WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

SECURE CHAT USING ENCRYPTION

Abhay Singh Bais[1], Ashutosh Sao[2], Himanshu Jain[3], Shivam Sahu[4], Guide: Prof. Prageet Bajpai[5]

(B.Tech Students, Department of Computer Science and Engineering Shri Shankaracharya Technical Campus, Bhilai, Chhattisgarh)
(Professor, Department of Computer Science and Engineering Shri Shankaracharya Technical Campus, Bhilai,Chhattisgarh)

ABSTRACT :

In today's world the significance of online communication has increased dramatically leading to the need, for strong encryption. This project focuses on creating a chat application that incorporates end to end encryption to enhance data security. The encryption technique used combines RSA (Rivest, Shamir, Adleman) for exchange and AES (Advance Encryption Standard) for message encryption striking a balance between security and performance. A crucial aspect of the development process was ensuring uninterrupted encrypted chats without any delays. The user interface of the application was designed using Pythons tkinter library making it user friendly and easy to navigate. Throughout the development phase extensive testing was conducted to identify and address any vulnerabilities. The encryption employed by the chat application has proven to be highly resilient, against security threats. We built this application using Python showcases how advanced security measures can be seamlessly integrated into real time chat platforms while maintaining an interface.

Keywords:- Secure chat, end-to-end encryption, AES, RSA, secure communication.

INTRODUCTION

In the digital age, communication has become faster, more accessible, and increasingly dependent on internet-based platforms. Messaging applications such as WhatsApp, Telegram, and Signal have become essential tools for both personal and professional interaction. However, the widespread use of these platforms raises significant concerns about privacy, data integrity, and security. Cyber threats such as eavesdropping, data breaches, and man-in-the-middle attacks have become more prevalent, targeting sensitive conversations and confidential information. In this context, ensuring secure communication is no longer a luxury but a necessity. A secure chat system must address several critical aspects of communication security: confidentiality, integrity, authentication, and availability. Confidentiality ensures that only the intended recipient can read the message; integrity verifies that the message has not been altered in transit; authentication confirms the identity of the communicating parties. Many existing chat platforms claim to offer end-to-end encryption (E2EE), but not all of them are transparent in their implementation, and vulnerabilities often exist in key management or metadata protection. This paper introduces a secure chat application built using well-established cryptographic standards: Advanced Encryption Standard (AES) for symmetric message encryption and Rivest–Shamir–Adleman (RSA) for secure asymmetric key exchange.

METHODOLOGY

The architecture of the secure chat system is designed with modularity and security in mind. It comprises four primary components:

1. Client Interface: This is the frontend where users interact with the chat system. It allows users to send,receive, and view messages. The interface is designed to be intuitive while ensuring that messages are encrypted before they leave the device.

2. Encryption Engine: At the core of the client lies the encryption engine. This module uses AES (Advanced Encryption Standard) for encrypting the actual message content. For sharing keys securely, RSA encryption is used. Messages are never sent in plaintext; the encryption engine ensures end-to-end encryption.

3. Key Management: Each user has a unique public-private RSA key pair. During the initial setup or contact initiation, users exchange public keys. The AES session key used for each chat session is encrypted with the recipient's public key before transmission.

4. Server Component: The server acts as a message relay. It does not decrypt or store any user data in plaintext. It simply forwards the encrypted data packets between clients, thus minimizing server-side vulnerability. This architecture ensures that even if the server is compromised, the attacker cannot read any messages.
5. Cryptographic Techniques: The system implements a combination of symmetric and asymmetric encryption techniques to ensure the confidentiality, integrity, and authenticity of communications.

AES (Advanced Encryption Standard):
AES-256 is used for encrypting the message content. It is a symmetric encryption algorithm known for its speed and high level of security. Each session has a unique AES key to prevent key reuse.

RSA (Rivest-Shamir-Adleman):
RSA-2048 or higher is used for secure key exchange. Each user possesses a key pair (public and private). The AES session key is encrypted with the recipient's RSA public key and can
only be decrypted using their private key.

Digital Signatures:
To verify the authenticity of messages, digital signatures are implemented. The sender signs the hash of the message using their private RSA key. The recipient can then use the sender's public key to verify the signature, ensuring the message was not altered. Together, these cryptographic tools ensure strong end-to-end encryption and resistance to tampering.

RESULTS AND DISCUSSION

As a part of our project objective, we carried out a deep analysis of the Secure Chat application that we designed. Through this section, we provide an in-depth discussion of our results, giving insights into the empirical findings and potential implications of our chat solution.
The secure chat application was implemented using Python. Flask was used for the server-side logic, and the client interface was built using PyQt for desktop or web-based interfaces with HTML/CSS and JavaScript.
The key steps in the implementation are:
- Key Generation: Upon registration, each user generates an RSA key pair. Public keys are shared, while private keys are securely stored.
- Session Key Exchange: For each conversation, a new AES session key is created and encrypted with the recipient's public RSA key.
Message Handling: Messages are encrypted with AES and sent to the server. The server does not decrypt the messages but only forwards them.
Performance tests showed minimal latency for text-based messages. The encryption/decryption process was fast enough for real-time communication.
Security tests confirmed that data remains unreadable without the correct decryption keys. Penetration testing demonstrated resistance to common attacks like man-in-the-middle and replay attacks.
The implementation confirms the viability of the approach for secure real-time messaging in small- to medium-scale environments.

CONCLUSION

In this paper, we explored the design and implementation of a secure chat system that employs robust encryption methods to safeguard user communication. The system leverages both AES for fast and secure message encryption and RSA for secure key exchange, creating a reliable end-to-end encrypted environment. This combination ensures that messages remain confidential and cannot be accessed by unauthorized entities—even if intercepted during transmission .The proposed architecture emphasizes client-side encryption, meaning messages are encrypted before they leave the user's device and decrypted only by the intended recipient. This approach reduces the risk posed by compromised servers or network intrusions. Additionally, features such as digital signatures further enhance the system's trustworthiness by ensuring message integrity and authenticity .While the implementation demonstrates strong resistance to common attacks like MITM, replay, and brute-force, there are areas for future improvement. For instance, incorporating forward secrecy—where session keys are periodically updated and not derived from a static long-term key—would enhance the security posture further. Another advancement could include decentralized key management using blockchain or distributed ledger technologies to eliminate single points of failure.

REFRENCES

[1] Chatterjee, N., Chakraborty, S., Decosta, A., & Nath, A. (2018). Real-time Communication Application Based on Android Using Google Firebase. *International Journal of Advance Research in Computer Science and Management Studies*, 6(4). [Online] Available: www.ijarcsms.com.

[2] Sowah, R. A., Ofoli, A. R., Krakani, S. N., & Fiawoo, S. Y. (2017). Hardware design and web-based communication modules of a real-time multi-sensor fire detection and notification system using fuzzy logic. *IEEE Transactions on Industrial Applications*, 53(1), 559-566. https://doi.org/10.1109/TIA.2016.2613075.

[3] Anglano, C., Canonico, M., & Guazzone, M. (2016). Forensic analysis of the ChatSecure instant messaging application on android smartphones. *Digital Investigation*, 19, 44-59. https://doi.org/10.1016/j.diin.2016.10.001.

[4] Hegde, S., & Shah, S. (2015). A SURVEY ON THE LATEST WEB TECHNOLOGIES. [Online] Available: www.ijtra.com.

[5] G. Rovira Sánchez, "Implementation of a chat application for developers," 2017.

[6] A. Kumar and A. Singh, "Research paper on Group chatting Application." [Online]. Available: https://www.researchgate.net/publication/360483603. Accessed: 2023.

[7] Randall, N. (1997). Epilogue: The Soul of the Internet. In *The soul of internet: net gods, netizens, and the wiring of the world* (pp. 345-358). London: Computer Press.

[8] N. Sabah, J. M. Kadhim, and B. N. Dhannoon, "Developing an End-to-End Secure Chat Application,"Nov.2017.[Online].Available:https://www.researchgate.net/publication/322509087_Developing_an_End-to-End_Secure_Chat_Application.