IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 12, Issue 4, April 2024

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.379

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 4, April 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1204300 |

Enhancing Global Communication with Sync Sphere, a Secure and Multifunctional Chat Program

Hemanshu Vaidya, Soham Barve, Paritosh Gogate, Aditya Gode, Shreeya Ranwadkar,

Prof. Pranati Waghodekar

School of Computer Science, MIT World Peace University, Pune, India

School of Computer Science, MIT World Peace University, Pune, India

School of Computer Science, MIT World Peace University, Pune, India

School of Computer Science, MIT World Peace University, Pune, India

School of Computer Science, MIT World Peace University, Pune, India

Assistant Professor, Department of CET, MIT World Peace University, Pune, India

ABSTRACT: Our online chat program – Sync Sphere - is made to offer a safe, easy, and feature-rich communication experience. Modern features are incorporated to guarantee users' privacy, ease, and accessibility for a range of linguistic and communication needs.

The application protects user interactions from unauthorized access by prioritizing security with end-to-end encryption. A primary priority is message security, which makes sure that all transferred data is safe from any attacks.

The application facilitates real-time multilingual communication between users by supporting dynamic language translation, which improves worldwide connectivity. It also has sophisticated text processing features, like the ability to distinguish between differences in capital and lowercase letters to enhance message clarity. Participant lists streamline user engagement by enabling users to efficiently manage and communicate with chat participants. The ability to share files between discussions makes it easier for people to collaborate by allowing the exchange of documents.

The interface provides a user-friendly experience by being interactive and intuitively built. With user authentication methods, users may safely authenticate, search messages, and traverse discussions with ease.

In conclusion, our online chat program is a complete answer to the demands of contemporary communication, fusing cutting-edge functionalities with strong security measures to provide a smooth and entertaining talking experience for every user.

KEYWORDS: End-to-end encryption , Message security, Dynamic language translation , User authentication , File sharing

I. INTRODUCTION

Preamble - Web-based chat programs have grown essential to contemporary communication in recent years, allowing for cooperation and rapid messaging across a wide range of user groups. The demand for safe, effective, feature-rich platforms that meet international communication needs has fueled the development of these apps. The creation and assessment of an online chat program that integrates many crucial components to improve security and user experience is the main goal of this study.

End-to-end encryption is a critical feature that the program addresses. It guarantees that messages are encrypted and that only the intended receivers can decrypt them, protecting discussions from unwanted access. Another important factor to consider is message security, which uses strong protocols to safeguard data integrity and privacy while it is being transmitted.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 4, April 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1204300 |

Additionally, the program incorporates dynamic language translation features that facilitate multilingual conversation in real time and promote inclusivity among users with varying linguistic backgrounds. Understanding text variations—like the differences between capital and lowercase letters—improves message understanding and readability.
The application also places a strong emphasis on user authentication techniques to provide safe access and confirm users' identities. It also allows for smooth file sharing, which makes cooperation easier by allowing users to share documents during chat sessions.

This study assesses the built web chatting application's efficacy and user happiness, highlighting its role in facilitating safe, effective, and participatory online communication. The technological implementation, assessment techniques, and user testing results are covered in detail in the following sections. The ultimate goal of this research is to provide insightful information on how web-based chat apps should be designed and optimized to meet the demands of modern communication.

II. LITERATURE REVIEW

Authors Name	Title	Methodology	Algorithms	Pros	Cons	Research Gaps
Vandika, A. Y., & Tanjung, T.	Study Security Cloud with SHA-2 Algorithm.	The study employs a quantitative research approach to assess the security of cloud systems.	The SHA-2 algorithm is utilized to enhance data security within the cloud environment.	Strong resistance to cryptographic attacks: SHA-2 has demonstrated robustness against attacks like preimage and collision, ensuring the integrity and security of hashed data.	Potential vulnerabilities: Research indicates theoretical vulnerabilities in SHA-2, such as the slim possibility of collisions, which could compromise data integrity.	Lack of comparative analysis: Doesn't compare SHA-2 with other algorithms, missing insights on strengths and weaknesses
Granado-Criado, J. M., Vega-Rodríguez, M. A., Sánchez-Pérez, J. M., & Gómez-Pulido, J. A.	A new methodology to implement the AES algorithm using partial and dynamic reconfiguration.	The research introduces a novel approach for implementing the AES algorithm by incorporating partial and dynamic reconfiguration techniques.	The AES algorithm is employed to ensure robust encryption within the implemented methodology.	Strong security: AES is recognized as a highly secure symmetric encryption algorithm, resistant to various cryptographic attacks such as brute-force, differential, and linear cryptanalysis.	Vulnerabilities to side-channel attacks: While AES itself is considered secure, implementations may be vulnerable to side-channel attacks, such as timing attacks and power analysis, which exploit implementation flaws rather than weaknesses in the algorithm itself.	Could further explore the practical implications of the proposed methodology in real-world wireless network scenarios.
Ali, A., & Sagheer, A.	Design of secure chatting application with end-to-	The study employs a design-based approach to	End-to-end encryption is implemented to ensure	Data Integrity: End-to-end encryption also ensures	User Experience: End-to-end encryption	Lack of comprehensive performance evaluation

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 4, April 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1204300 |

	end encryption for android platform.	develop a secure chatting application for the Android platform.	secure communication within the developed application.	data integrity, as any tampering with the encrypted messages during transit will result in decryption errors, alerting the recipient to potential tampering attempts.	adds complexity to the user experience, as users may need to exchange encryption keys or verify each other's identities to establish secure communication channels. This may deter some users who prioritize convenience over security.	metrics for a holistic assessment.
Bamane, A., Bhoyar, P., Dugar, A., & Antony, L.	Enhanced Chat Application.	The study utilizes an enhancement-focused approach to develop a Chat Application with improved features	Various security algorithms, possibly including cryptographic protocols and authentication mechanisms, are integrated to fortify the security of the chat application.	Cross-Platform Compatibility: Ensuring compatibility across multiple platforms (e.g., Android, iOS, web) allows users to seamlessly access the chat application from various devices and operating systems, increasing accessibility and convenience.	User Adoption and Learning Curve: Introducing new features and functionalities may require users to adapt to changes in the interface and workflow, leading to a learning curve and potential resistance to change, particularly among less tech-savvy users.	Lacks discussion on implementation challenges, like bandwidth requirements.
Suchita Tayde and Seema Siledar.	File Encryption, Decryption Using AES Algorithm in Android Phone.	The research implements file encryption and decryption functionalities on Android devices using the AES algorithm.	The AES (Advanced Encryption Standard) algorithm is employed for secure encryption and decryption of files on Android phones.	Performance: AES encryption and decryption operations are computationally efficient, allowing for fast processing speeds even for large files, making it suitable for use on resource-constrained devices like	Resource Constraints: Resource-constrained devices like Android phones may have limited computational resources (e.g., CPU, memory) and storage capacity, which could affect the feasibility and	Limited Scope: Focuses primarily on AES encryption, leaving out detailed discussion or comparison of other encryption methods.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 4, April 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1204300 |

				Android phones.	efficiency of implementing AES encryption for large files or in memory-constrained scenarios.	
N. Chaudhari, S. Shinkar, and P. Pagare	Chatting Application with Real Time Translation	Data collection: Obtain user preferences and needs for the real-time translation chat application. To learn about customer requirements, preferred languages, and desired features, hold focus groups, interviews, or surveys. Selection of Algorithms: To enable the real-time translation of text communications in many languages, investigate and choose suitable machine translation methods or application programming interfaces. Take into account elements including translation speed, accuracy, language coverage, and expense.	Neural Machine Translation (NMT): Models based on deep learning that translate text between languages while taking the context and semantics of the source and target languages into account.	Global Audience: Chatting apps that provide real-time translation have the ability to reach a worldwide user base, drawing in speakers of other languages and encouraging accessibility and inclusivity.	Privacy Issues: Sending chat messages using third-party translation services gives rise to privacy issues pertaining to data security, confidentiality, and sensitive information access by third parties.	Contextual Understanding : Context-dependent linguistic subtleties and cultural allusions are frequently difficult for real-time translation algorithms to handle. To improve algorithms' comprehension and translation of contextual meaning, more research is required.
A. Ali and A.	Design of	Secure	Hash	Message	speed	Scalability: In

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 4, April 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1204300 |

Sagheer	secure chatting application with end-to-end encryption for android platform	transfer: To prevent eavesdropping and man-in-the-middle attacks, encrypt data transfer between the chat application and the server using secure communication protocols like HTTPS or TLS (Transport Layer Security).	Function: To verify the integrity of sent messages and make sure they haven't been altered, cryptographic hash algorithms like SHA-256 are utilized.	Integrity: To make sure that messages haven't been altered during transmission, cryptographic techniques like hash functions check the integrity of sent messages.	Overhead: Using encryption and decryption can result in computational overhead that affects the responsiveness and speed of applications, especially on mobile devices with limited resources.	order to handle a large number of users and concurrent discussions while maintaining security and performance, scalability difficulties in secure chatting apps require research.
M. Singh, A. Verma, A. Parasher, N. Chauhan, & G. Budhiraja	Implementation of Database Using Python Flask Framework	The architecture of Model-View-Controller (MVC): Put the MVC design into practice to divide the application's concerns. Define views to display HTML templates, controllers to manage user requests and responses, and models to communicate with the database.	Query optimization algorithms: To analyse and optimize SQL queries for effective execution, techniques including cost-based optimization and heuristic optimization are utilized. Concurrency control algorithms: To manage concurrent access to the database and guarantee data consistency and isolation, algorithms such as Two-Phase Locking (2PL), Multiversion Concurrency Control (MVCC), and Optimistic Concurrency Control (OCC)	ORM Support: By offering an object-oriented interface for dealing with the database, integration with ORM libraries such as SQLAlchemy minimizes the need to write raw SQL queries and streamlines database interaction.	Learning Curve: Compared to other web frameworks, Flask may have a higher learning curve because of its minimalistic approach and lack of built-in capabilities, which may force developers to write more boilerplate code and handle low-level details.	Data privacy: Research is required to create methods for anonymizing and safeguarding sensitive data kept in databases, as concerns about data privacy and compliance with rules (such as the GDPR) grow.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 4, April 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1204300 |

			are employed.			
R. Vijayaraghavan	An Architecture for Logging and Searching Chat Messages	Implementing Logging Services: To record incoming and outgoing chat messages instantly, use logging services. For the purpose of receiving, processing, and storing messages from chat clients in the database, define message queues or APIs.	Search Indexing Algorithm: To facilitate quick and effective searches, the search indexing algorithm indexes chat messages. Typical indexing algorithms are those found in full-text search engines such as Apache Lucene or Elasticsearch, as well as inverted indexing for keyword-based searches.	Searchability: By adding search functionality, users can quickly look up and retrieve particular chat messages using a variety of parameters, which improves productivity and usefulness.	Privacy Issues: The storage and retrieval of chat messages give rise to privacy issues, particularly with regard to the security and privacy of user correspondence. Protecting user privacy requires the implementation of suitable encryption and access control systems.	Real-time indexing: To ensure quick and current search results, research is required to create systems and methods for real-time indexing that can effectively index chat messages as they are logged.
I. Karabey and G. Akman	A cryptographic approach for secure client-server chat application using public key infrastructure (PKI)	Digital Signatures: To ensure message integrity and authenticity, use digital signature techniques like RSA or ECDSA. Use the sender's private key to sign chat messages, and then use their public key—which can be found in their digital certificate—to confirm the signature.	Asymmetric Encryption Algorithm: Digital signatures and shared secret keys are encrypted asymmetrically using RSA or Elliptic Curve Cryptography (ECC).	Authentication: PKI-based authentication guards against impersonation and man-in-the-middle attacks by confirming the legitimacy of clients and the server.	Performance Overhead: PKI-based authentication and encryption may result in performance overhead, which can affect the scalability and responsiveness of the system, particularly during key exchange and handshake procedures.	Key management: More investigation is required to create secure and effective key distribution, revocation, rotation, and generation procedures for PKI-based applications, particularly in multi-user settings.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 4, April 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1204300 |

III. METHODOLOGY

A. Introduction

HTTP (Hypertext Transfer Protocol) is the primary protocol used for communication between the client (web browser) and the server while creating a LAN chat web application using Flask.

B. HTTP Standard Protocol

The common protocol for sending data over the internet is HTTP. It outlines the format and method of message transmission between the client and the server. In our Flask application, the client sends HTTP requests to the server to perform actions (like submitting a form) or fetch resources (like HTML pages, CSS stylesheets, JavaScript files, etc.). We are using Web Sockets, a protocol that offers full-duplex communication channels over a single TCP connection, to integrate real-time functionality in our application, such as instant messaging and chat rooms.

C. Flask-Socket IO

A Flask addition called Flask-Socket IO allows interaction with Socket.IO, which facilitates real-time bidirectional communication between the client and the server using Web Sockets. This enables you to integrate functionalities like chat, real-time messaging, and live updates into your Flask application.

D. Fetch API in JavaScript

To store the file data and any additional form fields you wish to send with the file, create a FormData Object.

Create the Request for Fetch: Send the file data to a server-side endpoint that is capable of handling file uploads by using the get() function.

Handle the Server Response: After the file upload has been processed by the server, you may handle the response to verify that the upload was successful or address any issues.

E. Flask

Among web frameworks, Flask stands out for being lightweight and simple, making it ideal for LAN-based web chat programs. Because of its versatility and simplicity, developers are free to concentrate on the essential features rather than deal with superfluous details. Flask's user-friendly syntax makes it easier to get started with web development projects, allowing developers to get started on project implementation as soon as possible. Its integrated routing system simplifies the mapping of URLs, making it easier to include chat features like sending and receiving messages and joining rooms. Moreover, the addition of Jinja2(future prospects), a powerful template engine, makes it easier to create dynamic HTML content—which is necessary for easily displaying user interfaces and chat messages.

Real-time communication between servers and clients is made easier by Flask's interoperability with WebSocket libraries like Flask-Socket IO, which improves functionality like live chat and instant messaging in web chatting apps. Flask enhances the functionality and security of the application by utilizing the vast ecosystem of Python to add several libraries and tools for encryption, authentication, and database integration. Furthermore, Flask's integrated debugging and development tools facilitate the troubleshooting process, resulting in more seamless development iterations. Flask facilitates quick problem-solving and the adoption of best practices in web application development by providing developers with easily accessible tools and collective wisdom. It is backed by a thriving community and extensive documentation.

Therefore, you can use Flask-Socket IO to integrate Web Sockets to provide real-time capabilities in your LAN chat web application, even if HTTP is the main protocol used for general client-server communication in Flask applications.

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 4, April 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1204300 |

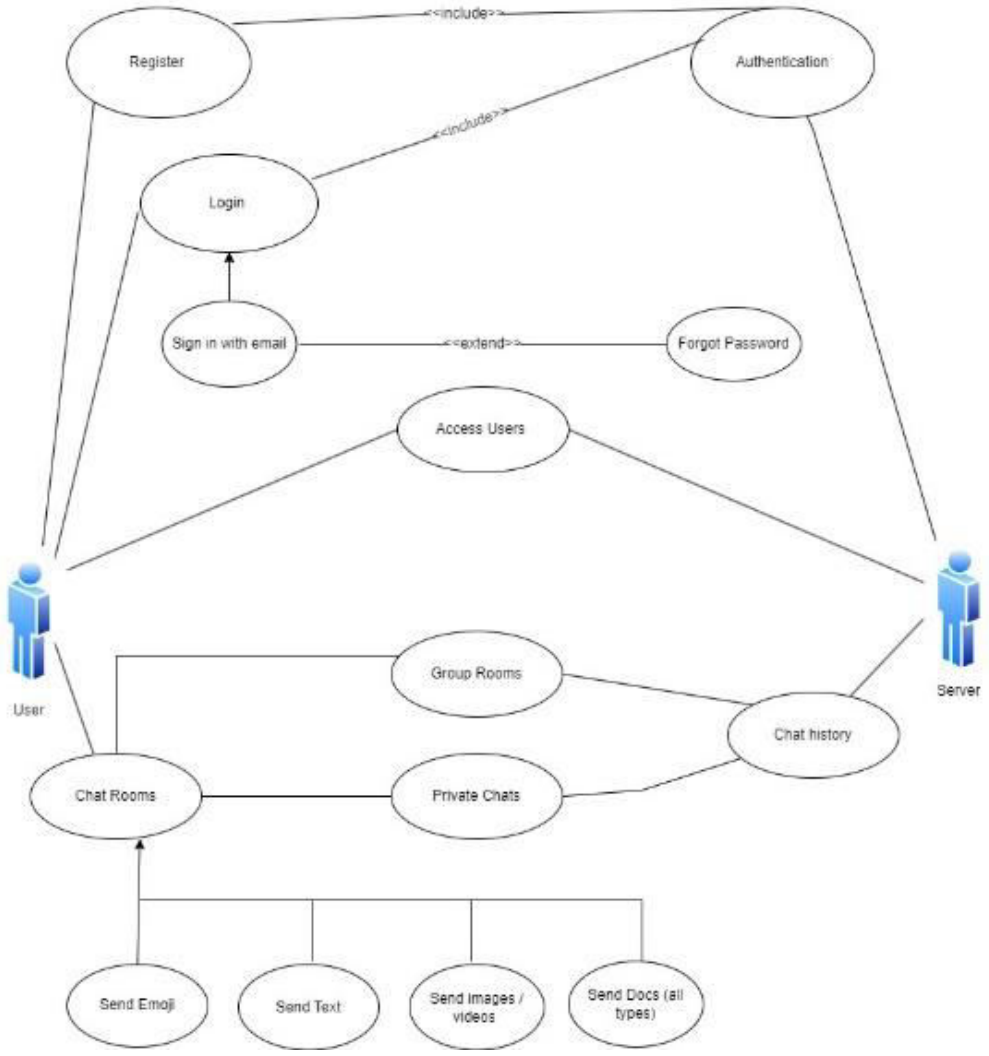

Fig. Use case diagram

IV. RESULTS

It was determined how well the server code performed when 25 clients were making numerous requests at once. The server code performed admirably, handling more than 100 client-initiated actions and handling all requests with no lag. Important Factors Affecting Performance:

Operating System on the Machine: The operating system of the server is a crucial element that determines how well concurrent processes are managed overall, as well as how well tasks are scheduled and resources are allocated.

Hardware configuration: The server's ability to process and reply to client requests quickly depends on its hardware setup, which includes its processing speed, memory size, and storage.

Network Performance, Configuration, and Congestion: The latency and throughput of the server's responses are greatly influenced by the network infrastructure that connects the server and clients. This infrastructure is defined by its performance metrics and levels of congestion.

Together, these elements provide the foundation for the server code's capacity to manage a large number of simultaneous requests, guaranteeing smooth operation and peak performance.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.379 | Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 4, April 2024 ||

| DOI: 10.15680/IJIRCCE.2024.1204300 |

V. DISCUSSION

An additional degree of security is added by the web chatting application's successful integration of end-to-end encryption, which guarantees that only the users who are conversing may read the messages. In the modern digital world, where consumers are more concerned about security and privacy, this functionality is crucial.

Because users may not always speak the same language in globalized environments, the dynamic language translation capability is essential. Language barriers are successfully removed by this function, allowing people with varied linguistic backgrounds to communicate easily.

To guarantee that only authorized users may access the chat material, the user authentication functionality verifies the identity of the users before allowing access to the chat interface. Ensuring the confidentiality of user communications requires this functionality.

The application's adaptability is increased by the seamless file transfer between users made possible by the file sharing feature. In professional situations, where exchanging files and documents during talks is common, this functionality is especially helpful.

This application was developed using Flask, HTML, AJAX, CSS, and JavaScript, and it worked well, offering a stable and dynamic framework. Because of the user-friendly design of the application, users can explore and use its numerous functions with ease.

Finally, the research has resulted in the creation of a safe, effective, and user-friendly web chat program. To improve the user experience even more, future research may investigate adding other capabilities like group chat or video calling.

VI. CONCLUSION

To summarize, the creation and evaluation of web-based chat systems has become increasingly important in satisfying the changing needs of modern communication. This study emphasizes the importance of combining essential features such as end-to-end encryption for message security, dynamic language translation for inclusivity, strong user authentication for safety, and frictionless file sharing for collaboration. The efficacy and user satisfaction of the developed chat application were proved using thorough assessment approaches and user testing, emphasizing its role in promoting secure, efficient, and engaging online interactions among varied user groups. Moving forward, continued study and refining of these technologies will improve web-based chat systems to match the dynamic demands of today's communication environments.

REFERENCES

[1] A. Y. Vandika and T. Tanjung, “Study Security Cloud with SHA-2 Algorithm,” Jurnal Pendidikan Tambusai, vol. 7, no. 2, pp. 18154–18157, 2023

[2] J. M. Granado-Criado, M. A. Vega-Rodríguez, J. M. Sánchez-Pérez, and J. A. Gómez-Pulido, “A new methodology to implement the AES algorithm using partial and dynamic reconfiguration,” Integration, vol. 43, no. 1, pp. 72–80, 2010.

[3] A. Ali and A. Sagheer, “Design of secure chatting application with end to end encryption for android platform,” Iraqi Journal for Computers and Informatics, vol. 43, no. 1, pp. 22-27, 2017.

[4] A. Bamane, P. Bhoyar, A. Dugar, and L. Antony, "Enhanced Chat Application," Global Journal of Computer Science and Technology Network, Web & Security, vol. 12, no. 11, pp. 1-7, 2012.

[5] S. Tayde and S. Siledar, “File encryption, decryption using AES algorithm in android phone,” International Journel of Advanced Research in computer science and software engineering, vol. 5, no. 5, 2015.

[6] N. Chaudhari, S. Shinkar, and P. Pagare, "Chatting Application with Real Time Translation," 2018.

[7] A. Ali and A. Sagheer, “Design of secure chatting application with end to end encryption for android platform,” Iraqi Journal for Computers and Informatics, vol. 43, no. 1, pp. 22-27, 2017.

[8] M. Singh, A. Verma, A. Parasher, N. Chauhan, & G. Budhiraja, “Implementation of Database Using Python Flask Framework,” International Journal of Engineering and Computer Science, vol. 8, no. 12, pp. 24890-24893, 2019.

[9] R. Vijayaraghavan, "An Architecture for Logging and Searching Chat Messages," University of Madras, India, 1999.

I. Karabey and G. Akman, "A cryptographic approach for secure client-server chat application using public key infrastructure (PKI)," in 2016 11th International Conference for Internet Technology and Secured Transactions (ICITST), pp. 442-446, December 2016.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi crossref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details