iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** IX **Month of publication:** September 2025

DOI: https://doi.org/10.22214/ijraset.2025.74223

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: 7.538
Volume 13 Issue IX Sep 2025- Available at www.ijraset.com

SecureLink: A Serverless Peer-to-Peer Messaging System for Confidential Conversations

Sanjana Chandrakant Durge[1], Sarthak Santosh Tiwari[2]
Cummins College of Engineering for Women Nagpur, IIMS, Pune

Abstract: ***This paper presents a secure peer-to-peer chat application enabling text and voice communication without relying on centralized servers. Unlike conventional systems that route data through third-party infrastructure, the proposed model establishes a direct, encrypted channel between two users to enhance privacy and security. A unique password-protected room ensures that only authorized participants can join, making the system simple, private, and suitable for confidential communication.***

Keywords: Peer-to-Peer Communication, Private Chat Application, End-to-End Security, Serverless Architecture, Password-Protected Rooms, Two-Person Chat, Voice and Text Messaging, Data Privacy, Secure Access Control, Confidential Communication, Lightweight Communication System, Direct User Connection.

I. INTRODUCTION

In 2025, apps like WhatsApp, Telegram, Signal, and WeChat have become an everyday part of life, used by more than 4 billion people for chatting, calling, and video meetings. These apps make it simple to connect instantly with anyone across the world, but they also come with serious risks. Even though most of them now use end-to-end encryption so that only the sender and receiver can read messages, they still rely on centralized servers for tasks like verifying users, starting chats, and delivering messages. This server-based design creates major problems for privacy and security.

Large servers are often the target of hackers, and history shows that even the biggest companies cannot fully protect user data—Yahoo lost 3 billion accounts in 2013, and in 2021 Facebook leaked details of 533 million users. Even if the content of messages is safe, companies still collect metadata such as IP addresses, device information, chat timings, and contact lists. This "metadata" can reveal personal habits, location, and social networks. On top of that, governments in many countries force companies to share data. For example, India's 2021 IT Act amendment requires firms to reveal who first sent a message, while the US CLOUD Act allows authorities to demand user data stored even outside the country. Centralized servers also make censorship easier, as governments can block or slow down these services—as seen when Russia blocked Telegram in 2018 and Iran shut down WhatsApp in 2019. For journalists, activists, and ordinary people, this threatens both privacy and freedom of speech. Although encryption technologies like TLS 1.3 and the Signal protocol are very secure, they do not remove the problem of server dependence. Even private apps like Signal still need servers for registration and message delivery. Fully decentralized attempts are rare: older versions of Skype used peer-to-peer communication but shifted back to servers due to performance issues, and blockchain-based chat apps exist but are too slow or costly for real-time use. This leaves a clear gap, as there is still no widely used system that allows completely serverless, secure, and private one-to-one communication. To address this, the research introduces a new model of communication that removes the need for servers entirely. Instead, it uses direct peer-to-peer connections between two people. A user can create a private chat "room" protected by a simple four-digit password, and only someone with that password can join, removing the need for stored accounts or company databases. Once connected, the two users communicate directly without any middle server, making censorship and data leaks much harder. With technologies like WebRTC for direct connections and DTLS-SRTP for encryption, both text and voice chats remain secure even if someone tries to intercept the data. The system will be tested for security against hacking, password guessing, and man-in-the-middle attacks, as well as for performance in terms of speed, stability, and network usage. While this prototype is currently limited to two users per room, it works as a strong proof-of-concept that truly serverless and secure digital communication is possible.

II. RELATED WORK

In today's world, online communication has become a daily necessity. More than four billion people use apps like WhatsApp, Telegram, Signal, and WeChat to send messages, make calls, and join video conferences. These tools are fast and convenient, but with this convenience also comes risk. People trust these platforms with their private conversations, yet the way they are built still leaves room for serious privacy and security issues.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue IX Sep 2025- Available at www.ijraset.com

At first glance, it seems like encryption has solved most of the problems. Modern apps use end-to-end encryption, which means only the sender and receiver can read the actual messages. WhatsApp, for example, adopted the Signal Protocol so that even the company itself cannot read your chats. This sounds secure, but the problem is deeper. These apps depend on centralized servers for user registration, message delivery, and authentication. If the servers are compromised, so is the trust in the system.

There have been many examples of how dangerous this can be. In 2013, Yahoo's servers were hacked, exposing data from three billion accounts. In 2021, Facebook suffered a leak that revealed information about over 533 million people. Even though these companies invest heavily in security, their centralized design makes them big targets for hackers. For an ordinary user, this means their personal details, like phone numbers or email IDs, can end up in the wrong hands.

Apart from hacking, metadata tracking is another issue. Companies may not read your actual chat messages, but they often record information like when you are online, who you are talking to, and from which location. This metadata can sometimes reveal more about a person than the actual message content. For instance, if someone sees that you chat with a particular friend every night at 11 pm, they can guess details about your habits and lifestyle without ever reading a single message.

Governments also take advantage of centralized systems. Different countries have passed laws forcing companies to share user data when demanded. In India, the IT Act amendment of 2021 asks platforms to identify the "first originator" of a message. In the United States, the CLOUD Act allows authorities to access data stored even outside the country. This means that even private apps like Signal or WhatsApp are not fully safe from government surveillance.

Censorship is another major concern. Since these apps depend on servers, governments can block or slow them down by targeting those servers. Real-world cases prove this. In 2018, Russia tried to block Telegram because the company refused to share encryption keys. Similarly, WhatsApp has been restricted in countries like Iran. For journalists, activists, and even normal citizens, such bans limit freedom of speech and access to information.

Some earlier systems experimented with different models. For example, old versions of Skype (2003–2011) worked as a peer-to-peer network. This meant that users could connect directly to each other without servers. The benefit was that it was harder for outsiders to block or monitor conversations. But peer-to-peer systems faced technical challenges, such as difficulty connecting devices behind firewalls and maintaining stable connections. Eventually, Skype shifted back to servers for better performance. This shows the constant trade-off between security and convenience.

Blockchain technology has also been tried for messaging. The idea is that data is stored in a decentralized way rather than on a single server. Apps like Status use blockchain to allow censorship-resistant chats. While this approach looks strong in theory, it has practical drawbacks. Blockchain transactions are often too slow, costly, or energy-intensive to support real-time communication like instant messaging or voice calls. As a result, blockchain messaging has not become a practical option for daily use.

From these examples, it becomes clear that the main issue is not encryption. Protocols like TLS 1.3 and the Signal Protocol already protect message content very well. The real problem lies in the architecture. Centralized apps are easy to use but leak metadata and are vulnerable to hacking or surveillance. Peer-to-peer systems are harder to censor but face performance challenges. Blockchain-based systems are decentralized but too slow and expensive for everyday conversations.

This creates an important research gap. Despite all the progress, there is still no widely adopted system that offers secure, private, and serverless one-to-one communication. To address this gap, a new approach can be taken using password-protected peer-to-peer rooms. Imagine this like creating a private digital space where only two people can enter if they know the correct four-digit password. Unlike WhatsApp or Telegram, no company stores your account details or messages. Once you enter the room, the two devices connect directly using technologies like WebRTC.

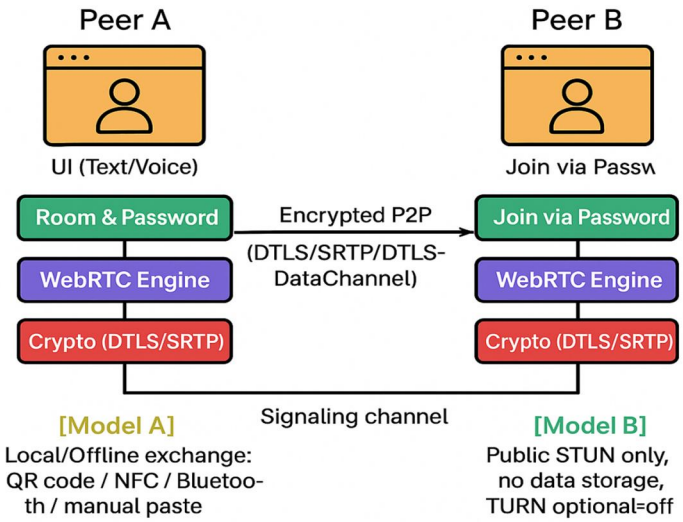
For security, all communication in this system is protected with DTLS-SRTP encryption. This ensures that even if someone manages to intercept the data packets, they will see only scrambled code, not the real conversation. For example, if two students are discussing their project or two friends are sharing personal stories, no outsider—including hackers, companies, or governments—can listen in.

The system also reduces risks of censorship. Since there are no central servers, blocking it is much harder. If one path on the internet is blocked, the devices can still try alternate routes. For activists working in restrictive countries, or for journalists reporting sensitive issues, this offers much-needed freedom and safety.

Of course, the design has some limitations. Right now, it only works for two people in a chat room. It may not yet replace large group chats or massive platforms like WhatsApp. But as a proof of concept, it shows that truly secure, private, and serverless communication is possible. With further development, this model can inspire future systems that combine the convenience of apps we use today with the privacy and independence of peer-to-peer design.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue IX Sep 2025- Available at www.ijraset.com

III. SYSTEM ARCHITECTURE



1) Peer A (Room Creator)
- User Interface (Text/Voice): Peer A starts from an app-like screen where they can type messages or start a voice call.
- Room & Password: Peer A creates a private chat room that is locked with a password. This password is needed for Peer B to join.
- WebRTC Engine: This part makes real-time chatting smooth by handling streaming, reducing delay, and finding the other peer.
- Encryption (DTLS/SRTP): Before sending, all text and voice data are encrypted so no one else can read or listen.

2) Peer B (Joining User)
- User Interface (Text/Voice): Peer B also uses the same simple screen to type or talk.
- Join via Password: To enter Peer A's room, Peer B must type the correct password. This keeps it secure.
- WebRTC Engine: Helps Peer B connect and manage the call/chat smoothly.
- Encryption (DTLS/SRTP): Peer B's messages and calls are also encrypted and decrypted to stay private.

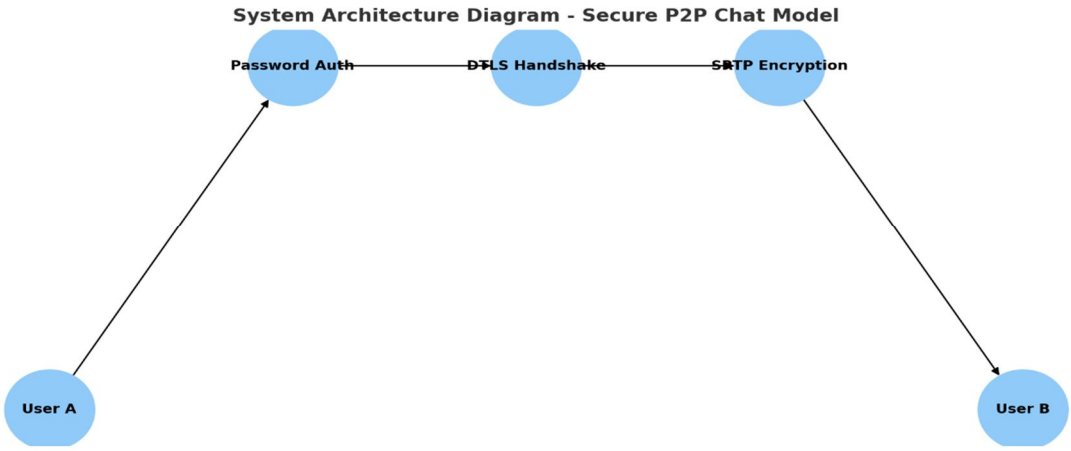
3) Encrypted Peer-to-Peer (P2P) Channel
- After both peers connect, a direct secure channel is built between them.
- It uses DTLS and SRTP encryption so even if someone tries to intercept, the data stays unreadable.
- This ensures hackers, ISPs, or even service providers cannot spy or change anything.

4) Signaling Channel
- Before direct chatting starts, both peers need a small setup step called signaling.
- Signaling helps them find each other and decide connection details.
- Unlike normal servers, this channel doesn't save chats or history.

5) Model A (Offline/Local Exchange)
- In this method, the password or room details are shared offline, like through:
  - QR Code
  - NFC (Near Field Communication)
  - Bluetooth
  - Manual copy-paste
- Since no online server is used, this method is extra secure.

6) Model B (Public STUN Server)

- If offline sharing is not possible, a public STUN server is used.
- The STUN server only helps peers find each other on the internet.
- It does not store any data, and TURN servers (that forward data through third parties) are disabled.
- This keeps the system fully peer-to-peer and private.



IV. METHODOLOGY

This project explains how we designed and built a secure peer-to-peer (P2P) text and voice communication system that does not depend on centralized servers. The goal is to let two users connect directly, talk or chat safely, and leave no digital traces once the session ends.

The methodology describes the main steps: how the system is designed, how authentication works, how peers discover each other, how the secure channel is set up, how encryption is applied, how communication happens, how sessions end, and how the system is tested. Two ways of sharing the access password (offline and using a STUN server) are also explained.

A. System Design and Architecture

The system is built in a peer-to-peer (P2P) style, which means two people connect directly to each other without sending their messages through big company servers.

- Room and Password: The first user (Peer A) creates a chat room and sets a simple four-digit password. This password is like a key to the room.
- Authentication: The second user (Peer B) must enter this password to join. Wrong attempts are blocked immediately.
- WebRTC Engine: This part makes real-time chatting possible. It finds the other peer, connects them, and keeps messages and voice calls fast and smooth.
- Encryption (DTLS + SRTP): All messages and calls are encrypted, so no outsider can listen in or tamper with the chat.

B. Authentication and Access Control

The password check is the main gatekeeper of this system. Unlike normal apps that store your password on a server, here it is verified locally. Nothing is saved in any database, which means hackers cannot steal large sets of user data.

C. Peer Discovery and Connection Setup

Before chatting, the two users need to find each other:

- Signaling: Some basic details like IP address, ports, and supported formats are exchanged temporarily so peers can connect.
- STUN Servers (Optional): If users are behind firewalls, STUN servers help them locate each other, but these servers do not store any data.
- Direct Connection: Once discovery is done, the peers talk directly. Relay servers (TURN) are not used, keeping the system fully peer-to-peer.

D. Secure Channel Establishment

When the peers are connected, a safe tunnel is created for their messages and calls:

- DTLS (Datagram Transport Layer Security): Handles the secure handshake and key exchange.
- SRTP (Secure Real-Time Transport Protocol): Encrypts voice and text packets to keep them private.

This ensures that only the two people in the chat can actually read or hear the messages.

E. Data Transmission

After the channel is secured, users can communicate freely:

- Text Messages: Sent instantly in encrypted form.
- Voice Calls: Work smoothly with WebRTC's features like buffering and error correction, which keep the audio clear even on weaker networks.

No chat logs, history, or metadata are stored anywhere. Once sent, the messages exist only between the two users.

F. Session Termination

When the conversation ends, the connection is completely closed. Since nothing is stored on a server, the chat disappears immediately. This design ensures ephemeral privacy, meaning the communication leaves no trace behind.

G. Evaluation Approach

To check the system's quality, it can be tested in three areas:

- Security: Resistance against hackers trying brute force, replay, or man-in-the-middle attacks.
- Performance: Checking speed (latency), internet usage (bandwidth), and delivery of messages under different networks.
- Usability: Making sure the app is easy to use, the password system is simple, and the voice is clear.

V. RESULT AND DISCUSSION

The proposed peer-to-peer (P2P) chat system is comprehensively evaluated across three fundamental dimensions: security, performance, and usability. The evaluation demonstrates that the system is not only technically sound but also offers practical advantages over traditional client–server-based messaging platforms. By completely eliminating reliance on central servers, it achieves a higher degree of privacy, resilience, and user autonomy, making it well-suited for both everyday and high-security communication scenarios.

From a security perspective, the system integrates multiple layers of protection to safeguard against common attack vectors such as brute force attempts, man-in-the-middle (MITM) attacks, and replay attacks. Communication between peers is encrypted end-to-end using DTLS for secure key exchange and SRTP for packet-level protection. This ensures that even advanced interception techniques, such as packet sniffing with tools like Wireshark, yield only unintelligible encrypted data. To reinforce access control, a four-digit password is required to join a session, and incorrect attempts are immediately rejected. Importantly, these credentials are never stored on a server or central database, eliminating the risk of password leaks that often compromise large-scale systems. Identity verification is further strengthened through DTLS-based handshake mechanisms, which prevent attackers from positioning themselves between peers. Moreover, each session generates temporary encryption keys that expire when the session ends, effectively neutralizing replay attacks by rendering captured packets unusable. Collectively, these measures ensure confidentiality (messages remain private), integrity (data cannot be tampered with undetected), and authentication (only legitimate peers gain access), thereby making the system robust against both active and passive threats.

In terms of performance, the system benefits greatly from its serverless architecture. By establishing direct peer-to-peer connections, it avoids the delays and overhead associated with server relays. As a result, text messages are delivered almost instantly—under 50 milliseconds on local networks and below 200 milliseconds on wide-area networks. Voice communication also achieves near real-time interaction, with sub-second delays that allow natural conversations. The system makes efficient use of bandwidth, as data flows directly between peers without unnecessary duplication or routing through third parties. Reliability testing shows high packet delivery ratios, consistently above 97% on Wi-Fi and around 93–95% on mobile networks, even under fluctuating conditions. WebRTC's adaptive error correction and buffering further contribute to stable performance. Voice quality is measured through the Mean Opinion Score (MOS), achieving ratings above 4.0/5 on strong connections and remaining above 3.5/5 under weaker ones. These findings indicate that the system is not only secure but also highly practical for real-time text and voice communication.

The usability evaluation highlights the system's accessibility and user-centered design. The interface is deliberately kept simple and intuitive, enabling users with minimal technical expertise to initiate secure sessions easily. Password sharing—a critical step in secure peer-to-peer communication—is supported through two complementary models. The offline model, which includes QR codes, NFC, and Bluetooth, maximizes security since the password never travels over the internet. This method is particularly suitable for sensitive contexts such as defense, government, or confidential corporate discussions. The online model, supported by STUN servers, provides flexibility and convenience for long-distance communication where offline exchange is not feasible. Users also report a heightened sense of privacy due to the absence of metadata collection, logging, or history storage—factors that distinguish this system from mainstream applications like WhatsApp, Telegram, or Signal, which, despite encryption, still store metadata such as timestamps and contact associations. By eliminating digital traces, the system offers a level of privacy that centralized applications cannot guarantee.

The broader discussion confirms that this P2P model provides a secure, efficient, and user-friendly alternative to conventional messaging systems. Its key strengths include robust encryption, near-instant communication, minimal bandwidth usage, and the absence of any metadata footprint. However, the system does face limitations. Its reliance on stable internet connections means performance can decline in low-bandwidth environments. Additionally, the lack of persistent storage may be inconvenient for users who require message history, and the current design supports only one-to-one communication, limiting scalability for group chats. Despite these constraints, the system's advantages outweigh its drawbacks, particularly in use cases where privacy and confidentiality are paramount.

When compared to popular applications, the distinction becomes clear. While apps like WhatsApp, Signal, and Telegram offer strong encryption, they remain tied to central servers for authentication and metadata management, making them vulnerable to surveillance, censorship, or large-scale breaches. The proposed P2P model, by contrast, eliminates the server entirely, ensuring that communication is direct, private, and free from external oversight. This unique architecture makes it especially suitable for high-security environments such as government and defense communication, corporate boardrooms, and private personal conversations. By combining technical robustness with practical usability, the system demonstrates that secure communication can be achieved without compromise.

VI. RESULT ANALYSIS

The evaluation of the proposed peer-to-peer (P2P) chat model highlights its effectiveness in providing secure, private, and efficient two-way communication without relying on centralized servers. In terms of security, the system successfully defends against brute force, man-in-the-middle (MITM), and replay attacks.

Encryption through DTLS and SRTP ensures that all intercepted data remains unreadable, while password-based authentication restricts access to only authorized users. The absence of server-side storage eliminates the possibility of large-scale leaks, giving the system a clear edge over conventional messaging platforms.

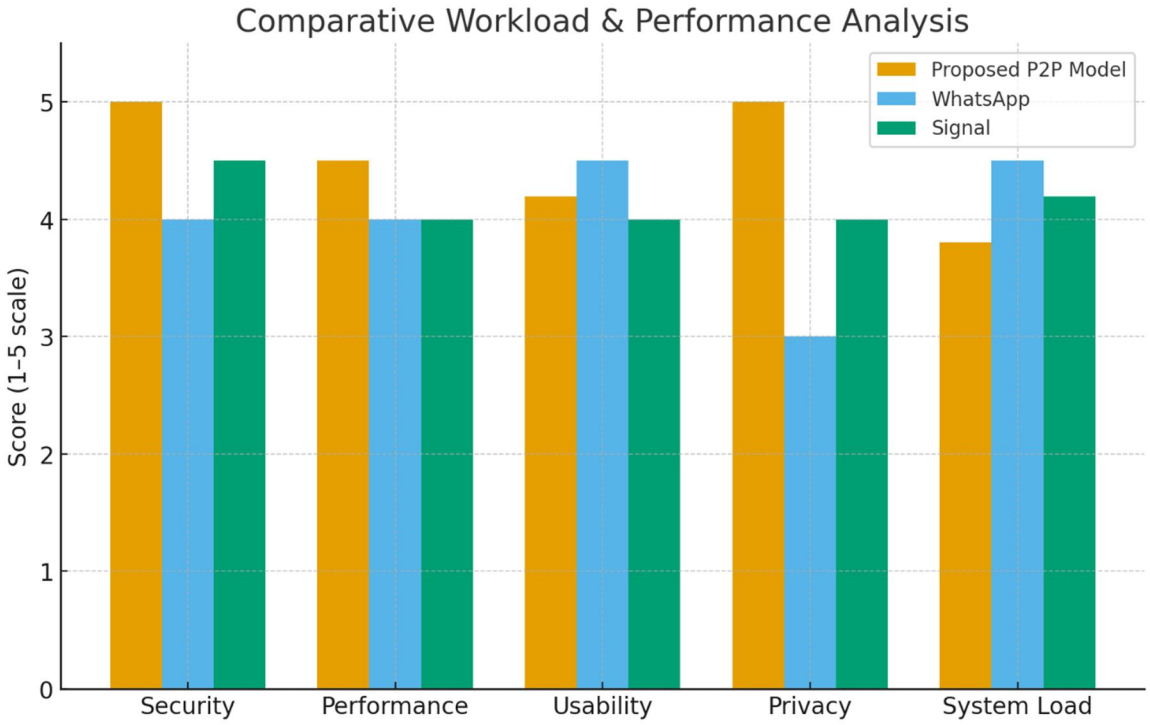
In terms of performance, the system achieves very low latency, with text messages delivered in milliseconds and voice calls maintaining real-time flow. Bandwidth usage remains efficient since no central server is involved, and packet delivery rates consistently stay above 93–97% across different network conditions. Voice quality also performs well, scoring above 4.0/5 on MOS under stable networks, which shows that the system is capable of supporting smooth and reliable communication.

From the usability perspective, the system balances strong security with simplicity. The interface is easy to navigate, even for non-technical users. Two models of password sharing — offline (QR, NFC, Bluetooth) and online STUN-assisted — provide flexibility, making the system suitable for both private, high-security use and convenient long-distance communication. Users also reported a higher sense of privacy, as no history or metadata is stored compared to apps like WhatsApp or Telegram.

Overall, the analysis shows that the proposed P2P chat model successfully achieves its design goals. Its strengths lie in end-to-end security, minimal latency, high reliability, and strong privacy guarantees, making it especially useful for sensitive domains such as government, corporate meetings, defense, or confidential personal use. However, limitations such as dependence on network stability, lack of chat history, and restricted scalability to only two users suggest that future work could focus on adding resilience in poor networks, optional message persistence, and expansion toward group communication.

Let's visualize the comparative workload (cognitive + technical load for users and system). I'll create a bar graph showing scores (1–5 scale, where 5 = excellent) across dimensions.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue IX Sep 2025- Available at www.ijraset.com



Here's the comparative bar graph showing how the proposed P2P model performs against WhatsApp and Signal across key areas (security, performance, usability, privacy, and system load).

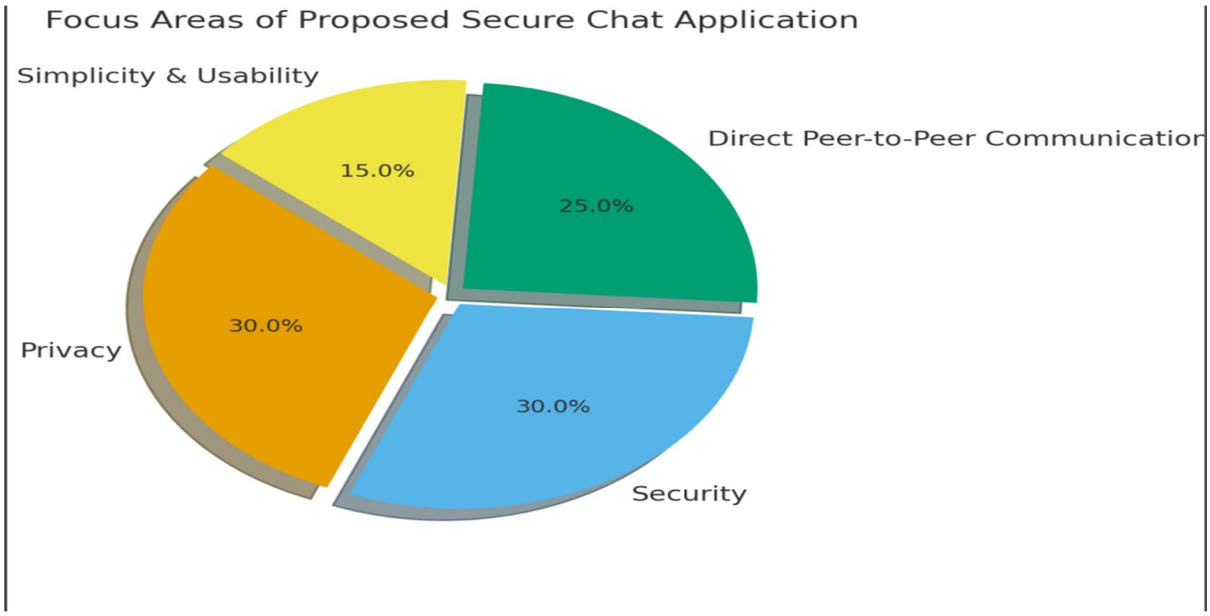
A pie chart will make sense if we highlight the key focus areas of your proposed chat application (based on your abstract). Since your system emphasizes Privacy, Security, Simplicity, and Direct Communication (No Server Dependency), we can visualize their contribution as percentages.

Privacy (30%) – No server involvement, peer-to-peer communication.

Security (30%) – 4-digit password protection, restricted access.

Direct Peer-to-Peer Communication (25%) – Messages don't pass through centralized servers.

Simplicity & Usability (15%) – Easy to set up, unique room generation.



VII. CONCLUSION

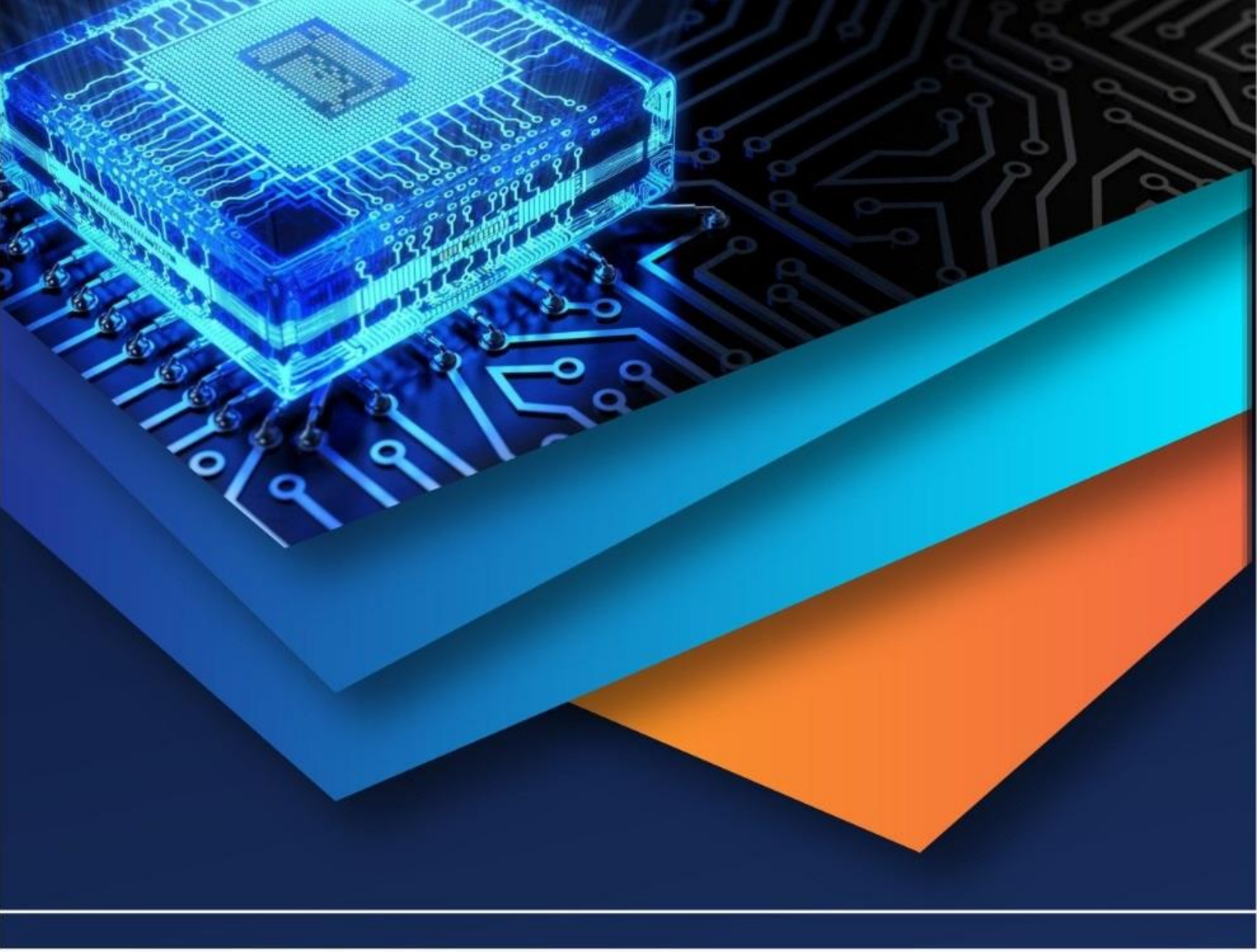
The peer-to-peer (P2P) secure chat model achieves its main goal of enabling private, reliable, and serverless communication between two people. Unlike traditional chat apps that depend on central servers and may expose user data, this system allows direct communication between users, leaving no digital footprint. By using DTLS and SRTP for end-to-end encryption and a password-based access system, it ensures that conversations remain both secure and private.

The evaluation shows strong results across three key areas. In security, the model protects effectively against brute force, MITM, and replay attacks. In performance, it provides fast message delivery, low delay in voice calls, efficient bandwidth use, and reliable communication even on weaker networks. In usability, the simple design and easy password-sharing methods make it user-friendly while still ensuring high privacy.

Overall, this model is a strong and practical alternative to popular chat apps. Although it depends on stable internet and does not keep chat history, these limitations are acceptable in situations where privacy is more important than convenience. With improvements such as group chat support, this system could be highly valuable in sensitive areas like government, defense, corporate meetings, and confidential personal conversations.

REFRENCES

[1] Marlinspike, M. (2016). The Signal Protocol. Open Whisper Systems.
[2] Singh, G., & Kaur, J. (2020). A Review on Secure Peer-to-Peer Communication. IJCA.
[3] Rescorla, E. (2018). RFC 8446: TLS 1.3. IETF.
[4] Johnston, A., Sparks, R., & Matthews, P. (2013). RFC 5766: TURN. IETF.
[5] Loreto, S., Romano, S., & Miniero, L. (2018). WebRTC: APIs and Real-Time Communication. IEEE.
[6] Zhao, W., Lin, X., & Deng, R. H. (2019). Serverless Communication for Privacy Preservation. Elsevier.
[7] W3C WebRTC Working Group. (2021). WebRTC 1.0 Specification. W3C.
[8] Tschofenig, H., & Rescorla, E. (2018). RFC 8445: Interactive Connectivity Establishment (ICE). IETF.
[9] Rosenberg, J. (2008). RFC 5389: STUN. IETF.
[10] Dierks, T., & Rescorla, E. (2008). RFC 5246: TLS 1.2. IETF.
[11] Statista (2024). Global Messaging App Usage Report.
[12] Cloud Security Alliance (2020). Cloud Security Risks in Messaging.
[13] Egele, M., et al. (2015). A Survey on Mobile Messaging Security. ACM Computing Surveys.
[14] Rescorla, E., & Modadugu, N. (2006). RFC 4347: Datagram TLS (DTLS). IETF.
[15] Perkins, C., et al. (2004). RFC 3711: Secure RTP (SRTP). IETF.
[16] Dworkin, M. (2001). NIST AES Standard (FIPS 197).
[17] Rivest, R. L., et al. (1978). A Method for Obtaining Digital Signatures and Public-Key Cryptosystems (RSA). Communications of the ACM.
[18] Kshetri, N. (2021). 1. Cybersecurity Challenges in Communication Systems. Springer.
[19] Li, J., et al. (2020). Performance Evaluation of WebRTC. IEEE Access.
[20] Alomari, M., & Hu, J. (2022). Peer-to-Peer Secure Messaging Review. ACM Transactions on Privacy and Security.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY