IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Review of Computer Engineering Studies
Vol. 11, No. 3, September, 2024, pp. 39-44
Journal homepage: http://iieta.org/journals/rces

Secure End-to-End Chat Application: A Comprehensive Guide

Mainka Saharan[1], Neeraj Kumar[2], Vijay Kumar[2*], Akshay Juneja[3]

[1] Department of Computer Science and Engineering, National Institute of Medical Sciences University Jaipur, Rajasthan 303121, India
[2] Department of Information Technology, Dr. B. R. Ambedkar National Institute of Technology Jalandhar, Punjab 144008, India
[3] Amity School of Engineering and Technology, Amity University, Punjab 140306, India

Corresponding Author Email: vijayk@nitj.ac.in

https://doi.org/10.18280/rces.110302

Received: 22 January 2024
Revised: 11 July 2024
Accepted: 5 September 2024
Available online: 12 September 2024

Keywords:
secure, chat application, server, client, end-to-end

ABSTRACT

Chatting is a technical means of communication used to connect individuals and ideas. There is a significant increase in chat applications and their users since last decade. However, the majority of these applications do not ensure end-to-end security for users. This paper presents the procedure to design an end-to-end chat application that guarantee secure communication. In this application, users can communicate with each other via server with secure memory storage. The proposed architecture focuses on end-to-end encryption of messages, such that the messages are only visible in the sender's and receiver's devices. There is no third-party interference in the communication. This approach reduces the network traffic and provides dedicated communication.

1. INTRODUCTION

There are millions of individuals who communicate with each other using chat applications every day. But they are unaware about what happen to their messages once they send it [1]. From past few years, these applications are constantly updated by the developers, such that security and privacy are their utmost priority. In the initial years, the encryption was needed to secure business-related data from cyber threats. However, in today's era, almost every individual has personal and professional information, banking and money-transfer applications, and permanent login credentials of several applications [2]. Therefore, it has become essential to install or operate only those applications that are designed by either trustworthy third-parties, or the applications that provide end-to-end encryption [3].

In this paper, designing of an end-to-end encrypted chat application has been discussed to ensure information exchange between the senders and the receivers by using secured keys and protocols.

The structure of this paper is as follows:

Section 2 presents the survey of chat applications that exist in the literature. Section 3 discusses the requirements of designing a secure chat application for the users. The methodology of designing and deployment of chat application architecture is discussed in Section 4. Finally, section 5 presents the concluding remarks.

2. LITERATURE SURVEY

This section presents various chat applications, namely Facebook messenger, WhatsApp, Telegram, Line, and Viber, that have been developed over the years. These applications are globally available and used by thousands to millions of individuals in their daily routine [4]. These applications are discussed as follows:

2.1 Facebook messenger

This application employs different measures to ensure the security and privacy of users' data [5].

2.1.1 End-to-end encryption

This feature is available in "Secret Conversation" chat. In this chat, messages are accessed by only sender and receiver. Also, the secret messages are available on the sender's and receiver's devices only, i.e., devices initiating and accepting the messages [6].

A unique encryption key is generated for a conversation when sender sends a message. This message is encrypted and is only decrypted by using the decryption key present in the receiver's device.

2.1.2 Transport Layer Security

Transport Layer Security (TLS) is used by Facebook servers that act as an intermediate between sender's and receiver's device, when regular messages are send. Unlike secret conversation, regular messages are encrypted in transition stage by TLS. It is performed to protect the data from intruders.

2.1.3 User authentication

The user login into Facebook or its messenger by performing either one-factor or two-factor authentication. In

one-factor authentication, the user enters his credentials, i.e., email ID and password, to login into his account, to ensure the privacy of user's account. An optional two-factor verification feature is provided by Facebook to enhance the security of user's account. In this step, a code is sent to the user's registered mobile number to authenticate his identity.

2.1.4 Access controls

Facebook messenger have privacy settings that allow users to control the traffic on their profile. They can decide who can send them messages or see their online status. If someone tries to violate Facebook's community standard, the users can report or block that individual.

2.1.5 Data security

The regular messages are stored on Facebook's server. It has robust security measures to prevent unauthorized individuals to access the information. Also, users can use the "Secret Conversation" feature to chat as the messages in this conversation are not even stored on Facebook servers.

Also, users must secure their messenger applications using passwords, PIN, biometric locks, or some other security measures such that the intruder cannot access personal information.

2.1.6 Additional features

The users can set a timer to perform automated deletion of messages in secret conversation. The chats are deleted from both sender's and recipient's devices. The users can also set the notification preferences, i.e., manage the timing of message alerts, to ensure the user's privacy in outdoor environment.

2.2 WhatsApp

WhatsApp is primarily known for its enhanced security and privacy measures. It leverages the end-to-end encryption for all the messages [4, 7]. It comprises of several features as follows:

2.2.1 End-to-end encryption

WhatsApp has a default feature of end-to-end encryption for all messages and media files. It is used to ensure the privacy of both sender and receiver, such that the chat is inaccessible even by WhatsApp. It is because a combination of public-private key and symmetric encryption are used to secure messages.

2.2.2 Authentication

Only genuine users are allowed to send and receive messages as the users have to get their mobile number registered and verified. Also, user has to enter an optional six-digit PIN whenever he tries to login into a new device.

2.2.3 Access controls

The users decide who are allowed to see their last seen status, profile photo, about information, and status updates to guarantee their privacy. Also, they have the option to block the spammers and unwanted contacts.

2.2.4 Data security

The messages are stored only on sender's and receiver's devices. If the receiver's device is not connected to internet, it is stored temporarily on WhatsApp's servers in the encrypted form. Once the message is delivered to the receiver, it gets deleted from the WhatsApp server.

Therefore, this application must be secured by PIN, password, biometric locks, or other security measures.

2.2.5 Additional features

The feature of disappearing messages is of utmost importance. The chat gets automatically deleted after a specified duration, i.e., 24 hours, 7 days, or 90 days. Recently, WhatsApp has introduced a new feature of end-to-end encryption of backup data using Google Drive or other cloud platform. It is protected by a 64-digit encryption code to ensure the security of the users' data.

The users can also select the option of "View Once Media" such that the message can be viewed by the receiver only once before it gets disappeared.

2.3 Telegram

Another application used to send messages, videos, images, audios, and other media files is Telegram. This application was developed by Nikolai and Pavel Durov that works on MTProto Protocol. Although, Telegram application provides end-to-end security to the users, all the messages/communications among the users are not secured by default [4].

The default messages are accessible by the service providers even if they are encrypted. If the conversation is confidential, the users must deliberately start a special type of chat known as "Secret Chat" as it is based on client-to-client encryption policy. Secret chats are initiated by exchanging the encryption keys between the users such that the invitation is to be accepted [8].

2.4 Line

Line is a chatting application used for sharing messages, audios, videos, and other media files on computers, tablets, smartphones, etc. It is the originally designed by Naver Corporation, a Korean organization. Recently, it has introduced client-to-client encryption by using ECDH protocol [4].

However, this application does not support login into more than one device at a time. If a user login into a second device, he automatically gets logged out from first device [9].

2.5 Viber

Rakuten Viber is an application designed by a Japanese company named Rakuten. It is used for instant messaging, exchanging images, audios, videos, and other media files. The earlier versions of Viber were not secured because the service provider had the encryption key for all the conversations. It was not possible to perform the review of code. Also, the documentation of the security design was improper. However, the latest version of the Viber has introduced end-to-end encryption on all operating systems, i.e., iOS, Windows, and Android [4, 10].

Thus, it is observed that the data generated on and transmitted via messaging apps are vulnerable in terms of privacy, security, and confidentiality. So, there is a need for more robust and secure chat application, with both built-in security features as well as the location of stored data. The proposed chat application uses a client server architecture instead of peer-to-peer network. The encryption is achieved using XSalsa20 algorithm, i.e., this algorithm is used to

encrypt even the user's password. Further, a session key is generated which is only known to the sender and the receiver, and no other party can interfere in the communication.

Table 1 depicts the comparative analysis of different existing chat applications.

Table 1. Comparison of different chat applications

Messaging App	Messenger	WhatsApp	Telegram	Line	Viber
End-to-end Encryption	✓(optional)	✓	✓	✓	✓
Encryption in Transit	✓	✗	✓	✗	✗
Provider Accessible Private Key	✗	✓	✓	✗	✓
Deleted from Server	✗	✗	✗	✗	✓
Open Source	✗	✗	✓	✗	✗
Password Lock	✗	✓	✓	✗	✓
Verification on SMS/E-Mail	✗	✓	✗	✓	✓

3. REQUIREMENTS

There are certain requirements to be fulfilled before the designing and deployment of a chat application for users [11]. These requirements are as follows:

3.1 Server storage with password encryption

The passwords or PINs must be encrypted before their encryption on the servers. It is performed to ensure the security and privacy of the user accounts. In this technique, a robust algorithm called bcrypt is used to secure the account by using salting and hashing, against brute force and rainbow table attacks. It is observed that the passwords remain secure even if the servers are attacked. Thus, it increases the computational cost of the algorithm used by the intruder to decrypt the passwords.

3.2 Secure transmission of messages

The easiest method followed by the unauthorized individuals or parties is to interrupt the communication during its transmission. It is achieved by using TLS, such that the data is encrypted while it is in transit between the sender's device and the server. Another technique which is adopted is to provide a secure session between the sender's and the receiver's devices. Thus, it becomes difficult for the attackers to tamper the messages.

3.3 End-to-end encryption of messages

Nowadays, this requirement has become a necessity to ensure the privacy of the users. The conversation between the sending and the receiving parties is encrypted using different protocols, such as signal protocol. This conversation can be decrypted in their devices only. Thus, end-to-end encryption is performed to enhance the security of the users' personal information, such that the messages are secured from the potential intruders as well as service providers.

3.4 Device storage encryption

To ensure the security of users' personal information and protect it from the unauthorized individuals, it is necessary to provide encryption on their devices. In case the device is lost, stolen, or accessed by an unknown individual, it is mandatory to secure the confidential conversation on a device. It is achieved by encrypting the storage space on the device, i.e., the conversations or the media files are inaccessible without the decryption key. It is achieved by using Advanced Encryption Standard (AES) or another device-level encryption algorithm.

3.5 Message sending protocol

All the messages and media files must be encrypted before sending, followed by TLS for transmission, and it must get decrypted only on the receiver's device. It is performed to maximize the efficiency of the encryption process. The message sending protocol handles the message delivery notifications and confirmations. In case of transmission failures, it retries until the message is delivered, along with maintaining storage security. A reliable message service maintains the confidentiality of conversations between users.

Thus, detailed security measures are implemented for robust protection of users' personal information, such as messages, media files, and account details.

4. PROPOSED METHODOLOGY

A client-server architecture-based chat application is proposed in this paper. It consists of clients (i.e., senders and receivers) and servers (i.e., message server and user's server).

The conversation starts only when the recipient accepts the request of the sender.

Figure 1 depicts a generic layout of the proposed architecture. The different steps of conversation between two users are signing-up for new account, exchange of the encryption keys, user authentication, message transmission, and storage encryption.

4.1 Signing-up for new account

The users download and install the chat application using App Store, such as Google Play Store in Android devices. Once the application is installed, new users sign up with their mobile number or email ID, and create a password if asked. The existing users have to login using their credentials.

4.1.1 Server verification

A new account is created for new users. The server generates a unique identifier for the user to distinguish him from other individuals using the application. When the existing users have to login into their account, the credentials are received by the server for verification.

4.1.2 Security setup

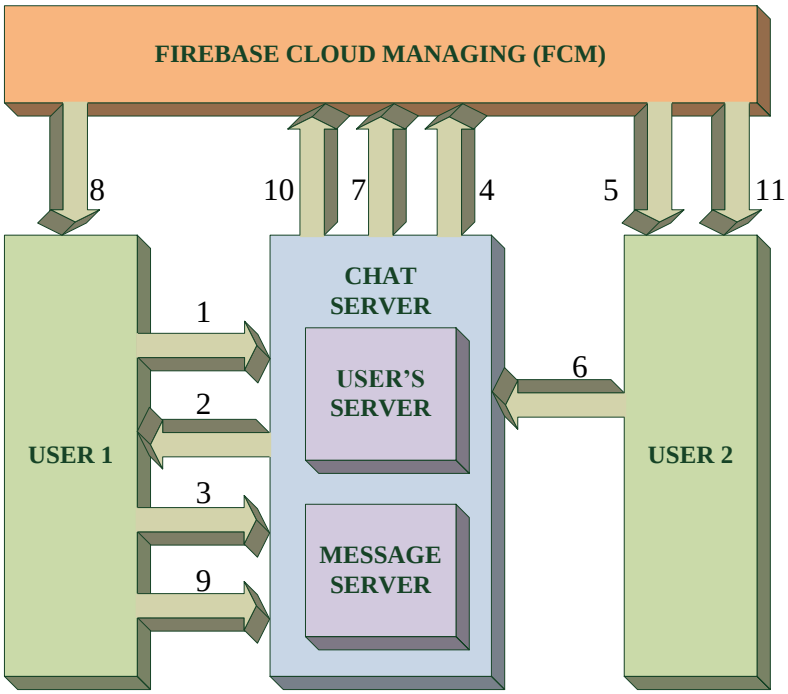
Transport Layer Security (TLS) generates an encryption key to ensure the security of conversation between sender's device and the server.

Further, the messages are only accessible to sender's and receiver's account or device. It is because the users are provided with a public key for encryption and decryption.

However, these messages can be accessed by the unauthorized individuals from users' devices. To prevent the intruder's action, a symmetric storage key is used to encrypt the data in users' devices.

4.1.3 Establishing connections

With the increasing number of social media applications, the users find their acquaintances and near ones to grow their network. When first user sends a request to second user and he accepts the request, both the users exchange their public keys. Thus, the conversation between both the users remains secure and private, because the message is encrypted and decrypted by them only.



1. User1 registers an account.
2. User1's account is created.
3. User1 sends a friend request to the chat server for User2.
4. The chat server forwards the friend request to FCM.
5. FCM delivers the friend request to User2.
6. User2 accepts the friend request and sends a notification to the chat server.
7. The chat server sends the notification of the accepted friend request to FCM.
8. FCM notifies User1 of the accepted friend request.
9. User1 sends a secured message to the chat server for User2.
10. The chat server forwards the secured message to FCM.
11. FCM delivers the secured message to User2.

Figure 1. Generic layout of proposed architecture of proposed application

4.2 Exchanging the keys

A secure communication environment is created for the users by introducing the key exchange feature. This feature is functional even if the receiver is offline. It is mostly beneficial in a group chat, where all the receivers are not available online at the instant. So, they will also receive the encryption key.

4.2.1 Generating keys

A random key is generated by the sender for encryption using the public key of the receiver. On the other hand, receiver generates another random key to encrypt it with the public key of the sender.

4.2.2 Sharing keys

The random keys generated by both sender and receiver are exchanged for encrypting the messages at both ends.

4.2.3 Encryption of messages

To ensure that only the receiver has access to their communications, i.e., decrypt the message using their matching private key, each user encrypts their messages using the key they produced.

4.3 User authentication

User authentication is the process of identifying the genuine users involved in the exchange of messages. It ensures the security and privacy of the users at both ends of communication. No server is involved in this process.

4.3.1 Initiation

An authenticated request is sent from sending end to receiving end.

4.3.2 Acceptance

The authentication process proceeds only when receiver accepts the request.

4.3.3 Secret keys and hash values

The secret keys are exchanged between the users and hash values are generated for identity verification.

4.3.4 Completion

Users start exchanging real communications and experience security as they realize that they are speaking with the right person after the successful completion of the verification procedure.

4.4 Exchanging messages

The users share personal data, such as medical records [12], research projects [13, 14], images [15], and videos [16], via chat applications User messages must be encrypted for secure communication to prevent unauthorized login of third parties. It is achieved by using the master keys.

4.4.1 Encryption

A master key is used for encryption of messages. To ensure security, every user has a different master key for outgoing and incoming communications.

4.4.2 Transmission

Messages transmitted across the network encrypted, so that only the intended receiver read them.

4.4.3 Decryption

The receiver ensures the communication stays private and safe by using their master key to decrypt the message. Thus, it ensures that the data is secure.

4.5 Local storage

The memory storage in a user's device having large or small media files, such as images, videos, audios, documents, and other private data is called local storage. It includes chat history, contact lists, passwords, master keys, and signature keys as well.

4.5.1 Confidential information

To enhance security, data including chat histories, contact lists, passwords, master keys, and signature keys are kept locally on the user's device rather than on the server.

4.5.2 Encryption

A symmetric storage key is used to encrypt all locally stored data. This guarantees that the data is not accessible by intruders or unauthorized individuals without the decryption key.

4.5.3 Access control

Local storage access is strictly regulated. However, users are recommended to secure their data by utilizing device-level security features like passwords, PINs, or biometric authentication.

Thus, the chat application guarantees a high degree of security and privacy for users by implementing thorough practices and security measures, monitoring their accounts, conversations, and stored data against breaches and unwanted access. It has many applications including medicine delivery system [17], academics [18], and emergencies such as accidents [19], In future, the chat application can be combined with Internet of Things to make it self-answerable, or block the spam contacts [20].

5. CONCLUSION AND FUTURE SCOPE

This paper presents the guidelines and criteria for creating a secure chat application. It analyzes the privacy and security features of the existing chat applications. It encourages to design an improved and more secure chat application by making small adjustments in the existing applications. Also, a generic layout of architecture for a secure chat application is proposed in this paper. In this paper, the importance of end-to-end encryption feature in an application is focused. The conversation between the sender and the receiver is secured as it is encrypted by sender's device and decrypted by receiver's device, without storing any information on the server.

This research can be extended by future researchers to include secure voice and video call features. Also, the technical details can be improved to improve the chat server's scalability and flexibility. Another feature that can be added is automatic blocking of spammers or intruders, based on the past activities of users.

REFERENCES

[1] Sabah, N., Kadhim, J.M., Dhannoon, B.N. (2017). Developing an end-to-end secure chat application. International Journal of Computer Science and Network Security, 17(11): 108-113. http://paper.ijcsns.org/07_book/201711/20171114.pdf.

[2] Read, A. (2016). The biggest internet phenomenon since the app store: How messaging apps are changing social media. Buffer. https://buffer.com/resources/messaging-apps/, accessed on Aug. 08, 2024.

[3] Dixon, S.J. (2024). Most popular mobile messaging apps 2024. Statista. https://www.statista.com/statistics/258749/most-popular-global-mobile-messenger-apps/, accessed on Aug. 27, 2024.

[4] Botha, J., Vant, W.C., Leenen, L. (2019). A comparison of chat applications in terms of security and privacy. In Proceedings of the 18th European Conference on Cyber Warfare and Security, University of Coimbra, Portugal, pp. 55-62. https://www.researchgate.net/publication/334537058.

[5] Merelo, J.J., Castillo, P.A., Mora, A.M., Barranco, F., Abbas, N., Guillén, A., Tsivitanidou, O. (2024). Chatbots and messaging platforms in the classroom: An analysis from the teacher's perspective. Education and Information Technologies, 29(2): 1903-1938. https://doi.org/10.1007/s10639-023-11703-x

[6] Valero, C., Pérez, J., Solera-Cotanilla, S., Vega-Barbas, M., Suarez-Tangil, G., Alvarez-Campana, M., López, G. (2023). Analysis of security and data control in smart personal assistants from the user's perspective. Future Generation Computer Systems, 144: 12-23. https://doi.org/10.1016/j.future.2023.02.009

[7] Davies, G.T., Faller, S., Gellert, K., Handirk, T., Hesse, J., Horváth, M., Jager, T. (2023). Security analysis of the WhatsApp end-to-end encrypted backup protocol. In Proceedings of 43rd Annual International Cryptology Conference, Santa Barbara, CA, USA, pp. 330-361. https://doi.org/10.1007/978-3-031-38551-3_11

[8] Cogliati, B.M., Ethan, J., Jha, A. (2023). Subverting Telegram's end-to-end encryption. IACR Transactions on Symmetric Cryptology, 2023(1): 5-40. https://doi.org/10.46586/tosc.v2023.i1.5-40

[9] Hasal, M., Nowaková, J., Ahmed Saghair, K., Abdulla, H., Snášel, V., Ogiela, L. (2021). Chatbots: Security, privacy, data protection, and social aspects. Concurrency and Computation: Practice and Experience. 33(19): e6426. https://doi.org/10.1002/cpe.6426

[10] Kadhim, M.F., Al-Janabi, A., Alhilali, A.H., Ali, N.S. (2022). Security approach for instant messaging applications: Viber as a case study. Indonesian Journal of Electrical Engineering and Computer Science, 26(2): 1109-1115. https://doi.org/10.11591/ijeecs.v26.i2.pp1109-1115

[11] Mahmud, M., Zannat, N., Nowshin, N. (2022). An elderly-centered design approach for mobile chat application. Journal of Information Systems and Digital

Technologies, 4(1): 147-172. https://doi.org/10.31436/jisdt.v4i1.263
[12] Juneja, A., Kumar, V., Kaur, M., Singh, D., Lee, H.N. (2024). XcepCovidNet: Deep neural networks-based COVID-19 diagnosis. Multimedia Tools and Applications. https://doi.org/10.1007/s11042-024-19046-6
[13] Juneja, A., Kumar, V., Singla, S.K. (2024). Single image dehazing using hybrid convolution neural network. Multimedia Tools and Applications, 83(13): 38355-38386. https://doi.org/10.1007/s11042-023-17132-9
[14] Juneja, A., Kumar, V., Singla, S.K. (2024). Desmogging of still images using residual regression network and morphological erosion. Multimedia Tools and Applications, 83(3): 7179-7214. https://doi.org/10.1007/s11042-023-15893-x
[15] Juneja, A., Singla, S.K., Kumar, V. (2023). HUDRS: Hazy unpaired dataset for road safety. The Visual Computer, 39(9): 3905-3922. https://doi.org/10.1007/s00371-022-02534-x
[16] Juneja, A., Kumar, V., Singla, S.K. (2023). Aethra-net: Single image and video dehazing using autoencoder. Journal of Visual Communication and Image Representation, 94: 103855. https://doi.org/10.1016/j.jvcir.2023.103855
[17] Bhat, M.W., Thippeswamy, V.S., Bhushan, H., Shrivastava, K., Sahoo, A.K. (2020). Secure online medicine delivery system. Review of Computer Engineering Studies, 7(3): 74-78. https://doi.org/10.18280/rces.070305
[18] Ma, J., Cui, L. (2019). Algorithm research on the analysis of college student score. Review of Computer Engineering Studies, 6(1): 6-10. https://doi.org/10.18280/rces.060102
[19] Shamie, M.M., Almustafa, M.M. (2021). Improving association rule mining using clustering-based data mining model for traffic accidents. Review of Computer Engineering Studies, 8(3): 65-70. https://doi.org/10.18280/rces.080301
[20] Herbadji, A., Herbadji, D., Labiad, A. (2020). Information gathering and controlling over the internet by Internet of Things (IoT). Review of Computer Engineering Studies, 7(3): 49-54. https://doi.org/10.18280/rces.070301