iJRASET
International Journal For Research in Applied Science and Engineering Technology

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

**Volume:** 9 **Issue:** VI **Month of publication:** June 2021

**DOI:** https://doi.org/10.22214/ijraset.2021.35305

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
*ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429*
*Volume 9 Issue VI Jun 2021- Available at www.ijraset.com*

# Real Time Chat Web Application having E2E Encryption

Aparna Mete Sawant[1], Pratik Fandade [2], Adnan Sadar[3], Yash Jadhav[4], Akash Dhadiwal[5], Nikhil Ravulaparthy [6]
[1, 2, 3, 4, 5, 6]*Department of Information Technology,* Vishwakarma Institute of Technology, Pune, India

***Abstract: Data Privacy is a term which is being coined so often since the popular messaging app WhatsApp updated its privacy policy starting from 8th February 2021. The users started looking towards alternative options like Signal and Telegram which were more privacy driven since their inception. Users have started showing concern for their data privacy and are more aware about this topic. End-to-end encryption service is being used by chat applications which involves communication where only the users communicating can read the messages being sent. Some of the popular encryption algorithms that are being used currently are AES, RSA, Triple-DES, Two-Fish.***
***Keywords: AES256, E2E, Socket.io, NodeJS, React***

## I. INTRODUCTION

End-to-end encryption (E2EE) earlier was the sole domain of the tech savvy because it involved complicated operations to use it. However, technological advances in these past few years have made end-to-end encryption much easier to use and also accessible. Inbuilt libraries are also present which can be used by developers to integrate encryption service in their applications without having to worry about the inner working and study of the algorithm being used. Encryption algorithms mainly consist of two categories: Symmetric and Asymmetric algorithms. Symmetric Algorithm involves the use of a single key to encrypt and decrypt data, whereas asymmetric algorithm involves use of two keys i.e. public key which can be shared with everyone and a private key which is known to the key's generator only. RSA algorithm is a popular asymmetric encryption algorithm, it is based on the factorization of the product of two large prime numbers, it is a very secure encryption algorithm therefore being popular. It is often used in digital signatures; it does have a drawback though that it works slower when the volume of data is large.

The algorithm which we have used in this project is the AES (Asymmetric Encryption Standard) algorithm which is a symmetric encryption algorithm. In this algorithm, fixed blocks of data are encrypted at a time i.e. 128 bits. The keys used to decrypt the text can be of varying sizes like 128, 192, or 256 bit long. We have used the AES256 algorithm which encrypts the data in 14 rounds. Each step consists of substitution, transposition, and a few more steps. AES encryption technique is the most commonly used algorithm at present because of its robust security.

## II. LITERATURE REVIEW

### *A. Review On Various Cryptography Techniques*

In "*A Study of different data encryption algorithms at security level*" [1] authors have discussed three types of Data Encryption Algorithms:-

*1)* Symmetric or Secret Key Cryptography: In this kind of algorithm, both encryption and decryption is done using single key. This key is known to both sender and receiver of data. Some examples for following method are, DES, AES, RC5 etc.

*2)* Asymmetric or Public key Cryptography: In this kind of algorithm, both encryption and decryption is done using two different keys. One of the key is public whereas other is private. Sender encrypts the data using public key which is available to all publicly. On the other side receiver uses private key to decrypt the message. Even though these 2 keys are different from each other but, they are mathematically connected with each other. Some examples for following algorithms are RSA, Elliptic curve etc.

*3)* Hash Function: In this kind of algorithm no key is used and only some mathematical methods are used. This kind of cryptography is also known as one-way encryption as data cannot be encrypted back to plain text. Some examples for following algorithms are MD5, SHA-1 etc.

In "*Advanced Encryption Standard (AES) Algorithm to Encrypt and Decrypt Data*" [2] author gives references to a few papers mentioning symmetric key is much more effective and faster than Asymmetric. Some of the common symmetric algorithms are Advance Encryption Standard (AES), Blowfish, Simplified Data Encryption Standard (S-DES) and 3DES. AES is one of the most popular block cipher algorithm and has not been cracked till date. The author talks about the 3 different key sizes like AES 128,192,256 bit, each cipher containing 128-bit block size. Three types of cryptography techniques are mentioned, namely symmetric cryptography that is the same key is used for encryption and decryption data, Asymmetric cryptographic which relies on two different keys for encryption and decryption, and cryptographic hash function using no key instead key it is mixed the data.

*B. Review On End-To-End Secure Chat Application*

In "*Developing an End-to-End Secure Chat Application*" [3] author proposed a set of requirements to make secure chat application:

1) req1: Password stored on the chat server should be encrypted.
2) req2: Providing either secure session or TLS. Secure session is a unique key for each session. Ensures that communication is with the right person and no man-in the-middle can read the messages.
3) req3: Messages must be encrypted to maintain security and privacy.
4) req4: Local storage must be protected by encryption
5) req5: Messages are not stored on the chat server but stored on the user's device.
6) req6: It is not allowed to exchange messages if they are not friends.

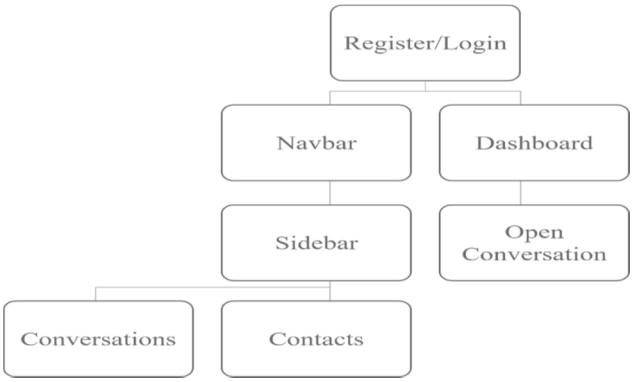
In "*Multi-User Chat Application*" [4] author helps us understand how things like billion users and large secure databases and security towards the biggest cyber-crime of 21st century that is data theft is taken care of, ensuring better user experience and safety. As one of our objectives is to deliver safety towards the data exchange, this made us come to a conclusion that encryption and decryption is solution towards handling the secure transfer of data and some protocols which needs to be followed for better safety.

## III. METHODOLOGIES

*A. Front-End*

We as a group of web developers have higher proficiency in JavaScript and the wide support of libraries and frameworks have helped to make us this choice, there are many other methods like java spring and J2EE which can be also used for the Application but this is what we find best in our scenario We are making use of React as our framework of choice for developing the web app, because of the wider support for react.

1) *UI Design & Process:* We are very familiar with the GUI defined for the web apps and messaging applications, a quick study about the UI from different applications such as WhatsApp, Facebook Messenger, and other chat application tell us about the important components we need to consider while designing the UI. So, the defining the important components that we need in our chat application are, first is the Login & Register component this helps in feeding the important new incoming users to the platform, this also the entry point of the application where the authentication of the user takes place and for a new client this will be responsible with associating them with the unique id, these help us eliminate the unauthorized access and minimize data threats. After login, there will be a dashboard component that will be rendered as the full application which will contain all the features and functionalities we are offering in our application. A dashboard will render multiple components in itself which will further expand on the features. Also, with this there will be a navbar component that will display Application information and user information, it will also contain methods that will help set the user as null in case he needs to exit from the application. This component will also contain the feature that will just provide a more aesthetically pleasing look to this and that will the dark mode implementation. Now as we have cleared this part of the application we expand on the dashboard and talk about the components inside it. The first component of our Dashboard is the sidebar, Sidebar will serve as a way to navigate through our contacts and conversations which will be saved in our application. There will be methods that will help establish contacts with the help of the unique id initialized during the registration process & also make multiple conversations with those contacts. They will also render all the contacts & conversations of the user and help in navigating through them. Lastly, we will have the components will be responsible for loading the conversation on the dashboard which is selected, and also methods that will help in forwarding messages and read messages in our dashboard. With that said given below is a flow diagram that tells us more about how these components are structured and work with each other.



2) *Custom Hooks:* Hooks provide a format of data that can be used to achieve state management in react. React also helps us make some custom hooks that can be used as in imported modules. We will define our custom hook which will make use of Local Storage memory to make the data available at all the time and help in maintaining state permanently. This will also help us in understanding the inner working of complex frameworks like react and help in eliminating code recurrence and code reliability.
3) *The context API for Global Data Tree:* Context API in react provides a cleaner way of implementing the reusable code which will be used often and will reduce the size of our components and also help retain global data for our application to use. We have used this functionality to make our code shorter and faster, we have implemented a total of 4 Context Classes that comes with their provider and helps in retaining information as global data. These contexts make use of our custom hook that is Use Local Storage Hook which will help us retain state at re-render. So firstly, we will need a global tree that will save data of all the contacts that will be used by us, so we declare a Contact context that will have a provider providing ways to add contact and access existing ones saved in the application. Next, we have is our Conversation context these do the same job as the contact context and provides the same functionality as the contact context. Next, we have a more important context that is the socket context, this takes care of the socket communication we will be doing this will help us in sending messages to the server and the other persons using this socket context, the provider will also give us methods to load the newer messages and organize the socket.io code in one file. Lastly, we have a Theme Context that enables us to make the theme as global data and save the user preferences accordingly.

### *B. Back-End*

Our Back end is mainly coded using Node.js as a server running environment and we are making use of Express App for the HTTP REST API and on the same port will be having a Socket.io based Web Socket API for end-to-end connection between the users.

1) *Rest API:* For this we are making use of Express library which has CORS enabled so that it can communicate properly with the front-end and also help to develop a REST API, this API is responsible for the user Authentication for which we are using a NoSQL MongoDB Database for storing all the authenticated users. We have also made a router for this which takes care of the Addition, Deletion, Updating procedures of users.
2) *Web Socket API:* For this we are making use of a Node package known as Socket.io, which uses the web-socket model for communication, this runs on the same server as our REST API and handles all the message delivering and users end-to-end connection.

### *C. Encryption*

As we have reviewed different techniques used for encryption and also concluded about the AES being superior, we also discovered the AES implementation varies from user to user and everyone uses different types of techniques to make it even better. So, for adapting the current ongoing version of AES we are making use of a node module known as CryptoJS which provides us support for all the different algorithms used in encryption and also due to the extensive support of other developers who have made AES better with time.

## IV. EXPERIMENTATION

| Algorithm | Key size | Block size | Structure | Features |
|---|---|---|---|---|
| **DES** | 64 bits | 64 bits | Festial | Not strong enough |
| **RSA** | 1024 bits | 128 bits | Public Key Algorithm | Excellent security and low speed |
| **RC6** | 128-256 bits | 128 bits | Festial | Good Security |
| **AES** | 128, 192, 256 bits | 128 bits | Substitution Permutation | Security is excellent. Best in category |

After testing various cryptography algorithms, it is observable that AES256 outperforms other algorithms when security is concerned.

*ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429*
*Volume 9 Issue VI Jun 2021- Available at www.ijraset.com*

## V. FUTURE SCOPE

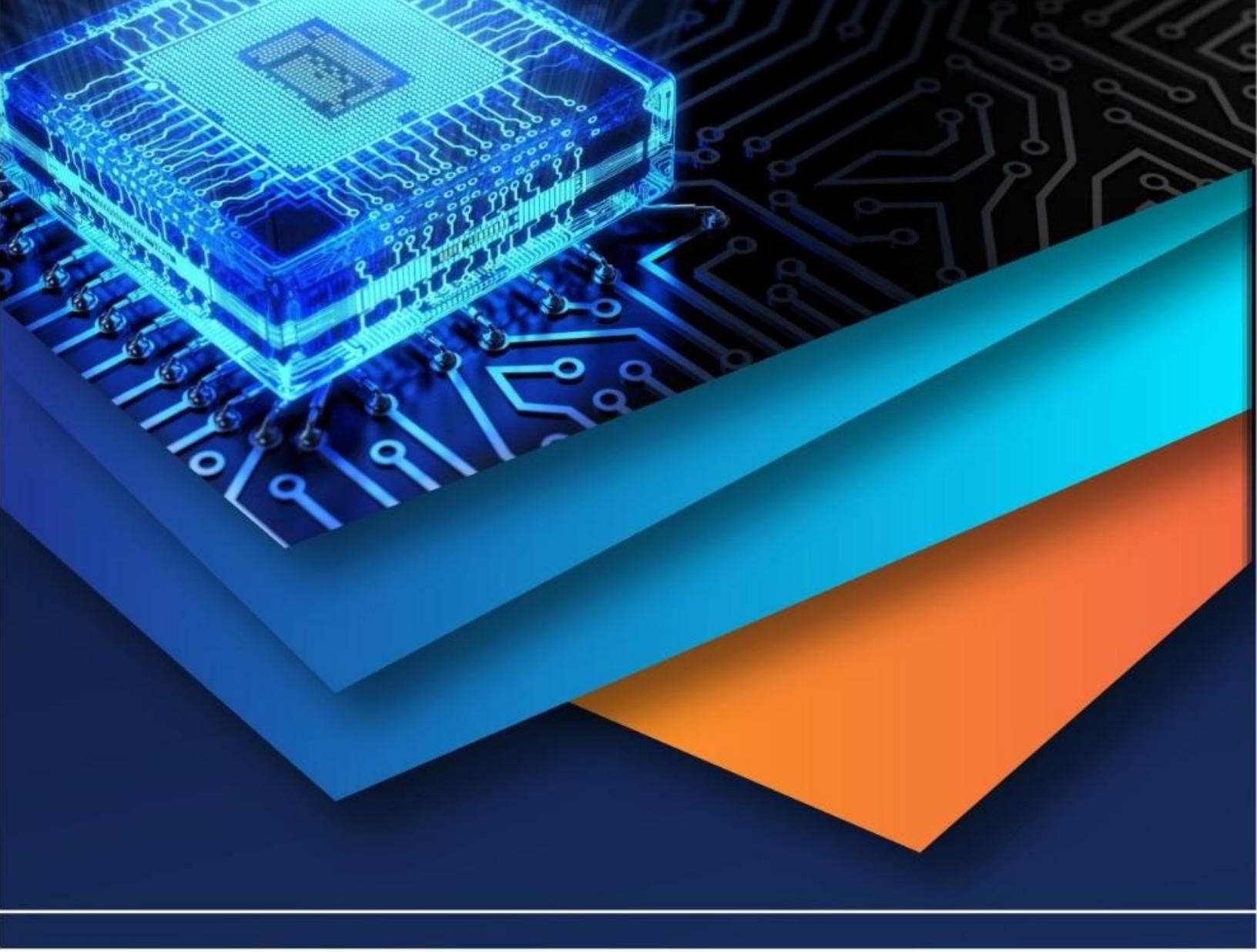
Developing this chat application in real time has helped us gain knowledge in various trends and technologies that can be and are being used in the modern times to integrate services over an entire web-ecosystem. There are a ton of messenger and chat applications in the market as of now and with the emerging and the upcoming apps as well, there is a lot of promise and a bright future. Newer implemented messaging and chat apps like FASTRA; you can upload and take real times videos using AR filters and they've a separate section for your friends and families. Users can also share various other multimedia and content and earn money as well with the aid of respective advertisers. Social media applications in general have a huge role in a person's life and messaging apps alone consume more than 5hrs on an average per day. Keeping this in mind we sure can say that, messaging apps have been created globally to update people on technology and it seems that the future of these apps is sure to be booming as well. As far as our app is concerned; additional features that we would further work on to make the application more effective; would be adding multi-media support as well as a news Chatbot of some sort and also a provision to exchange audios

## VI. CONCLUSION

The main objective of the project was to develop a Secure Chat Application for the users to converse and communicate with. We had taken a wide range of literature review for achieving all the tasks, from where we also came to know about the latest trends and applications that are existing in the market. We have made a detailed research in that path to cover the loop holes that the existing systems are facing and have tried our best to mitigate and/or eradicate them in our application. In the process of research, we also came to know about the latest technologies and different algorithms that can be and have been generally used to develop such services.

## REFERENCES

[1] A Study of different data encryption algorithms at security level Alongbar Daimary et al, (IJCSIT) International Journal of Computer Science and Information, Vol. 6 (4), 2015, 3507-3509.
[2] Advanced Encryption Standard (AES) Algorithm to Encrypt and Decrypt Data by Ako Mohammed Abdullah, June 16,2017.
[3] Developing an End-to-End Secure Chat Application Noor Sabah, Jamal M. Kadhim and Ban N. Dhannoon Department of Computer Science, Al-Nahrain University, Baghdad, Iraq.
[4] Multi-User Chat Application International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-9 Issue-5, June 2020.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY