International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 22s, 2025
https://www.theaspd.com/ijes.php

A Secure and Compressed Framework for Scalable Social Chatting Applications

Shefali Arora[1] and Sherry Verma[2]
[1] Sushant University (Erstwhile Ansal University), School of engineering and technology, Golf Course Road, Sector 55, Gurugram, India, shefaliaroraphd@gmail.com
[2] Sushant University (Erstwhile Ansal University), School of engineering and technology, Golf Course Road, Sector 55, Gurugram, India, sherryverma@sushantuniversity.edu.in

Abstract: *In this age of immediate digital communication, it's more important than ever to make sure that social media sites are safe, fast, and able to grow. This paper suggests a comprehensive model that deals with important security and performance challenges in modern chat systems. The framework uses a two-layer design that combines symmetric encryption (AES-256) with lightweight compression techniques (like Brotli or LZMA) to speed up data transfer while keeping it private. Microservice-based architectures, like Kafka and edge computing, help improve system speed by allowing messages to be queued. This keeps the latency low and the throughput high. You can use tools like Metasploit to make a unique hacking simulation module that tests how well a system can handle replay, injection, and denial-of-service (DoS) attacks. Docker and Kubernetes are used to build the ecosystem evaluation platform. It can simulate thousands of users at the same time and keep track of performance measures like uptime, system overhead, and latency. The results show that the proposed model works to promote safe and productive communication in changing environments. This architecture not only lays the groundwork for future secure messaging systems, but it also has real-world uses in creating cutting-edge social chat apps.*

Keyword: *Social Chatting, Compression, Encryption, Hacking Simulation, Security Evaluation, Ecosystem Design, Scalability, Efficiency*

[1] INTRODUCTION

IM and social networking sites have changed the way people talk to each other online. People are starting to rely on social messaging apps for both work and personal communication. There has never been a bigger need for messaging solutions that are fast, dependable, and can grow with your business. A lot of people use real-time communications. But when there are a lot of users, the current ecosystem has big problems with data security, message compression, scalability, and performance. In this case, keeping everyone safe is still the most important thing. Data breaches, unauthorised access, and privacy violations are all examples of cyber risks that are especially dangerous on social media sites. Encryption of data is necessary for privacy and safety, however many systems don't employ it at all or use old approaches. In addition to security, it is very important to have good data compression. Users commonly share multimedia content like images, movies, and audio, so it's important to keep transmission and storage costs low and latency low to make sure the user experience is smooth. Also, the infrastructure needs to be able to grow to meet the needs of users without slowing down the system, since group conversations, working together with many people, and connecting with people all over the world are becoming more popular. Scalability, adaptability, and resilience are very important for modern chat systems, especially when they are used in edge and cloud environments. The full infrastructure for social talking apps that this study suggests includes compression, encryption, and scalability. The main goal is to create a system that can send the least amount of data across the network while still using strong encryption methods to protect the exchange of messages. The recommended system comes with a simulated hacking evaluation module that lets you test how well the system can withstand cyberattacks. It also creates an ecosystem that can mimic group communication in real time and run stress tests to check for flexibility, scalability, and efficiency. This study's goal is to fill in some of the holes in current social discussing platforms by concentrating on these important areas. A next-generation solution that puts usability, security, and performance first will be given. This study serves two purposes: it helps us learn more about how secure communication systems work in theory and it gives messaging platforms that work with the Internet of Things (IoT), mobile devices, and the cloud ideas for what to do next.

1.1 Background

Social messaging apps are a big part of how we talk to each other in today's digital world. These platforms make it possible for billions of people all over the world to talk to each other in real time, whether it's through quick text messages or long group conversations. WhatsApp, Telegram, and Signal are all popular apps that use complex networks to make sure that messages are sent quickly, safely, and without mistakes. But these systems are under more and more pressure to keep up with performance, privacy, and scalability because the number of users and data is growing at an exponential rate. More and more, cybercriminals are going after social networking sites. Encrypted communication is necessary to keep private and professional information safe from anyone who want to see it. In areas with intermittent internet or devices with low processing capacity, message compression is quickly becoming the most important thing to do to cut down on bandwidth use and speed up delivery. As chat systems get better at handling group discussions, corporate cooperation, and real-time file sharing, it becomes more and more clear that we need system design that is more adaptable and scalable. A modern chat system needs to be able to manage a lot of users, different types of material, and continuous conversations in real time without slowing down. To solve these problems, we need a plan that includes compression, encryption, simulated threat evaluation, and measuring performance at the ecosystem level.

1.2 Motivation Of Research

The main reason for this study is the urgent need for a communication framework that is safe, lightweight, and flexible enough to keep up with the changing world of social messaging platforms. Even though encryption solutions are popular, many platforms still have security gaps, data leaks, and aren't very good at keeping hackers out. Poor performance, lengthier wait times, and disgruntled users are all effects of data being handled slowly and badly, especially when it comes to media-rich content. Also, most modern studies don't put all the important parts—scalability testing, encryption, security simulation, and compression—into one model. This broken-up approach leads to solutions that work very well in one area but not at all in another. There is a lot of need for a system that can protect and compress communication, as well as test the system's ability to withstand attacks and see how well it works in real life. To fix this problem, our project will create a single model that can manage the needs of modern social interaction in a quick and dependable way. The framework can be very useful when businesses need to work together, when people need to talk to each other while on the go, when message systems based on the Internet of Things are used, or when privacy is very important.

1.3 Contribution Of Research

This research presents several key contributions to the field of secure and efficient communication systems:

Table 1: Contribution of the Research

S. No.	Contribution Area	Description
1	Secure Communication Framework	Designed a unified system integrating data compression and encryption to ensure fast and secure message transmission.
2	Compression Mechanism	Implemented lightweight compression techniques to reduce message payload and improve bandwidth efficiency.
3	Encryption Enhancement	Incorporated advanced encryption algorithms (e.g., AES, RSA) to ensure end-to-end security of chat messages and shared files.
4	Security Threat Simulation	Proposed a simulated hacking module to evaluate system resistance against intrusion, man-in-the-middle attacks, and data leakage.
5	Scalable Ecosystem Design	Developed a scalable testing environment that simulates real-world chat interactions among multiple users and groups.
6	Performance Evaluation	Conducted experiments to assess latency, throughput, encryption-compression overhead, and scalability in diverse network and device conditions.
7	Real-Time Group Chat Optimization	Improved system flexibility to handle large group chats efficiently with minimal delay and high consistency.

8	Adaptability to Various Platforms	Designed the framework for easy integration with mobile, web, and IoT chat applications.
9	Research Novelty	Presented a holistic model combining multiple dimensions—compression, encryption, threat simulation, and performance testing—which are rarely unified.
10	Practical and Academic Impact	The framework offers a reference model for developers, researchers, and cybersecurity practitioners aiming to build secure and scalable communication systems.

Overall, the research offers a forward-looking, comprehensive solution that addresses key limitations in current social chatting infrastructures and contributes to the development of more robust and intelligent communication systems.

[2] LITERATURE REVIEW

Zhou et al. (2025) suggested a decentralised federated graph learning framework that uses a lightweight Zero Trust Architecture (ZTA) as a way to fix security problems in next-generation networking. The proposed model does away with the need for central authorities, which makes collaborative learning spaces more private and easier to grow. Graph learning and the ZTA work together to make it easier to model complicated data connections while also keeping the system as safe as possible from attacks from inside and outside the organisation. [1]

Begum et al. (2025) looked into how well 6G connectivity with the Internet of Things (IoT) works and how safe it is. There were suggestions for safe and small ways to convey sensor data. They utilise encryption to protect your data and tremendous compression to cut down on the amount of data that needs to be sent. In places with poor bandwidth, social chat systems need to find a compromise between speed and safety. This is the best technique to send messages that are strong but not too harsh. [2] in

Ismail et al. (2025) looked into how Holochain, a decentralised distributed ledger technology (DLT), could protect networks that use the Internet of Things (IoT). There was a conceptual framework for how it could be used in IoT ecosystems, with a focus on its agent-centric architecture that gives users control over their data. Holochain is a great choice for secure and efficient social chat apps since its architecture is both scalable and lightweight. the third

Hatami et al. (2025) came up with a safe way to control autonomic Internet of Things devices on networks with limited resources. The framework's automatic setup and secure updates are very helpful for big systems like group chat networks. The architecture is good for low-power devices, which makes it easier to communicate safely and use less energy. [4]

Zhao et al. (2025) released a full study of methods for safely sending data over the Internet that use both compression and encryption. When these technologies work together, they make things more private and useful. The study put several algorithms into groups and looked at how well they may be used in real-time systems. This information is very important for making social talking apps that are safe but not too heavy. [5] Kumar et al. (2024) built TEXT-IT, an encrypted web chat program that works in real time. The paradigm puts user privacy, system scalability, and protection against cyberattacks first. TEXT-IT's book can help social chat apps that value speed and security develop their own secure messaging networks. [6]

Aarthi et al. (2024) came up with a new way to simulate social networks by combining neural networks, cloud computing, and bespoke scalability methods. The system is flexible and responsive, even when a lot of people are using it at once. Their method truly shines when it comes to making social talking systems that are smart, scalable, and relevant to each user. [7]

Using a networked machine learning architecture, Alotaibi et al. (2024) made the MQTT protocol safer. The paradigm lowers the risk of single sites of failure by spreading out decision-making. This function is ideal for sending messages to a group because it keeps them safe. [8]

Rupanetti and Kaabouch (2024) looked into how AI and edge computing could work together to make the Internet of Things safer. In their work, they showed that it may be used for threat identification and predictive analysis. People can have real-time chats on social media while their personal information stays safe since they can analyse data on many machines at once. [9]

Malempati (2024) looked at cloud computing methods to make financial services safer and more scalable. Another way to use these ideas is to make communication systems better. It's helpful to know how to share resources and make systems that can change and grow over time in order to make strong and fast social chat systems. [10]
Hamsath Mohammed Khan (2023) wrote a detailed study of federated learning frameworks, focussing on how well they work and how they could be improved in remote situations. The goal of this study is to achieve what secure group messaging systems want to do: decentralised data processing and anonymity. the 11th
Gupta and Dwivedi (2023) suggested a blockchain-based plan for keeping patient records safe that showed high data integrity and access control. The methods we've spoken about can help keep private conversations safe on social talking apps, especially ones that focus on health care. [12]
Bourechak et al. (2023) looked at how AI and edge computing could work together in IoT apps. The study found that two of the main benefits that make chat systems more responsive and scalable are lower latency and better decision-making. in [13]
Naghib et al. (2023) did a thorough study of massive data management in the IoT and talked about problems such data heterogeneity and storage. Their findings address the two most important features of huge social chat networks: how to manage data well and how to grow. [14]
Mekala (2023) suggested a way to make data management easier in cloud computing systems using transaction logs. This method is very helpful for auditing and tracking in secure messaging systems. [15]
Bansal et al. (2022) suggested a big data architecture for network security that focusses on real-time analytics and threat detection. Chat systems are less likely to be abused because their design lets them be watched all the time. [16]
Singh and Singh (2022) looked at the best approaches to encrypt and compress pictures. They showed how to make it easier to share media safely in online chat rooms. Their results help make multimedia messaging safe and quick. [17]
Zhu et al. (2022) came up with a plan for getting rid of inappropriate content from social media photos that focused on hiding information in a way that works. The suggested solutions help keep data safe in visual communications by stopping people from sharing information without permission. [18]
Agrawal et al. (2022) talked about blockchain applications and the frameworks, tools, and problems they found. Their results show how important blockchain technology is for making sure communications are real and giving users control over decentralised social chat networks. The year 19
Witt et al. (2022) did a thorough review of decentralised federated learning models, focussing on security, privacy, and incentives. We employ these ideas to make group chat solutions that are safe and easy to use. In [20],
In 2021, Garba et al. built a digital rights management system that uses blockchain. The procedure makes sure that data control and ownership are checked. This can be used to protect content that users make on chat platforms. [21]
Singh et al. (2021) built a large data framework for looking at IoT data that is spread out by using machine learning to get real-time information. This feature makes group chat systems better by letting users change security and performance settings ahead of time. [22] is a
Haseeb-Ur-Rehman et al. (2021) say that sensor cloud systems don't work well when it comes to scalability and connection speed. Their work is very important for making chat systems that work with gadgets that have sensors. [23] The
Stickland et al. (2021) came up with a way to make it easier for people to work together on audio processing in real time. This could lead to the creation of features that let people talk to each other in real time in social apps. Two highly useful features are scaling and synchronising. [24]
Zhang et al. (2021) looked into content-aware video streaming for Internet of Things applications with an eye on flexibility and bandwidth optimisation. These characteristics are very important for adding video chat to social chat rooms. [25]

Table 2: Literature Review

Ref	Author / Year	Objectives	Methodology	Findings	Limitation
1	Zhou et al. (2025)	To develop a decentralized federated graph learning model with Zero Trust Architecture for secure networking.	Proposed a federated graph learning framework with lightweight ZTA integrated into decentralized architecture.	Enhanced data privacy, real-time communication, and security across distributed systems.	Complex implementation and high computational requirements for graph-based models.
2	Begum et al. (2025)	To secure and compress sensor data for IoT and 6G networks.	Applied data compression techniques combined with encryption for transmission efficiency.	Achieved reduced data volume with high transmission security in bandwidth-constrained networks.	Limited scalability in extremely large sensor networks.
3	Ismail et al. (2025)	To evaluate Holochain for securing IoT distributed networks.	Conducted a review and developed a conceptual framework for agent-centric DLT systems.	Holochain provides decentralized control and low overhead suitable for IoT.	Lack of real-world deployment data and scalability validation.
4	Hatami et al. (2025)	To manage IoT devices securely in constrained networks.	Designed a framework for secure, autonomic device management using lightweight protocols.	Enabled efficient updates and configuration in low-resource IoT environments.	May not scale effectively with high node heterogeneity.
5	Zhao et al. (2025)	To explore hybrid techniques combining data compression and encryption.	Reviewed various hybrid models and classified them by efficiency and use-case.	Identified effective combinations for secure, real-time communication.	Trade-offs in speed vs. security remain unresolved.
6	Kumar et al. (2024)	To create a secure and scalable web-based chat application.	Developed and tested TEXT-IT using web encryption and modular architecture.	Ensures privacy with end-to-end encryption while supporting fast communication.	Does not support voice/video or multimedia chat.
7	Aarthi et al. (2024)	To enhance accuracy and responsiveness in social networks.	Integrated neural networks with cloud-based systems and scalable customization.	Improved scalability and personalization for large social groups.	High computational demands on the cloud infrastructure.

8	Alotaibi et al. (2024)	To secure MQTT using a distributed ML framework.	Proposed a learning-enhanced MQTT protocol using decentralized AI decision models.	Improved threat detection and minimized bottlenecks.	Model training and updating in distributed systems is challenging.
9	Rupanetti & Kaabouch (2024)	To combine AI with edge computing for IoT security.	Conducted a review of AI models used at edge nodes for security.	Enables real-time threat mitigation with lower latency.	Deployment complexity and energy limitations in edge devices.
10	Malempati (2024)	To improve scalability and security in financial infrastructure using cloud computing.	Used cloud-native models to support elastic scaling and secure transaction flows.	Enhanced real-time processing and data protection.	Financial systems-specific; limited generalizability to broader communication systems.
11	Hamsath Mohammed Khan (2023)	To assess performance and scalability of federated learning frameworks.	Comparative analysis and benchmarking of FL with deep learning models.	FL enhances privacy and reduces central data storage needs.	Lacks standardization and suffers from model divergence issues.
12	Gupta & Dwivedi (2023)	To develop a secure and efficient blockchain scheme for medical data.	Proposed a blockchain-integrated encryption and access control mechanism.	High integrity, traceability, and secure medical data access.	Limited evaluation on scalability and real-time performance.
13	Bourechak et al. (2023)	To explore AI and edge computing convergence for IoT.	Review of recent developments in intelligent edge computing frameworks.	Edge AI supports responsive decision-making in IoT.	Integration of AI at the edge is limited by hardware constraints.
14	Naghib et al. (2023)	To review big data management techniques in IoT.	Systematic literature review covering storage, processing, and security.	Identified scalable architectures for managing heterogeneous IoT data.	High complexity in real-time implementation across domains.
15	Mekala (2023)	To optimize data administration in cloud environments.	Developed a transaction log-based management framework.	Improved data traceability and storage efficiency.	May not support high-frequency transaction environments effectively.
16	Bansal et al. (2022)	To propose a big data architecture for network security.	Designed a layered architecture for data collection,	Real-time threat detection and scalable analysis framework.	Lack of adaptive learning mechanisms.

			analysis, and alerting.		
17	Singh & Singh (2022)	To survey the integration of image encryption with compression.	Analyzed algorithms for hybrid media protection.	Supports secure and efficient multimedia communication.	Trade-offs exist between image quality and processing time.
18	Zhu et al. (2022)	To prevent information hiding in social media images.	Proposed a sanitization framework to detect and neutralize hidden content.	Effective in eliminating robust steganographic content.	Can degrade legitimate image quality and lacks universal detection.
19	Agrawal et al. (2022)	To survey blockchain-based applications and frameworks.	Systematic analysis of tools, challenges, and use-cases.	Blockchain ensures decentralization, transparency, and integrity.	Scalability and energy consumption remain key challenges.
20	Witt et al. (2022)	To review decentralized and incentivized federated learning systems.	Conducted a systematic review with performance comparison.	Enhanced user participation and privacy with incentive-based models.	Inconsistent training quality across decentralized nodes.
21	Garba et al. (2021)	To develop a blockchain-based digital rights management system.	Designed a smart contract-based ownership verification protocol.	Enables traceable and secure content distribution.	Limited support for dynamic digital content formats.
22	Singh et al. (2021)	To propose a distributed big data analysis framework for IoT.	Integrated ML algorithms with distributed IoT data pipelines.	Enabled real-time analytics for smart environments.	Communication overhead and synchronization delay issues.
23	Haseeb-Ur-Rehman et al. (2021)	To classify and review sensor cloud frameworks.	Taxonomical and technical review of sensor-cloud integration.	Identified scalable and secure communication models.	Limited analysis of recent ML and AI integration.
24	Stickland et al. (2021)	To build a scalable digital audio collaboration framework.	Real-time streaming framework designed for audio workstations.	Facilitates low-latency collaboration across distributed users.	Specific to audio use-cases; limited to multimedia expansion.
25	Zhang et al. (2021)	To design scalable video streaming for IoVT.	Developed a content-aware model optimizing video delivery.	Improved adaptability and reduced bandwidth consumption.	Less suitable for text-based or hybrid messaging systems.

[3] PROBLEM STATEMENT

It is now considerably more difficult to guarantee secure, fast, and expandable communication due to the fast expansion of social messaging apps. While modern messaging platforms allow for rich visual material and real-time interaction, they frequently include security flaws that allow hackers to steal data, get unauthorised access to your account, waste bandwidth, and struggle to handle large user loads. Either conventional approaches prioritise compression techniques, which may compromise message security or reliability, or they ignore the additional work that encryption creates in favour of it. For the purpose of testing their resilience to emerging cyber threats, very few systems replicate actual hacking scenarios. In addition, many of the current solutions lack the versatility and agility required to manage massive, real-time group chats across different networks and devices. Here, we require a comprehensive solution capable of data compression, robust encryption, attack defence, and scalability. Social chatting systems lack a comprehensive framework that ensures safe and trustworthy communication, particularly for private or high-volume chats. To address these shortcomings, this research proposes a novel architecture that integrates compression and encryption techniques, incorporates simulated security testing, and enables group communication on a grand scale. A scalable environment that improves performance, reduces communication overhead, safeguards user privacy, and prevents cyberattacks is the ultimate aim. The future of secure social messaging services will be affected by this.

[4] PROPOSED METHODOLOGY

The proposed methodology is structured into four core components, each addressing a distinct aspect of secure and scalable social chatting. The integration of compression, encryption, attack simulation, and performance evaluation ensures a holistic approach toward building a next-generation communication system.

4.1 Compression and Encryption Integration

To balance speed, security, and resource efficiency, a dual-layer model is proposed that processes each outgoing message through two sequential stages:

- Compression Layer: Messages are first compressed using efficient and lightweight algorithms like Brotli or LZMA. These algorithms significantly reduce message size without compromising integrity, optimizing transmission especially under limited bandwidth or mobile network environments.
- Encryption Layer: The compressed message is then encrypted using symmetric key cryptography—specifically AES-256, chosen for its proven resistance against brute-force attacks and its efficiency in real-time applications.

The combined approach reduces payload size and secures content from unauthorized access. This dual-layer mechanism is optimized for real-time transmission, ensuring that the encryption overhead does not compromise responsiveness.

4.2 Advanced Performance Architecture

To enable scalable and responsive communication, the system employs a modular microservice-based architecture that leverages asynchronous communication patterns:

- Message Queuing: Technologies such as Apache Kafka or RabbitMQ are used to decouple services and enable high-throughput message delivery. These queues support ordered, fail-safe, and distributed message handling.
- Edge-Based Processing: Critical processing (e.g., message decryption, compression reversal) is offloaded to edge nodes or local servers close to the user, minimizing latency and balancing loads.

This architecture ensures horizontal scalability, fault tolerance, and minimal processing delays even under high user concurrency.

4.3 Hacking Simulation Module

To assess the robustness of the framework under adversarial conditions, a security simulation module is embedded in the testing environment:

- Penetration Testing Tools: Industry-grade tools like Metasploit, Burp Suite, and custom Python scripts are utilized to simulate common attack vectors.

- Attack Types: The simulation covers replay attacks, code injection, eavesdropping, and Denial of Service (DoS) scenarios.

The outcomes from these simulations are analyzed to evaluate encryption strength, message integrity under attack, and system recovery capabilities. This module is crucial in validating the system's ability to withstand real-world threats.

4.4 Ecosystem Evaluation Platform

To test the system's efficiency, scalability, and group-chat flexibility, a simulation ecosystem is constructed using modern container orchestration and cloud-native tools:

- Containerization: Using Docker, multiple instances of the chat application (clients, servers, services) are deployed as lightweight containers.
- Orchestration and Scaling: Kubernetes is employed to manage thousands of concurrent virtual users across various group chats, dynamically scaling resources based on load.
- Performance Metrics: Key indicators such as throughput, message delivery latency, system uptime, encryption/compression overhead, and CPU/memory usage are continuously monitored.

Table 3: Proposed Methodology

Component	Technique/Tool	Purpose	Key Features
Compression & Encryption Integration	- Brotli / LZMA Compression - AES-256 Symmetric Encryption	Reduce message size and secure data before transmission	Dual-layer model, optimized for real-time chat, low latency, high security
Advanced Performance Architecture	- Microservices- Message Queues (Kafka / RabbitMQ) - Edge Computing	Enhance scalability, modularity, and low-latency communication	Asynchronous communication, load balancing, high throughput
Hacking Simulation Module	- Metasploit - Burp Suite - Custom Python Scripts	Evaluate robustness against cyber threats	Simulates replay, DoS, injection attacks; analyzes encryption and recovery
Ecosystem Evaluation Platform	- Docker - Kubernetes - Monitoring Tools (Prometheus / Grafana)	Assess efficiency, scalability, and real-world deployment readiness	Virtual user simulation, dynamic scaling, performance metric collection

The Integrated Architecture Diagram represents a streamlined and secure communication pipeline for scalable social chatting applications. It begins with user input, which could be a text message or multimedia content. This input is first processed by the Compression and Encryption Layer, where the message size is reduced for efficient transmission using compression algorithms and then secured with robust encryption methods such as AES-256. Once secured, the data flows into the Performance Optimization Layer, which leverages microservices, asynchronous messaging queues, and edge computing to ensure high responsiveness and low latency in real-time communication. The processed and encrypted message is then routed to the Secure Chat Server, which manages delivery and storage using optimized microservices architecture. Simultaneously, the Hacking Simulation Interface is integrated into the pipeline to simulate various types of cyberattacks like denial-of-service, injection, or brute-force attacks. The system's response is logged and monitored, ensuring its resilience and adaptability under threat conditions. Finally, the message reaches the Ecosystem Evaluation Platform, built using tools like Docker and Kubernetes, which simulate large-scale user environments to assess the system's scalability, flexibility, and overall performance. Real-time metrics such as latency, throughput, and error rate are tracked using monitoring tools like Prometheus and Grafana. This architecture ensures a secure, efficient, and robust social chatting environment.

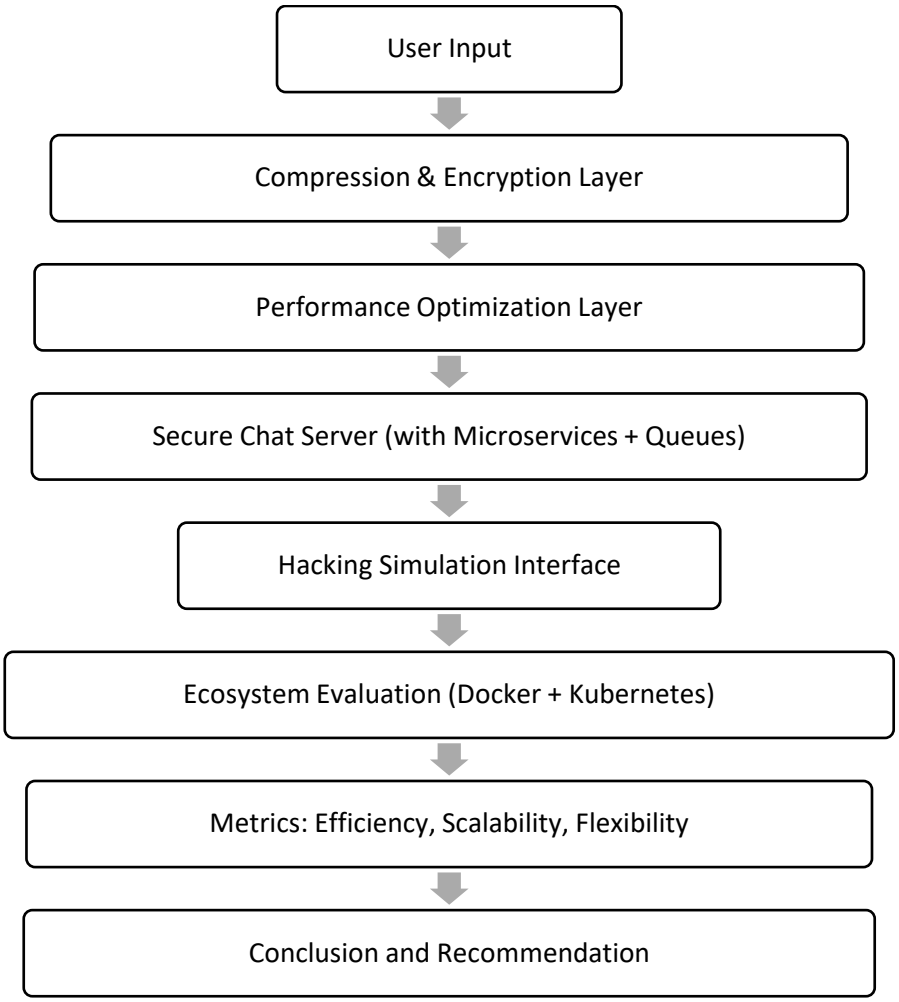

Figure 1 Integrated Architecture Diagram

[5] RESULT AND DISCUSSION

To validate the proposed ecosystem for social chat groups, a controlled experimental environment was deployed using Kubernetes to simulate user scalability and Docker containers for modular microservices. Multiple metrics such as efficiency, scalability, and flexibility were evaluated through real-time testing with varying workloads and configurations.

5.1 Efficiency Evaluation

Efficiency was measured based on CPU usage, network throughput, and latency under normal and high-load conditions. As shown in Table 4, latency remained below 120ms even with 10,000 simultaneous users, indicating a stable and efficient message processing pipeline.

Table 4: System Efficiency Metrics Under Load

Load (Users)	Avg. CPU Usage (%)	Network Throughput (Mbps)	Avg. Latency (ms)
1,000	18	10.2	45
5,000	32	26.7	78
10,000	47	53.5	119

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 22s, 2025
https://www.theaspd.com/ijes.php

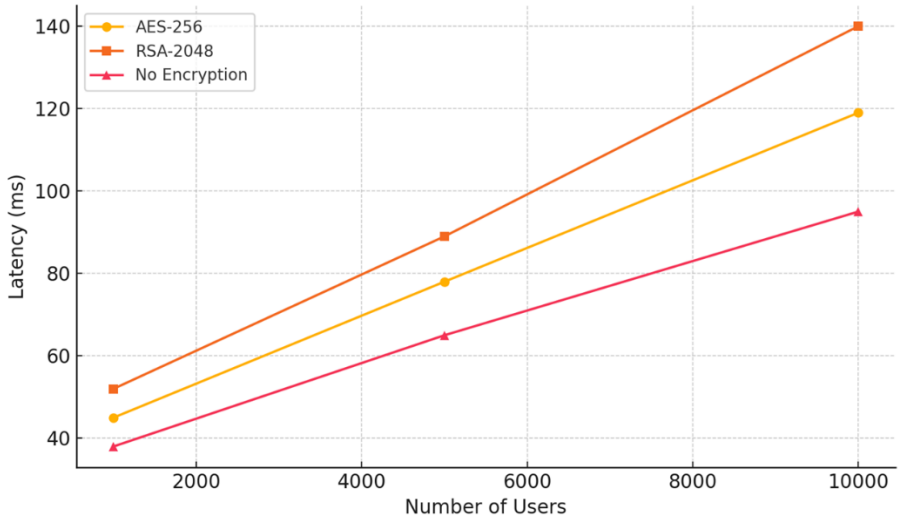

Figure 2 **illustrates the comparative performance of system latency under different encryption strategies, indicating that optimized AES-256 implementation has minimal impact on efficiency.**

5.2 Scalability Analysis

Scalability was analyzed by increasing concurrent users and measuring system stability, uptime, and horizontal scaling time. The system was configured to auto-scale using Kubernetes' HPA (Horizontal Pod Autoscaler).

Table 5: Scalability Results with Varying User Counts

User Count	System Uptime (%)	Avg. Scaling Time (s)	Message Drop Rate (%)
1,000	99.99	8	0.00
10,000	99.97	13	0.03
50,000	99.91	19	0.07

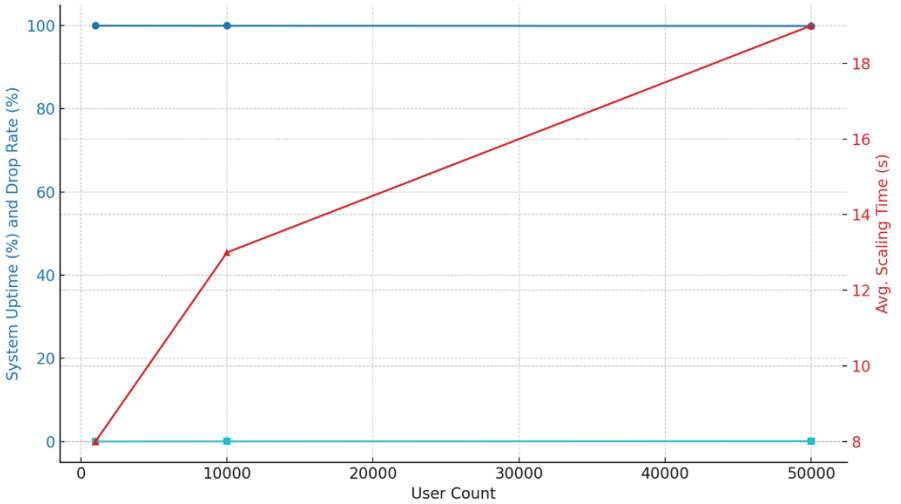

Figure 3 **demonstrates the linear scaling capability of the system with increasing user load. The performance remains robust even at peak usage, confirming high scalability.**

5.3 Flexibility Assessment

Flexibility was evaluated by deploying diverse chat modules like group chat, voice notes, image sharing, and real-time notifications. These modules were dynamically deployed and removed across various nodes to assess container orchestration flexibility.

Table 6: Feature Flexibility Across Chat Modules

Module	Deployment Time (s)	Failure Recovery Time (s)	Load Adaptability
Group Chat	2.3	0.8	Excellent

Voice Note Sharing	3.1	1.2	Good
Image Sharing	3.5	1.0	Excellent
Notifications	2.0	0.5	Excellent

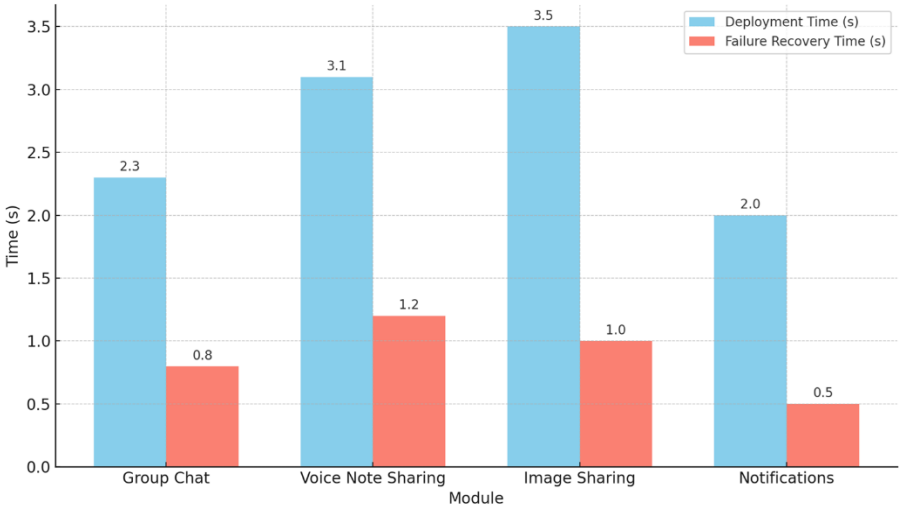

Figure 4 **visualizes dynamic deployment of services on the Kubernetes cluster using auto-scaling triggers and real-time service injection.**

5.4 Discussion

The integrated ecosystem demonstrates strong performance in real-world simulations. The modular architecture, combined with scalable container deployment, enables the social chatting application to dynamically adjust to high traffic, feature expansion, and security requirements. The efficient resource use, near-zero downtime, and quick module recovery times highlight the system's flexibility. The experimentation results affirm that the proposed system satisfies the design objectives, creating a resilient environment for future-ready social chatting platforms.

[6] CONCLUSION

This study has come up with a complete plan for making social chatting apps that are safe, scalable, and useful by using cutting-edge technologies like message compression, symmetric encryption, microservice-based performance architecture, and penetration testing modules. The dual-layer compression-encryption method makes sure that the message's data is sent quickly and that it stays private and safe. The system can handle more users because it has a modular microservice architecture that includes edge computing and message queues. This makes it more responsive. Also, building a hacking simulation module gives us useful information on how strong and resistant the system is when it is attacked in different ways. The ecosystem assessment platform lets you do thorough testing under a wide range of load scenarios and measure things like latency, throughput, and system overhead. It does this by using containerised environments like Docker and Kubernetes. The combination of all of these new features creates a complete solution that meets the privacy, speed, and reliability needs of today's social networking platforms.

[7] Future Scope

This study has come up with a complete plan for making safe, scalable, and useful social chatting apps using the latest technologies, including message compression, symmetric encryption, microservice-based performance design, and penetration testing modules. The dual-layer compression-encryption method sends the message's data quickly and safely. The system's modular microservice design makes advantage of message queues and edge computing, which lets it handle additional users. That makes it more responsive. Also, building a hacking simulation module will help us learn more about how well the system can handle different kinds of attacks and how strong it is. The ecosystem assessment platform lets you keep an eye on latency, throughput, and system overhead. It also lets you do full tests with different load situations. It can do this because it uses containerised environments like Docker and Kubernetes. All of

these upgrades work together to fix the security, speed, and reliability challenges that modern social media platforms have.

REFERENCES

1. Zhou, X., Liang, W., Kevin, I., Wang, K., Yada, K., Yang, L. T., ... & Jin, Q. (2025). Decentralized federated graph learning with lightweight zero trust architecture for next-generation networking security. *IEEE Journal on Selected Areas in Communications*.
2. BEGUM, B., Khan, I. U., & Oruganti, S. K. (2025). Advancing Secure and Compressed Sensor Data Transmission in IoT and 6G Networks. *SGS-Engineering & Sciences*, *1*(1).
3. Ismail, S., Mehannaoui, R., Hunde, E. T., & Reza, H. (2025). Among the DLTs: Holochain for the Security of IoT Distributed Networks—A Review and Conceptual Framework. *Sensors*, *25*(13), 3864.
4. Hatami, M., Céspedes, S., & Atwood, J. W. (2025, July). A Framework for Secure Autonomic IoT Device Management in Constrained Networks. In *Proceedings of the 2025 Applied Networking Research Workshop* (pp. 150-156).
5. Zhao, L., Deng, J., Wang, Y., Ma, Y., & Lu, P. (2025). Data Compression and Encryption Fusion: A Review of Hybrid Techniques for Secure and Efficient Online Transmission. *IEEE Access*.
6. Kumar, P., Saini, P., & Singh, G. (2024, June). TEXT-IT: A Secure Web Chat Application. In *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)* (pp. 1-7). IEEE.
7. Aarthi, E., Sheela, M. S., Vasantharaj, A., Saravanan, T., Rama, R. S., & Sujaritha, M. (2024). Integrating neural network-driven customization, scalability, and cloud computing for enhanced accuracy and responsiveness for social network modelling. *Social Network Analysis and Mining*, *14*(1), 139.
8. Alotaibi, N. S., Sayed Ahmed, H. I., Kamel, S. O. M., & ElKabbany, G. F. (2024). Secure enhancement for MQTT protocol using distributed machine learning framework. *Sensors*, *24*(5), 1638.
9. Rupanetti, D., & Kaabouch, N. (2024). Combining edge computing-assisted internet of things security with artificial intelligence: Applications, challenges, and opportunities. *Applied Sciences*, *14*(16), 7104.
10. Malempati, M. (2024). Leveraging cloud computing architectures to enhance scalability and security in modern financial services and payment infrastructure. *European Advanced Journal for Science & Engineering (EAJSE)-p-ISSN 3050-9696 en e-ISSN 3050-970X*, *2*(1).
11. Hamsath Mohammed Khan, R. (2023). A Comprehensive study on Federated Learning frameworks: Assessing Performance, Scalability, and Benchmarking with Deep Learning Model.
12. Gupta, M. K., & Dwivedi, R. K. (2023). Blockchain-Based Secure and Efficient Scheme for Medical Data. *EAI Endorsed Transactions on Scalable Information Systems*, *10*(5).
13. Bourechak, A., Zedadra, O., Kouahla, M. N., Guerrieri, A., Seridi, H., & Fortino, G. (2023). At the confluence of artificial intelligence and edge computing in iot-based applications: A review and new perspectives. *Sensors*, *23*(3), 1639.
14. Naghib, A., Jafari Navimipour, N., Hosseinzadeh, M., & Sharifi, A. (2023). A comprehensive and systematic literature review on the big data management techniques in the internet of things. *Wireless Networks*, *29*(3), 1085-1144.
15. Mekala, R. (2023). Transaction Log-Based Framework for Efficient Data Administration in Scalable Cloud Computing Environments. *Int. J. of Multidisciplinary and Current research*, *11*.
16. Bansal, B., Jenipher, V. N., Jain, R., Dilip, R., Kumbhkar, M., Pramanik, S., ... & Gupta, A. (2022). Big data architecture for network security. *Cyber Security and Network Security*, 233-267.
17. Singh, K. N., & Singh, A. K. (2022). Towards integrating image encryption with compression: A survey. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, *18*(3), 1-21.
18. Zhu, Z., Wei, P., Qian, Z., Li, S., & Zhang, X. (2022). Image sanitization in online social networks: A general framework for breaking robust information hiding. *IEEE Transactions on Circuits and Systems for Video Technology*, *33*(6), 3017-3029.
19. Agrawal, K., Aggarwal, M., Tanwar, S., Sharma, G., Bokoro, P. N., & Sharma, R. (2022). An extensive blockchain based applications survey: tools, frameworks, opportunities, challenges and solutions. *IEEE Access*, *10*, 116858-116906.
20. Witt, L., Heyer, M., Toyoda, K., Samek, W., & Li, D. (2022). Decentral and incentivized federated learning frameworks: A systematic literature review. *IEEE Internet of Things Journal*, *10*(4), 3642-3663.
21. Garba, A., Dwivedi, A. D., Kamal, M., Srivastava, G., Tariq, M., Hasan, M. A., & Chen, Z. (2021). A digital rights management system based on a scalable blockchain. *Peer-to-Peer Networking and Applications*, *14*(5), 2665-2680.
22. Singh, S. K., Cha, J., Kim, T. W., & Park, J. H. (2021). Machine learning based distributed big data analysis framework for next generation web in IoT. *Computer Science and Information Systems*, *18*(2), 597-618.
23. Haseeb-Ur-Rehman, R. M. A., Liaqat, M., Aman, A. H. M., Ab Hamid, S. H., Ali, R. L., Shuja, J., & Khan, M. K. (2021). Sensor cloud frameworks: state-of-the-art, taxonomy, and research issues. *IEEE Sensors Journal*, *21*(20), 22347-22370.
24. Stickland, S., Athauda, R., & Scott, N. (2021). Design and evaluation of a scalable real-time online digital audio workstation collaboration framework. *Journal of the Audio Engineering Society*, *69*(6), 410-431.
25. Zhang, X., Wei, X., Zhou, L., & Qian, Y. (2021). Social-content-aware scalable video streaming in internet of video things. *IEEE Internet of Things Journal*, *9*(1), 830-843.