographic keying material.

The paper also has suggesting evidence to convey saying that AES 256 is the strongest level of encryption solutions among 128,192 or 256 bit keys[2].

The features of what and why AES has made us use this in our project since messages carry vital information .The AES is well suited for data of small lengths only was known to us through this paper .The AES encryption engine is independent on OS as for a software and hence it runs easily with require functionality in nodejs.

### *C. P2P Live Video Streaming in WebRTC*

This paper gives a detailed study on feasibility of implementing live video streaming protocols into web applications with the use of WebRTC. This paper also tells us that there are existing programs to distribute video content efficiently, But web pages until recently not been able to leverage these technologies and that , WebRTC would serve as a solution by enabling peer-to- peer communication directly between browsers without any need for a server as an intermediary or without the use of any plugins. Recent study predictions say that soon ninety percent of the Internet will be videos. Video streaming consumes significantly greater bandwidth than other

traffic. With the introduction of Web Real-Time Communication bandwidth required for internet video streaming has reduced and allows for real time communication between Internet browsers[3].

## III. Proposed Methodology

A user who prefers to register, the registeration's details are collected from the view. These details are passed to the model which signifies the structure of database. The details that the user enters is modified into an unreadable form using an XOR operation with a key. This is decrypted with the same key when the user performs login in the controller. Hence functionalities for encryption and decyption are present in different files. The Chat system works with the help of a backend mongDB database. Cookies are also an additional feature that are implemented to avoid man in the middle attacks. The chat system is enabled to work using Socket.io. The MongoDB model is created in the server side of node.js where every client connects to the socket function. Before a user sends his message to another user, the client's message is encrypted using AES-256 on the client side. The client side doesn't usually support require(") functionality when working with node.js. To ensure that this works properly, browserify is a great alternative. Browserify ensures that all the package modules which doesn't work on client side is installed automatically. Hence the require('crypto') package will also work successfully on the client side. With this package installed, the client's messages are encrypted before it is emitted to the server. The encryption key is a default key that would be stored on the client.

| | | |
|---|---|---|
| (10) ObjectId("56e65a3da6c560eded7... | { 6 fields } | Object |
| _id | ObjectId("56e65a3da6c560eded75ae83") | ObjectId |
| too | krishnaa | String |
| from | jimmy | String |
| msg | hey | String |
| created | 2016-03-14 06:29:17.617Z | Date |
| __v | 0 | Int32 |

Fig. 1. Initial System before encryption

| | | |
|---|---|---|
| (3) ObjectId("571b424d8e73a39e05e... | { 6 fields } | Object |
| _id | ObjectId("571b424d8e73a39e05e0c5ce") | ObjectId |
| too | krishnaa | String |
| from | sarmishta | String |
| msg | 3aaa | String |
| created | 2016-04-23 09:37:17.529Z | Date |
| __v | 0 | Int32 |
| (4) ObjectId("571b42c18e73a39e05e0... | { 6 fields } | Object |
| _id | ObjectId("571b42c18e73a39e05e0c5cf") | ObjectId |
| too | sarmishta | String |
| from | krishnaa | String |
| msg | 3aaa | String |
| created | 2016-04-23 09:39:13.178Z | Date |
| __v | 0 | Int32 |

Fig. 2. System after encryption

The mongo db model stores any messgae that is emitted from the client and is retrieved when the mongoDB model finds that there's a previous conversation list. Video calls are another feature which are implemented using WebRTC. WebRTC is a Real Time Communication feature which ensures video calls to be made between peers.WebRTC only works with SSL certified websites. This also ensures that there's an added envelop of security. The client will also be able to pause his video call and resume to continue his conversation simultaneously.

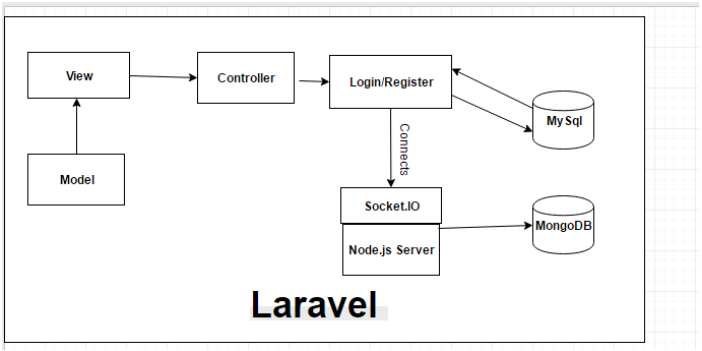


Fig. 3. System Architecture Diagram

## IV. Result

It can be noted that integrating socket.io,WebRTC and node.js with laravel php framework is a great way to enhance new features into a system easily. This application functions as a peer to peer chatting system where sharing of thoughts and information becomes easier. Along, with this users can also join a group and send messages. Users are also given an option to make video calls to their peers, using webRTC (web based real time communication). Generally, for making video calls many of the applications or web applications need plugins to be installed. Through WebRTC it is convenient to use SSL certified webpage rather than making a user to install a plugin.

Considering the importance of security in todays digitalized world, security has also been given its due consideration in this application. This has been done in two stages. In the first stage, the details of the users have been encrypted using an encryption standard. And, in the second stage, the messages that the user sends is encrypted using a standard AES-256 encryption technique. With this, it would be difficult for the third parties to listen to the conversations, thus, ensuring the users privacy.

## V. Conclusion

This paper is a great way to establish secure communication with your peers but there can be more to this project. Socket.io and WebRTC are huge frameworks which also lag in some features. Implementing these two together fully would increase scalability and also bring in features that one many not have with the other. Bringing them both inside Laravel PHP framework and node.js provides a great way to bring in modularity.

## Acknowledgment

I would like to express my deepest gratitude to my guide, Ms JeyaBharathi for her valuable guidance, consistent encouragement, personal caring, timely help and providing me with

an excellent atmosphere for doing research. All through the work, in spite of his busy schedule, she has extended cheerful and cordial support to me for completing this research work.

## REFERENCES

[1] brahim Muhammed Abba, Norshakirah Ab.Aziz, Umapthy Eaganathan," LAN CHAT MESSENGER (LCM) USING JAVA PROGRAMMING WITH VOIP",IEEE

[2] Technology Paper "128-Bit Versus 256-Bit AES Encryption Practical business reasons why 128-bit solutions provide comprehensive security for every need"

[3] Florian Rhinow, Pablo Porto Veloso, Carlos Puyelo, Stephen Barrett, Eamonn O Nuallain,"P2P Live Video Streaming in WebRTC",IEEE

[4] Handel, M and Herbsleb, J.D (2002)" What Is Chat Doing in the Workplace?", ACM Publishers, New York

[5] Varun Singh,Albert Abello Lozano,Jotgot,"Performance Analysis of Receive-Side Real-Time Congestion Control for WebRTC",IEEE

[6] Fei Shao, Zinan Chang, Yi Zhang,"AES Encryption Algorithm Based on the High Performance Computing of GPU Fei Shao, Zinan Chang, Yi Zhang",IEEE

[7] Pavel Segec, Peter Paluch, Jozef Papan, Milan Kubin,"The integration of WebRTC and SIP: way of enhancing real-time, interactive multimedia communication",IEEE