Veilcomm: Next-Generation Secure Messaging with Custom Encryption and Key Exchange

T Manikumar, V Kesavan, M Francies Antony, A Venkat Raman and Ramsubramanyam V G
Department of Computer Science and Engineering, Kalasalingam Academy of Research and Education, Krishnankoil, Srivilliputhur, Tamil Nadu 626126, India

Keywords: Secure Messaging, End-to-End Encryption, Cryptographic Key Exchange, Data Security, Privacy, Confidential Communication, Encryption Algorithms.

Abstract: This paper describes a secure messaging web application intended for confidential conversation between two users. The program prevents unwanted users from joining the communication session and emphasizes data security and privacy by not retaining session data. User credentials, including login and password information, are safely kept in a server-connected database. A fundamental aspect of the program is the use of algorithms such as X25519, X3DH, KYBER, and NTRU to securely exchange cryptographic keys between users. These keys are then used to encrypt and decode messages during the communication session. The program encourages the creation of dependable, secure, and creative digital infrastructure by including a bespoke encryption module that supports numerous encryption algorithms, including RSA-OAEP, AES-GCM, DOUBLE RATCHET, CHACHA20-POLY1305, KYBER, and NTRU. This strategy improves communication system resilience, hence strengthening the digital security environment. When you start a session, a random encryption method is chosen, offering an extra degree of protection. Messages are encrypted in the backend before transmission to ensure that every communication is private and safe. The application promotes confidence in digital platforms by exploiting modern cryptographic methods and technical breakthroughs, hence contributing to the construction of a more secure and sustainable digital environment. The application's design stresses simplicity of use with a user-friendly interface, while still adhering to secure computing best practices to protect user data and privacy. This paper exemplifies a holistic approach to secure communication, helping to build dependable and secure communication technology for an increasingly connected world.

1 INTRODUCTION

In an era where digital communication is ubiquitous, ensuring the security and privacy of online conversations has become a critical challenge. Cyber threats, data breaches, and unauthorized access to sensitive information pose significant risks to individuals and organizations alike. To address these concerns, this paper introduces a secure messaging web application that enables private communication between two users while maintaining the highest standards of security. This application is designed with a privacy-first approach, ensuring that all communications remain confidential and are accessible only to the intended recipients. Unlike conventional messaging platforms that may store session data or expose user information to third parties, this application prioritizes end-to-end encryption and does not store session data on the server. This prevents potential security vulnerabilities, such as data leaks or unauthorized access (Gour, A., Malhi, S. S., Singh, G., & Kaur, G. (2024)) (Cui, Li, Qianqian, Xing, Yi, Wang, Baosheng, Wang, Jing, Tao, Liu, Liu (2022)) (Len, J., Ghosh, E., Grubbs, P., & Rösler (2023)) (N. Unger (2015)).

1.1 Key Features and Security Mechanisms

To establish a secure communication channel, the application employs advanced cryptographic techniques, including key exchange algorithms like:

Manikumar, T., Kesavan, V., Antony, M. F., Raman, A. V. and G., R. V.
Veilcomm: Next-Generation Secure Messaging with Custom Encryption and Key Exchange.
DOI: 10.5220/0013914500004919

In *Proceedings of the 1st International Conference on Research and Development in Information, Communication, and Computing Technologies (ICRDICCT'25 2025) - Volume 4*, pages 447-460
ISBN: 978-989-758-777-1

X25519, X3DH (Extended Triple Diffie-Hellman), KYBER (post-quantum key exchange), NTRU (post-quantum lattice-based encryption)

These algorithms facilitate the secure exchange of cryptographic keys between users, ensuring that only the communicating parties can decrypt the exchanged messages. Once a secure session is established, messages are encrypted and decrypted using a custom encryption module that supports multiple encryption algorithms, including:

RSA-OAEP (Asymmetric encryption for secure key exchange), AES-GCM (Authenticated symmetric encryption), Double Ratchet Algorithm (Used in secure messaging protocols like Signal), ChaCha20-Poly1305 (High-speed and secure encryption), KYBER and NTRU (Post-quantum encryption techniques)

To further enhance security, the application implements a randomized encryption algorithm selection mechanism. Whenever a communication session is initiated, the system randomly selects one of the available encryption algorithms, making it significantly more difficult for attackers to predict or compromise the encryption process. (Phan, Duong & Pointcheval, David. (2004)) (Alwen, J., Coretti, S., Dodis, Y. (2019).) (Leon Botros, Matthias J. Kannwischer, and Peter Schwabe. (2019).) (Kim, Kyungho, Seungju Choi, Hyeokdong Kwon, Hyunjun Kim, Zhe Liu, and Hwajeong Seo. (2020)) (Hoffstein, J., Pipher, J., Silverman, J.H. (2001)) (Serrano, R., Duran, C., Sarmiento, M., Pham, C.-K., & Hoang, T.-T. (2022)) (A. Ruggeri, A. Galletta, A. Celesti, M. Fazio and M. Villari, (2021)) (J. Dong, F. Zheng, J. Cheng, J. Lin, W. Pan and Z. Wang, (2018)).

1.2 Backend Security and Data Privacy

All encryption and decryption processes take place in the backend before messages are transmitted, ensuring that no plaintext messages are exposed during transmission. Additionally, user credentials (login and password) are securely stored in a server-connected database, following industry best practices such as salting and hashing passwords to protect against unauthorized access. By eliminating session data storage and focusing on strong encryption methods, the application minimizes the risk of data interception, replay attacks, or unauthorized access. This approach aligns with modern security standards and ensures that messages remain confidential at all times. (Dabola, S., Tomer, V., Singh, N., Madan, P., & Jhinkwan, A. (2024))(Gour, A., Malhi, S. S., Singh, G., & Kaur, G. (2024))(Shamsi, S. E., & Nasrat, B. W. (2024))(Shuhan, M. K. B., Islam, T., Shuvo, E. A., Bappy, F. H., Hasan, K., & Caicedo, C. (2023)).

1.3 User Experience and Accessibility

Despite its robust security framework, the application is designed to be user-friendly and accessible. The interface provides an intuitive messaging experience without compromising on security. Users can communicate seamlessly without needing technical expertise, making the platform suitable for a wide range of users, from individuals seeking private conversations to organizations requiring secure internal communication. (Pandey, D., Gupta, R. R., & Choudhary, A. (2023)) (Mashari Alatawi and Nitesh Saxena. (2023)) (N. Unger (2015)) (Puyosa, I. (2023)).

2 LITERATURE REVIEW

This is a hybrid approach by combining symmetric and asymmetric cryptographic techniques along with digital signatures, resulting in an enhanced hybrid cryptography method combining symmetric and asymmetric techniques may increase the computational overhead. By using a methodical approach to literature review, we analyse studies carried out on a range of scientific platforms that connect the fields of cryptography and chat, as well as the intersection of databases and cryptographs. There is a huge chance that sophisticated encryption methods may improve user data security. There are benefits to using cryptography in a chat interface since several academics have developed and verified models that maximize the usage of encryption for data exchange between users. (Dabola, S., Tomer, V., Singh, N., Madan, P., & Jhinkwan, A. (2024)).

The methodology integrates a public blockchain, end-to-end encryption, and decentralized storage to ensure secure, immutable messaging without centralized control, this model is the reliance on a public blockchain, which can lead to scalability issues and increased transaction costs as network usage grows. This study suggests a brand-new messaging software that makes use of blockchain technology to provide improved decentralization, security, and privacy. The limits of conventional messaging apps and the advantages of blockchain-based solutions are covered in the first section of the article. The suggested application offers end-to-end encryption for enhanced privacy and makes use of a public blockchain network to guarantee the integrity and immutability of communications. (Pandey, D., Gupta, R. R., & Choudhary, A. (2023)).

Data is the lifeblood of both people and organizations in our ever-changing digital world, therefore protecting sensitive data and making sure that communication routes are secure have become critical. This study presents a brand-new hybrid cryptographic algorithm created to tackle the complex problems of secure communication and data protection. By using the advantages of both symmetric and asymmetric encryption techniques, the suggested method paves the way for reliable and flexible security solutions. This hybrid technique starts by effectively encrypting data while maintaining its anonymity using the Advanced Encryption Standard (AES), a cutting-edge symmetric encryption cipher. (Gour, A., Malhi, S. S., Singh, G., & Kaur, G. (2024)).

Cryptography and steganography are two well-liked methods for transmitting sensitive data covertly. One conceals the message's existence, while the other misrepresents it. As is well known, the most crucial aspects of information technology and communication, after the development of the internet, are data security and privacy. Large amounts of data are sent every day over the internet, file sharing websites, social networking sites, and other channels. At the same time, there is a rise in the number of people using the internet. The goal of this article is to create an easy-to-use application that would, at the very least, partially alleviate security issues with message delivery to authorized parties. In order to make the application run quickly, we used the C# programming language. After the user chooses the file and picture, the encryption process will be finished in a matter of seconds after they push the button. According to this assessment, users find this program to be user-friendly and likely to be helpful when delivering messages using cryptography and steganography techniques. (Shamsi, S. E., & Nasrat, B. W. (2024))

Data security is still one of the most important issues in today's digital world. The complexity of many encryption methods and their vulnerability to cyberattacks have caused problems. This paper presents Cypher-X, a sophisticated encryption and decryption system that skilfully combines traditional methods, namely Vigenère and RSA, to strengthen data secrecy. This study aims to solve current issues by concentrating on three main goals: enhancing data security protocols, assessing system effectiveness, and detecting weaknesses that cybercriminals take advantage of. Our study's findings confirm Cypher-X's strong encryption capabilities and highlight how it might be improved even further by adding multi-factor authentication and access control strategies. Cypher-X is a promising method to protect sensitive data in the digital arena, making it a beacon of hope in the data security space. As we traverse the intricate terrain of data security, it is imperative to recognize the continuous need to clarify our study question, methods, significant discoveries, and practical ramifications. (AlSideiri, A., AlShamsi, S., AlBreiki, H., AlMoqbali, M., AlMaamari, M., & AlSaadi, S. (2024)).

Over the last 20 years, individuals and organisations have increasingly used messaging services. Many of these systems employed end-to-end encryption to provide "future secrecy" for one-to-one communication. Most of them utilise client data on centralised servers owned by huge IT businesses. It also allows the government to track and control its citizens, endangering "digital freedom." These systems lack confidentiality, integrity, privacy, and future secrecy for group communications. We present Quarks, a blockchain-based secure messaging system that eliminates centralised control and fixes existing security issues. Our system's reliability and utility were tested using security models and definitions from the literature. DLT was used to develop a Quarks system Proof of Concept (PoC) and test its load. Even if the development and testing environment is limited, our proof-of-concept system has all the required characteristics for a centralised messaging method. This assures that such systems will be applicable soon if scaled up. (Shuhan, M. K. B., Islam, T., Shuvo, E. A., Bappy, F. H., Hasan, K., & Caicedo, C. (2023)).

In the rapidly changing field of Internet technologies, where blockchain-based computing and storage like Ethereum Virtual Machine (EVM), Arweave, and IPFS are becoming more popular, a comprehensive decentralised communication framework is still lacking. This mismatch emphasises the necessity for a system that allows direct messaging to wallet addresses and seamless cross-platform messaging to encourage interoperability and privacy across platforms. To answer this requirement, SendingNetwork delivers a reliable and secure decentralised communication network by addressing privacy, scalability, efficiency, and composability. We use edge computing and the modular libp2p framework to develop an adaptive relay network. Our dynamic group chat encryption solution uses the Double Ratchet algorithm for secure communication to increase resilience and scalability. For optimal message processing in big group conversations, we recommend delegation. The Delegation method performs better, according to our theoretical studies.

To encourage node involvement and system stability, we use "Proof of Availability" to control incentives using Verkle trees and assure network consistency, and "Proof of Relay" to confirm message relay burden using the innovative KZG commitment. Our whitepaper describes the network's design, core components, roadmap, and forthcoming SendingNetwork developments. (Yeung, M. (2024)).

Implementing cryptographic systems on devices with poor protection raises worries about long-term secret leaking, especially for IoT devices with minimal resources. Forward concealment may reduce the damage from such an occurrence. Some pub-/sub-based IoT systems utilise end-to-end (from publisher to subscriber) encrypted message transmission methods to overcome secrecy concerns created by malicious message brokers. No one guarantees upfront secrecy. This article introduces FSEE, a pub-/sub-based IoT forward-safe end-to-end encrypted message transmission system. For FSEE, we develop BA-GKE (group key exchange protocol). It employs a semi-trusted key exchange server for forward secrecy and asynchronous group communication. We prove forward secrecy using ProVerif. FSEE uses BA-GKE asynchronously to establish forward safe symmetric keys for each device. For safe communication encryption, the device and permitted subscribers get this device-specific key. FSEE may be added progressively without changing the message broker by adding a semi-trusted key exchange server to BA-GKE in the existing IoT architecture. FSEE is safer and performs comparable to other well-known research, according to experiments. (Cui, Li, Qianqian, Xing, Yi, Wang, Baosheng, Wang, Jing, Tao, Liu, (2022)).

Digital Markets Act (DMA) is a new EU policy passed in May 2022. The demand that “gatekeepers” like WhatsApp integrate “interoperability” with other messaging appsen crypted communications between service providersis one of its most contentious features. This need fundamentally changes the design assumptions of most encrypted communications systems, which are centralised. Technologists have not begun to consider the many security, privacy, and functionality issues raised by the interoperability requirement. Since the DMA's mandate may take effect in mid-2024, researchers must develop solutions. We start this path in this paper. The DMA affects encrypted messaging system design in three main areas: identity, or how to resolve identities across service providers; protocols, or how to secure connections between clients on different platforms; and abuse prevention, or how service providers can detect and act against abusive or spamming users. We define major security and privacy criteria, summarise current approaches, and evaluate whether they fit our standards for each area. Finally, we present our interoperable encrypted messaging system architecture and identify issues. (Len, J., Ghosh, E., Grubbs, P., & Rösler, P. (2023))

Recent revelations of government surveillance on personal communication have led several alternatives to guarantee secure and private messaging. This includes numerous new papers and a wide variety of commonly used utilities with security features. The drive to deliver solutions quickly has led to inconsistent threat models, unfulfilled objectives, dubious security claims, and a lack of knowledge of secure communication cryptographic literature during the last two years. Existing secure messaging solutions' security, usability, and ease-of-adoption are assessed in this research. We consider academic answers and innovative "in-the-wild" ways that aren't in the scholarly literature. The design environment for transit privacy, conversation security, and trust creation is mapped. Trust formation solutions with strong security and privacy features perform poorly in usability and adoption, whereas hybrid strategies that have gotten less academic attention may provide better trade-offs in real-world circumstances. Conversation security can be done without user involvement in most two-party discussions if trust is established, but there is no viable solution for larger groups. Finally, transport privacy seems to be the hardest to fix without a performance hit. (N. Unger (2015)).

Today's primary information source, the internet, is becoming harder to monitor and control. These days, network security is becoming a highly important consideration when sharing vast amounts of complicated data. Today, maintaining secure data transmission across a network and protecting data to prevent unauthorised users from accessing it while allowing authorised users to share it are the two key concerns. Network security is largely dependent on the use of different encryption techniques. By rendering information unintelligible, cryptography contributes to improved data confidentiality and privacy. In order to list the security level, calculation time, and data transmission time of key exchange protocols, a comparative analysis is conducted in this study. This study compares the performance of many symmetric and asymmetric key encryption algorithms, including Diffie-Hellman Key Exchange, RSA, and ECC. (Chopra, Aakanksha. (2015))

Skype, Zoom, and WhatsApp make text, video, and voice chats ubiquitous. Concerns about how the government and law enforcement monitor these platforms linger as more individuals use them to disclose private information. Due to these issues, private chats and electronic correspondence must be safe. End-to-end encryption (E2EE) might address this issue without depending on first or third parties like an internet service or a public key infrastructure (PKI), which government surveillance programs and law enforcement could target or compel. Start systematising knowledge research using the most popular E2EE applications and communications protocols. We categorise their E2EE capabilities and authentication methods based on current research. Even though previous research has evaluated messaging services, we analyse a larger selection of popular E2EE applications and their authentication methods. We observed that all E2EE applications, especially opportunistic ones, cannot protect against MitM attacks. No E2EE program offers more effective and user-friendly authentication mechanisms, making E2EE connections vulnerable to active MitM assaults. Systematisation may impact E2EE system deployment and MitM and eavesdropping authentication ritual study. (Mashari Alatawi and Nitesh Saxena. (2023))

Signal, a novel security protocol and software, encrypts instant messages end-to-end. The core protocol has been adopted by WhatsApp, Facebook Messenger, and Google Allo, which have at least one billion users. Ratcheting, which changes session keys with each communication, gives Signal uncommon security characteristics like "future secrecy" and "post-compromise security". The Signal protocol is important and unusual, yet it has gotten little academic attention. We provide Signal's double ratchet and key agreement's first security research as a multi-stage key exchange system. We build a security model to capture the "ratcheting" key update structure and extract a formal abstract protocol description from the implementation. Next, we use our model to demonstrate Signal's core's security by demonstrating many security traits. Our presentation and conclusions may be used to evaluate this widely used approach, and we found no serious design flaws. (K. Cohn-Gordon, C. Cremers, B. Dowling, L. Garratt and D. Stebila, (2017)).

Protecting point-to-point messaging apps: Understanding Telegram, WeChat, and WhatsApp in the United States is a year-long study of messaging app technology, policy, and usage. Atlantic Council's Digital Forensic Research Lab launched it. Over 3 billion people worldwide use messaging apps to communicate with friends and family, shop, read, and discuss current events. Senior Research Fellow Iria Puyosa will examine platform policies, security, and messaging app usage patterns and implications. After the introduction, experts will address the trade-offs of messaging firms' attempts to identify, reduce, or stop dangerous content in encrypted messaging applications. The panellists will cover client-side scanning, in-app reporting, metadata analysis, and behavioural signals as solutions. The discussion will illuminate how methods may affect global data privacy, user security, and human rights. (K. Cohn-Gordon, C. Cremers, B. Dowling, L. Garratt and D. Stebila, (2017))

network application that uses an unsecured communication channel has always had to secure network traffic. The goal is to prevent unauthorised disclosure and message tampering between communication parties while protecting the data being transferred over the network. The cryptographic primitive that can create a secure communication channel is a key exchange protocol. Diffie-Hellman established the first key exchange protocol. Enabling two parties to safely exchange a session key that may be used for further symmetric encryption of communications is the goal of the Diffie-Hellman protocol. Nevertheless, the communicating entities are not authenticated by Diffie-Hellman itself. We examine the Diffie-Hellman Key exchange protocol in this work. Then go on to explain the Diffie-Hellman protocol's variations, authenticated key exchange protocol and one-pass key exchange protocol. (Mishra, M. R., & Kar, J. (2017)).

Ten years old, the OAEP structure is employed in many real-world applications. Four years ago, the first effective and provably IND-CCA1 safe encryption padding was shown to meet the claimed IND-CCA2 security level with any trapdoor permutation, ending considerable debate. This intractability assumption is equal to full-domain one-wayness for the primary application (using the RSA permutation family), but it needs a quadratic-time reduction. Thus, it is worse than RSA's flimsy security proof. OAEP offers two practical advantages: efficiency from the two additional hashings and length from the minimal bit-length of the ciphertext. Randomisation (for semantic security) and redundancy (for plaintext-awareness, the only method to design strong CCA2 schemes) reduce bandwidth (ciphertext-to-plaintext ratio). The OAEP 3-round design in the randomoracle model was used to demonstrate the first IND-CCA2 safe encryption algorithms without redundancy or plaintext-awareness at Asiacrypt '03. Its practical properties were only similar to the original OAEP construction: security requires partial-domain one-wayness and a trapdoor permutation, limiting its use to RSA and yielding a poor reduction. To enhance the reduction, we demonstrate that the OAEP 3-round uses the permutation's full-domain one-wayness. The software is extended using ElGamal, Paillier, and other encryption primitives. Both relaxed CCA2 and random-oracle models maintain higher security. (Phan, Duong & Pointcheval, David. (2004)).

The new public key cryptosystem known as NTRU is discussed in this piece of literature, which also includes a description of the system. NTRU is distinguished by a number of characteristics, such as its minimal memory needs, its rapid speed, its keys that are relatively short, and its keys that are easy to construct. Additionally, the encryption and decryption techniques that are used by NTR are shown here. A mixing method that is derived from polynomial algebra is used by U in addition to a clustering technique that is founded on the basic theory of probability. Furthermore, the clustering method is used in conjunction with this process. It is the polynomial mixing system and the independence of reduction modulo two relatively prime numbers, p and q, that ensure the safety of the NTRU cryptosystem. Both of these factors contribute to the system's security. Furthermore, the total security of the system is enhanced by both of these other considerations. This degree of security is achieved by the interplay of these two components, which allow for communication between them. (Hoffstein, J., Pipher, J., Silverman, J.H. (2001))

Google Allo, Facebook Messenger, WhatsApp, Skype, and Signal have billions of users. Each connection is encrypted and authenticated with a new symmetric key in "double ratcheting," its essential principle. Forward, post-compromise, and "immediate (no-delay) decryption," unparalleled.... Despite formal review, we feel the Signal methodology and ramping evaluations are inappropriate. To solve this challenge, we define secure communications and show forward, post-compromise, and quick decryption. In contrast to other "provable alternatives to Signal," we are the first to formalise and explain instantaneous decryption, which lets parties recover from lost talks. We build a modular "generalised Signal protocol" using these parts: Continuous key agreement (CKA) is a clean basic that simulates "public-key ratchets" from public-key encryption (not only Diffie-Hellman, like Signal). Two-input PRF-PRNG hash function, FS-AEAD pseudorandom generators, "symmetric-key ratchets;" This helps us quickly create post-quantum security, remove random oracles in analysis, and instantiate our system as the popular Diffie-Hellman-based Signal protocol. We demonstrate that our architecture can include CRYPTO'18's "fine-grained state compromise" types without delaying decryption. The security cost of symmetric-key encryption to public-key cryptography may outweigh the efficiency. (Alwen, J., Coretti, S., Dodis, Y. (2019)).

In computer networks, Transport Layer Security (TLS) offers a secure route for end-to-end communications. In order to mitigate side channel attacks in cypher suites based on the Advanced Encryption Standard (AES), TLS 1.3 introduces the ChaCha20–Poly1305 cypher suite. In contrast to other encryption standards that use Authenticated Encryption with Associated Data (AEAD), the few implementations are unable to provide fast enough encryption. ChaCha20 and Poly1305 primitives are shown in this study. To lessen issues with fragmented blocks, a fault detector is also included with a compatible ChaCha20–Poly1305 AEAD with TLS 1.3. In a standalone core, the AEAD implementation achieves 1.4 cycles per byte. Furthermore, the system implementation uses a TileLink bus and operates at 11.56 cycles per byte in a RISC-V environment. 10,808 Look-Up Tables (LUT) and 3731 Flip-Flops (FFs) are shown by the implementation in the Xilinx Virtex-7 XC7VX485T Field-Programmable Gate-Array (FPGA), which is represented in 23% and 48% of ChaCha20 and Poly1305, respectively. Lastly, the ChaCha20–Poly1305 AEAD hardware implementation shows that it is feasible to use an alternative to the traditional AES-based cypher suite for TLS 1.3. (Serrano, R., Duran, C., Sarmiento, M., Pham, C.-K., & Hoang, T.-T. (2022)).

This study presents an optimised Galois Counter Mode of operation (GCM) implementation for low-end microcontrollers using the Advanced Encryption Standard (AES). The suggested implementations are subjected to two optimisation techniques. First, the AES counter (CTR) mode of operation guarantees consistent timing and is speed-optimized. The primary concept is substituting basic look-up table access with costly AES operations such as AddRound Key, SubBytes, ShiftRows, and MixColumns. The look-up table does not need to be updated during the encryption life-cycle, in contrast to earlier studies. Second, the Karatsuba algorithm, compact register utilisation, and pre-computed operands are used to further optimise the Galois Counter Mode (GCM) core operation. The suggested AES-GCM on 8-bit AVR (Alf and Vegard's RISC processor) architecture achieved 415, 466, and 477 clock cycles per byte for short-, middle-, and long-term security levels, respectively, using the aforementioned optimisation strategies. (Kim, Kyungho, Seungju Choi, Hyeokdong Kwon, Hyunjun Kim, Zhe Liu, and Hwajeong Seo. (2020)).

This research presents an improved software implementation of module-lattice-based key-encapsulation technology Kyber for the ARM Cortex-M4 microprocessor. Kyber competes in the NIST post-quantum paper's second round. We examine novel Kyber's number-theoretic transform (NTT) optimisation approaches that use the target architecture's "vector" DSP instructions' computational power. We also present findings for Kyber's improved parameter sets, which benefit from our optimisations. Our program is 18% faster than a Kyber implementation designed for the Cortex-M4 by Kyber submitters. Our NTT is twice as fast as their software's. Our software runs as fast as the latest speed-optimized Sabre, another module-lattice-based round-2 NIST PQC competitor. Our Kyber application utilises far less RAM to operate this well. Kyber uses half as much RAM as Sabre, which is slower. Our time-attack defence is cutting-edge since it doesn't use secret-dependent branching or memory access. (Leon Botros, Matthias J. Kannwischer, and Peter Schwabe. (2019)).

Exploring new horizons to guarantee safe communication between people and devices is being prompted by the widespread usage of the Internet and the rise in IoT devices over the last ten years. A Blockchain-based solution has not yet been investigated, and current options have significant security and distributed attack resistance issues. Despite being in use for many years, the Extended Triple Diffie-Hellman (X3DH) protocol is usually predicated on a single trust third-party server that serves as a single point of failure (SPoF), making it vulnerable to well-known Distributed Denial of Service (DDOS) assaults. To mitigate this danger, the BlockChain-Based X3DH (BCB-X3DH) protocol has already been suggested. It combines the well-known X3DH security methods with the inherent characteristics of Smart Contracts, such as data immutability and non-repudiation. In order to ensure full on-chain secure communication management and even optimise battery life-cycle, we have advanced our research in this paper to implement this protocol in Edge and IoT scenarios, including low-power embedded devices with limited hardware capabilities. (A. Ruggeri, A. Galletta, A. Celesti, M. Fazio and M. Villari, (2021)).

The key exchange, which is widely used in many Internet security protocols like TLS/SSL, offers a way for two parties to create a shared secret across an unprotected channel. Elliptic Curve Diffie-Hellman is now the industry's favourite and most widely used key exchange method. Although the current ECDH uses NIST P Curves as its underlying elliptic curve, the IRTF formally implemented Curve25519/448 to key exchange in RFC 7748, also known as the X25519/448 key exchange protocol, in January 2016 because to concerns about its security and the need for high speed. X25519/448 is now the default key exchange protocol suggested by several popular open-source paper. Scalar multiplication is X25519/448's bottleneck since its performance on a typical CPU cannot meet the constantly growing demand in scenarios with many concurrent requests, such cloud computing, e-commerce, etc. In this work, we use a variety of optimisations to speed up X25519/448 with GPUs. The GeForce GTX 1080's X25519/448 achieves 2.86 and 0.358 million operations per second, respectively, far surpassing the previous fastest work. (J. Dong, F. Zheng, J. Cheng, J. Lin, W. Pan and Z. Wang, (2018)).

3 PROPOSED METHODOLOGY

This application is a cutting-edge solution for securely transferring sensitive messages between users, revolutionizing the way private communication is handled. Built with the highest standards of security in mind, it utilizes a Dynamic Encryption method that combines the strengths of both symmetric and asymmetric encryption techniques to provide an unparalleled level of data protection. Unlike traditional messaging systems that rely on fixed encryption methods, this application dynamically selects a random encryption algorithm for each session. By doing so, it ensures that every communication is uniquely secured, making it significantly harder for attackers to intercept or decipher sensitive information. (AlSideiri, A., AlShamsi, S., AlBreiki, H., AlMoqbali, M., AlMaamari, M., & AlSaadi, S. (2024))(Yeung, M. (2024))(Gour, A., Malhi, S. S., Singh, G., & Kaur, G. (2024))

3.1 Dynamic Algorithm Selection

To achieve the highest level of security, the application incorporates some of the latest and most advanced encryption algorithms. The system intelligently selects from cutting-edge encryption methods such as:

X25519 – A Diffie-Hellman key exchange algorithm using elliptic curves, providing high security with efficient performance. It is widely used in TLS and secure messaging protocols. (Dabola, S., Tomer, V., Singh, N., Madan, P., & Jhinkwan, A. (2024))(Shamsi, S. E., & Nasrat, B. W. (2024)) (Chopra, Aakanksha. (2015)) (J. Dong, F. Zheng, J. Cheng, J. Lin, W. Pan and Z. Wang, (2018))

X3DH (Extended Triple Diffie-Hellman) – A key agreement protocol used in Signal for secure initial key exchange, enabling forward secrecy and post-compromise security. (Dabola, S., Tomer, V., Singh, N., Madan, P., & Jhinkwan, A. (2024))(Shamsi, S. E., & Nasrat, B. W. (2024)) (Chopra, Aakanksha. (2015)) (A. Ruggeri, A. Galletta, A. Celesti, M. Fazio and M. Villari, (2021)).

KYBER – A post-quantum key encapsulation mechanism (KEM) designed to resist attacks from quantum computers, part of NIST's PQC standardization. (Mashari Alatawi and Nitesh Saxena. (2023)) (K. Cohn-Gordon, C. Cremers, B. Dowling, L. Garratt and D. Stebila, (2017))(Shamsi, S. E., & Nasrat, B. W. (2024)) (Leon Botros, Matthias J. Kannwischer, and Peter Schwabe. (2019)).

NTRU – A lattice-based cryptosystem for public-key encryption, providing quantum resistance while being computationally efficient. (Shamsi, S. E., & Nasrat, B. W. (2024))(Yeung, M. (2024)) (Hoffstein, J., Pipher, J., Silverman, J.H. (2001)).

RSA-OAEP (Optimal Asymmetric Encryption Padding) – A secure padding scheme for RSA encryption that prevents deterministic attacks and enhances security. (Phan, Duong & Pointcheval, David. (2004)) (N. Unger (2015))(Dabola, S., Tomer, V., Singh, N., Madan, P., & Jhinkwan, A. (2024)).

AES-GCM (Galois/Counter Mode) – A block cipher mode that provides both encryption and authentication, widely used in secure communications. (Kim, Kyungho, Seungju Choi, Hyeokdong Kwon, Hyunjun Kim, Zhe Liu, and Hwajeong Seo. (2020)) (Mishra, M. R., & Kar, J. (2017)) (Yeung, M. (2024)) (Pandey, D., Gupta, R. R., & Choudhary, A. (2023)).

Double Ratchet Algorithm – A key update mechanism used in Signal Protocol to provide forward secrecy and post-compromise security for encrypted messaging. (K. Cohn-Gordon, C. Cremers, B. Dowling, L. Garratt and D. Stebila, (2017)) (K. Cohn-Gordon, C. Cremers, B. Dowling, L. Garratt and D. Stebila, (2017)) (Mashari Alatawi and Nitesh Saxena. (2023)) (Alwen, J., Coretti, S., Dodis, Y. (2019)).

ChaCha20-Poly1305 – A high-speed authenticated encryption algorithm combining the ChaCha20 stream cipher with the Poly1305 authenticator, optimized for performance and security. (Serrano, R., Duran, C., Sarmiento, M., Pham, C.-K., & Hoang, T.-T. (2022))(Dabola, S., Tomer, V., Singh, N., Madan, P., & Jhinkwan, A. (2024))(Pandey, D., Gupta, R. R., & Choudhary, A. (2023))(AlSideiri, A., AlShamsi, S., AlBreiki, H., AlMoqbali, M., AlMaamari, M., & AlSaadi, S. (2024))(Shuhan, M. K. B., Islam, T., Shuvo, E. A., Bappy, F. H., Hasan, K., & Caicedo, C. (2023)).

Each session leverages a combination of these algorithms to secure the data transmission. For example, RSA or ECC may be used to securely exchange keys, while AES-256 or ChaCha20 handles the actual message encryption. This hybrid approach balances performance, scalability, and security, ensuring that no single vulnerability can compromise the system.

3.2 No-Session-Data Policy

In addition to robust encryption, the application is designed with privacy as a core principle. It follows a strict no-session-data policy, meaning no sensitive information, session keys, or communication metadata are stored on servers. This ensures that even if the server is compromised, no trace of user conversations or encryption keys can be accessed. This approach eliminates the risks associated with data breaches or unauthorized access to archived information, making the system inherently secure and privacy-centric. (K. Cohn-Gordon, C. Cremers, B. Dowling, L. Garratt and D. Stebila, (2017)) (Mashari Alatawi and Nitesh Saxena. (2023)).

3.3 Ideal Use Cases

This application is ideal for securely transferring highly sensitive information in scenarios where privacy is of utmost importance. When sending sensitive information over different networks, it is crucial to use secure messaging with end-to-end encryption. Credit card information, banking credentials, and secret financial agreements may all be safely sent using this method. It allows customers and legal representatives to have private talks without worrying about interception in the legal sector. Protecting intellectual property, trade secrets, confidential data, and company information during transmission is a top priority for businesses. Secure messaging allows for the transfer of private patient information while guaranteeing adherence to strict privacy standards such as HIPAA. Secure messaging is also used by government communications to prevent unauthorised access to sensitive papers and information, which helps to maintain national security. (Cui, Li, Qianqian, Xing, Yi, Wang, Baosheng, Wang, Jing, Tao, Liu, (2022))(Len, J., Ghosh, E., Grubbs, P., & Rösler, P. (2023)).

3.4 Workflow Diagram

Figure 1 show the Secure Session-Based Communication Flow.

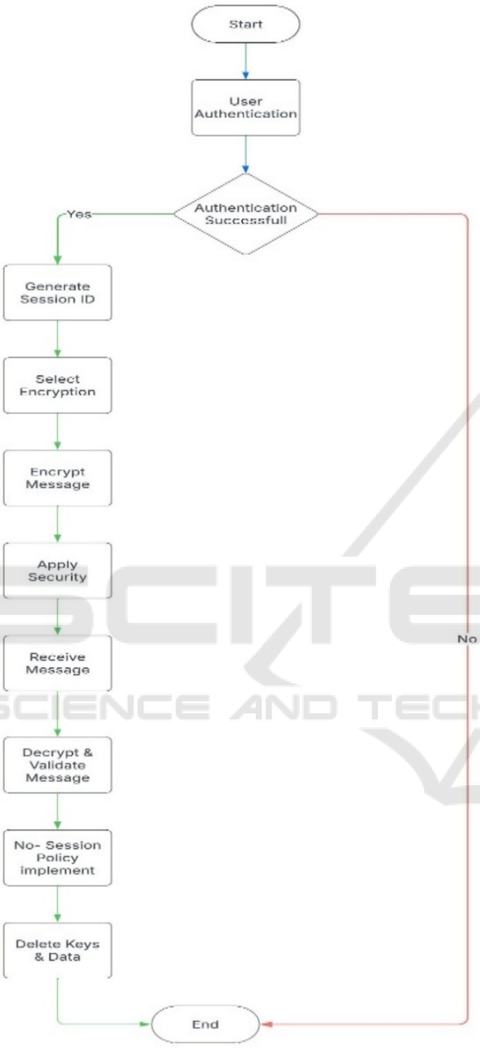
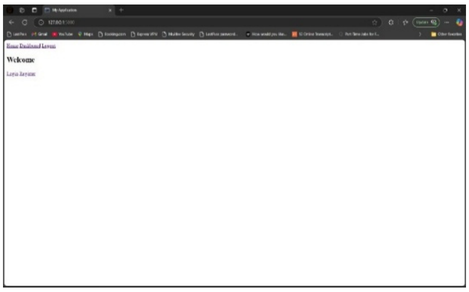


Figure 1: Secure Session-Based Communication Flow.

3.5 Enhanced Security Features

The application goes beyond encryption by incorporating additional security measures to protect against modern cyber threats. Secure messaging solutions are more protected and private thanks to modern security technologies. Even if a session key is compromised, all conversations, both past and future, will be safeguarded because to forward secrecy. By automatically erasing after a certain amount of time, self-destructing communications further protect privacy by erasing any evidence of the conversation. Because no third party can decipher an end-to-end encrypted communication, only the receiver is able to read it. Further enhancing security and reducing vulnerability exposure is real-time key rotation, which dynamically refreshes session encryption keys.

4 RESULT & DISCUSSIONS



Home Dashboard Logout

Welcome

Login Register

Figure 2: Home Page.

Figure 2 shows the Home Page's structure and essential components and user interface. It summarizes the key parts, navigation, and interactive features.

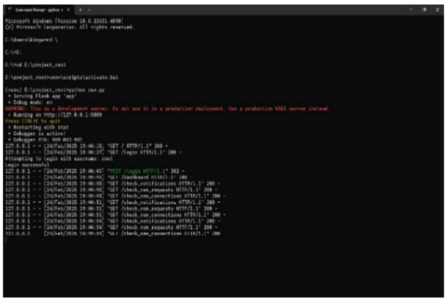
Home Dashboard Logout

Login

Username: root Password: •••• Login

Figure 3: Login Page.

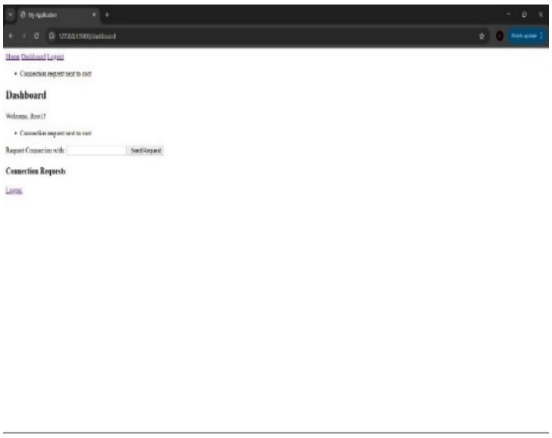
Figure 3 shows the Login Page's structure and user authentication interface. Credential input forms, validation messages, and secure login are included.



```
Microsoft Windows [Version 10.0.22631.4890]
(c) Microsoft Corporation. All rights reserved.

C:\Users\kinga>cd \

C:\>E:

E:\>cd E:\project_root

E:\project_root>venv\scripts\activate.bat

(venv) E:\project_root>python run.py
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 989-863-981
127.0.0.1 - - [24/Feb/2025 19:44:18] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [24/Feb/2025 19:44:27] "GET /login HTTP/1.1" 200 -
Attempting to login with username: root
Login successful
127.0.0.1 - - [24/Feb/2025 19:44:45] "POST /login HTTP/1.1" 302 -
127.0.0.1 - - [24/Feb/2025 19:44:45] "GET /dashboard HTTP/1.1" 200 -
127.0.0.1 - - [24/Feb/2025 19:44:48] "GET /check_notifications HTTP/1.1" 200 -
127.0.0.1 - - [24/Feb/2025 19:44:48] "GET /check_new_requests HTTP/1.1" 200 -
127.0.0.1 - - [24/Feb/2025 19:44:48] "GET /check_new_connections HTTP/1.1" 200 -
127.0.0.1 - - [24/Feb/2025 19:44:51] "GET /check_notifications HTTP/1.1" 200 -
127.0.0.1 - - [24/Feb/2025 19:44:51] "GET /check_new_requests HTTP/1.1" 200 -
127.0.0.1 - - [24/Feb/2025 19:44:51] "GET /check_new_connections HTTP/1.1" 200 -
127.0.0.1 - - [24/Feb/2025 19:44:54] "GET /check_notifications HTTP/1.1" 200 -
127.0.0.1 - - [24/Feb/2025 19:44:54] "GET /check_new_requests HTTP/1.1" 200 -
127.0.0.1 - - [24/Feb/2025 19:44:54] "GET /check_new_connections HTTP/1.1" 200
```

Figure 4: Login Backend.

Figure 4. the Login system's backend implementation shows authentication logic, database interaction, and security. It generates tokens, validates credentials, and controls access.



Home Dashboard Logout

- Connection request sent to root

Dashboard

Welcome, Root1!

- Connection request sent to root

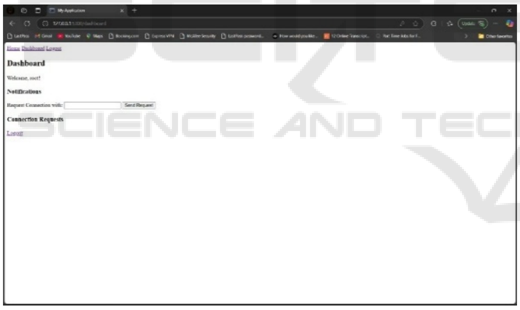
Request Connection with: [   ] Send Request

Connection Requests

Logout

Figure 5: User Dashboard.

Figure 5 illustrates the User Dashboard, displaying an overview of user activities, personalized data, and interactive features. It includes navigation options, account details, and essential functionalities for user engagement.



Home Dashboard Logout

Dashboard

Welcome, root!

Notifications

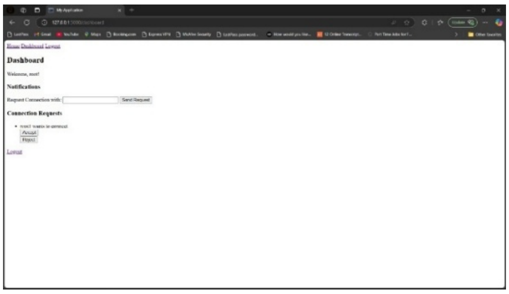
Request Connection with: [   ] Send Request

Connection Requests

Logout

Figure 6: Second User Dashboard.

Figure 6 shows the Second User Dashboard, which has a different layout or functionality for certain roles. Personalized information, better navigation, and role-based functions increase user experience.



Home Dashboard Logout

Dashboard

Welcome, root!

Notifications

Request Connection with: [   ] Send Request

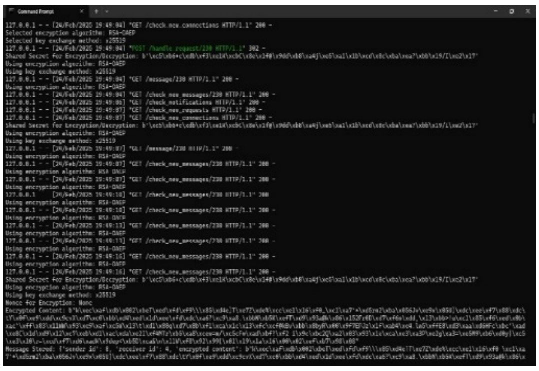
Connection Requests

- root1 wants to connect
  Accept
  Reject

Logout

Figure 7: Incoming Request.

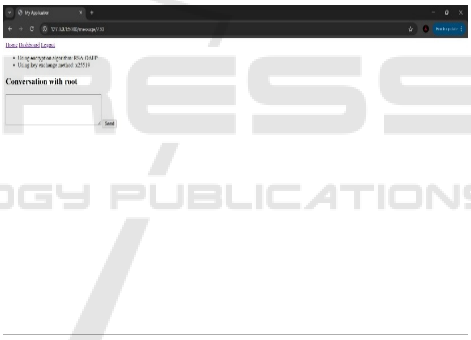
In Figure 7 the Incoming Request area shows organized user requests. Details include request sender, date, status, and request management or response actions.



```
127.0.0.1 - - [24/Feb/2025 19:49:04] "GET /check_new_connections HTTP/1.1" 200 -
Selected encryption algorithm: RSA-OAEP
Selected key exchange method: x25519
127.0.0.1 - - [24/Feb/2025 19:49:04] "POST /handle_request/230 HTTP/1.1" 302 -
Shared Secret for Encryption/Decryption: b'\xc5\xb6+c\xdb\xf3\xe1A\xcbC\x8e\x1f@\x9dd\xb8\xa4j\xe5\xa1\x1b\xcd\x8c\xba\xea?\xbb\x19/I\xe2\x17'
Using encryption algorithm: RSA-OAEP
Using key exchange method: x25519
127.0.0.1 - - [24/Feb/2025 19:49:04] "GET /message/230 HTTP/1.1" 200 -
Using encryption algorithm: RSA-OAEP
127.0.0.1 - - [24/Feb/2025 19:49:04] "GET /check_new_messages/230 HTTP/1.1" 200 -
127.0.0.1 - - [24/Feb/2025 19:49:06] "GET /check_notifications HTTP/1.1" 200 -
127.0.0.1 - - [24/Feb/2025 19:49:07] "GET /check_new_requests HTTP/1.1" 200 -
127.0.0.1 - - [24/Feb/2025 19:49:07] "GET /check_new_connections HTTP/1.1" 200 -
Shared Secret for Encryption/Decryption: b'\xc5\xb6+c\xdb\xf3\xe1A\xcbC\x8e\x1f@\x9dd\xb8\xa4j\xe5\xa1\x1b\xcd\x8c\xba\xea?\xbb\x19/I\xe2\x17'
Using encryption algorithm: RSA-OAEP
Using key exchange method: x25519
127.0.0.1 - - [24/Feb/2025 19:49:07] "GET /message/230 HTTP/1.1" 200 -
Using encryption algorithm: RSA-OAEP
127.0.0.1 - - [24/Feb/2025 19:49:07] "GET /check_new_messages/230 HTTP/1.1" 200 -
Using encryption algorithm: RSA-OAEP
127.0.0.1 - - [24/Feb/2025 19:49:07] "GET /check_new_messages/230 HTTP/1.1" 200 -
Using encryption algorithm: RSA-OAEP
127.0.0.1 - - [24/Feb/2025 19:49:10] "GET /check_new_messages/230 HTTP/1.1" 200 -
Using encryption algorithm: RSA-OAEP
127.0.0.1 - - [24/Feb/2025 19:49:10] "GET /check_new_messages/230 HTTP/1.1" 200 -
Using encryption algorithm: RSA-OAEP
127.0.0.1 - - [24/Feb/2025 19:49:13] "GET /check_new_messages/230 HTTP/1.1" 200 -
Using encryption algorithm: RSA-OAEP
127.0.0.1 - - [24/Feb/2025 19:49:13] "GET /check_new_messages/230 HTTP/1.1" 200 -
Using encryption algorithm: RSA-OAEP
127.0.0.1 - - [24/Feb/2025 19:49:16] "GET /check_new_messages/230 HTTP/1.1" 200 -
Using encryption algorithm: RSA-OAEP
127.0.0.1 - - [24/Feb/2025 19:49:16] "GET /check_new_messages/230 HTTP/1.1" 200 -
Shared Secret for Encryption/Decryption: b'\xc5\xb6+c\xdb\xf3\xe1A\xcbC\x8e\x1f@\x9dd\xb8\xa4j\xe5\xa1\x1b\xcd\x8c\xba\xea?\xbb\x19/I\xe2\x17'
Using encryption algorithm: RSA-OAEP
Using key exchange method: x25519
Nonce for Encryption: None
Encrypted Content: b"k\xec\xaf\xdb\x802\xbeT\xed\xfd\xf9\\\x85\xd4e]T\xe7Z\xde%\xcc\xe1\x16\xf0_\xc1\xa7'+\xd6rm2\xba\x866Jv\xe9x\x850]\xdc\xee\xf7\x88\xdc\ [illegible]
[illegible]
[illegible]
[illegible]
[illegible]"
Message Stored: {'sender_id': 8, 'receiver_id': 4, 'encrypted_content': b"k\xec\xaf\xdb\x802\xbeT\xed\xfd\xf9\\\x85\xd4e]T\xe7Z\xde%\xcc\xe1\x16\xf0 \xc1\xa [illegible]
7'+\xd6rm2\xba\x866Jv\xe9x\x850]\xdc\xee\xf7\x88\xdc\ [illegible]
```

Figure 8: Algorithm Selection.

Figure 8 shows the Algorithm Selection procedure, where users pick safe communication cryptographic methods. A collection of algorithms, selection criteria, and an encryption preferences interface are included.



Home Dashboard Logout

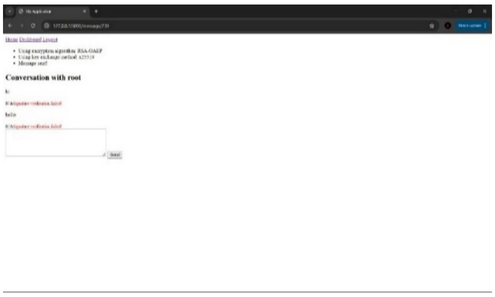
- Using encryption algorithm: RSA-OAEP
- Using key exchange method: x25519

Conversation with root

[   ] Send

Figure 9: Messaging Portal.

Figure 9 shows the Messaging Portal, which lets users send and receive encrypted communications securely. A chat window, message input field, encryption status indicators, and conversation management options are included.



Home Dashboard Logout

- Using encryption algorithm: RSA-OAEP
- Using key exchange method: x25519
- Message sent!

Conversation with root

hi

[illegible] signature verification failed

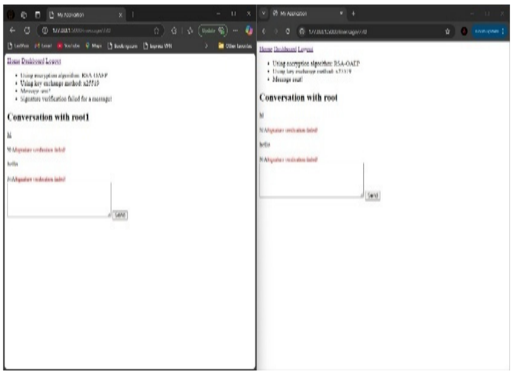
hello

[illegible] signature verification failed

[   ] Send

Figure 10: Transferring Messages.

Figure 10 shows how users safely transfer encrypted communications. Secure communication is ensured via encryption methods, message routing, delivery status, and decryption.



- Using encryption algorithm: RSA-OAEP
- Using key exchange method: x25519
- Message sent!
- Signature verification failed for a message!

Conversation with root1

hi

Signature verification failed!

hello

Signature verification failed!

- Using encryption algorithm: RSA-OAEP
- Using key exchange method: x25519
- Message sent!

Conversation with root

hi

Signature verification failed!

hello

Signature verification failed!

Figure 11: Transferred Messages.

In Figure 11, encrypted communications are successfully sent. It provides message information, timestamps, decryption status, and secure access to historical conversations.

Below is a comparative table that shows how the dynamic encryption model differs from conventional encryption approaches. Considerations including quantum resistance, data storage practices, key exchange systems, and overall security strength are assessed. Dynamic encryption is a privacy-focused and very robust alternative to traditional approaches. It rotates keys in real-time, has communications that self-destruct, and employs forward secrecy. This comparison highlights the need of constantly improving encryption methods in order to successfully combat contemporary cyber threats. Table 1 show the Comparison between Veilcomm and Traditional method.

Table 1: Comparison between Veilcomm and Traditional method.

FEATURE	VEILCOMM	TRADITIONAL METHOD
Encryption Type	Encryption techniques are dynamically chosen for each session (a mix of symmetric and asymmetric encryption).	Fixed encryption mechanism, usually AES or RSA.
Key Exchange	For safe key exchange, use X25519, X3DH, KYBER, or NTRU.	A Diffie-Hellman key exchange or an RSA type
Quantum Resistance	Accommodates post-quantum encryption algorithms (KYBER, NTRU)	Quantum assaults provide the greatest threat.
Forward Secrecy	Activated by the Double Ratchet Algorithm	Frequently inadequately executed or poorly endorsed
Algorithm Randomization	Each session employs a randomly chosen encryption technique.	Employs a uniform encryption technique over all sessions
Self-Destruction Messages	Messages may be programmed for automated deletion.	Most systems lack inherent message expiry features.
No Session Data Policy	Absence of storage for session keys or information on servers	Session data and metadata may be retained on servers.
Key Rotation	Immediate critical updates throughout the session	Generally, use a single key for each session

End to End Encryption(E2EE)	Utilised contemporary cryptographic standards (ChaCha20-Poly1305, AES-GCM)	Certain conventional systems use end-to-end encryption, but with predetermined algorithms.
Performance Efficiency	Enhanced with efficient encryption (ChaCha20, AES-GCM) for expedited processing	May need significant processing resources (e.g., RSA-2048)
Resistance to Data Breaches	Absence of saved information or encryption keys mitigates breach consequences.	Metadata and keys may be retained, hence increasing the danger of breaches.
Use Cases	Monetary exchanges, jurisprudence, commerce, medical services, public administration	Corporate communications and general messages

When compared to more conventional encryption systems, Dynamic Encryption Messaging System provides far higher levels of protection, adaptability, and anonymity. It makes assaults much more difficult by removing predictability via the random selection of encryption methods for each session. Additional safeguards against ever-changing cyber threats include features like self-destructing communications, real-time key rotation, and quantum-resistant encryption. Although they are still effective, traditional encryption techniques aren't flexible enough to withstand assaults from quantum computers.

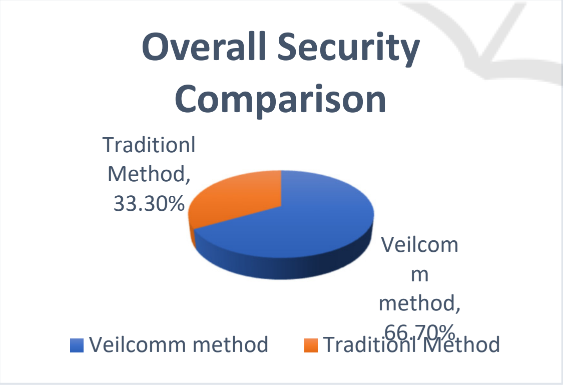


Figure 12: Comparison between Traditional and Veilcomm method.

The danger of breaches is further increased when metadata and session keys are stored. On the other hand, dynamic encryption's no-session-data policy guarantees complete anonymity, regardless of server penetration. Static encryption is insufficient due to the increasing sophistication of cyber-attacks. Dynamic Encryption is the best option for organisations and consumers concerned about privacy since it uses adaptive, multi-layered encryption techniques, which are the future of secure communication. Figure 12 show the Comparison between Traditional and Veilcomm method.

You can see how the two encryption techniques stack up against one another in terms of general security in the pie chart. A far bigger part is held by Dynamic Encryption, demonstrating its superiority in security characteristics. Although traditional encryption is still valuable, it is less prevalent, which shows that it isn't as well-suited to new threats and isn't as adaptable.

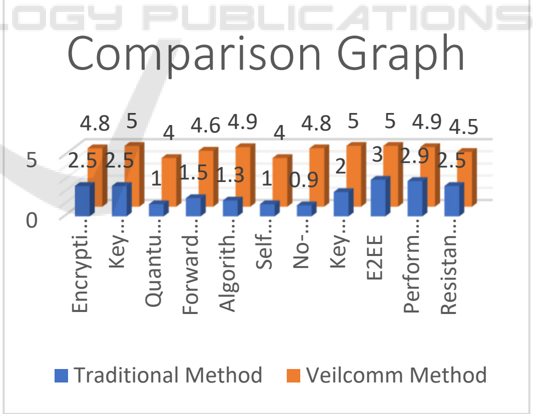


Figure 13: Comparison between Traditional and Veilcomm method.

Dynamic Encryption Messaging System and Traditional Encryption Methods are compared graphically in the bar graph across several parameters to show how strong their security is. For every metric, including forward secrecy, algorithm randomisation, key rotation, and quantum resistance, Dynamic Encryption always comes out on top, but Traditional

Encryption falls short, especially in those two categories. Because of this, dynamic encryption stands out as the best security strategy. Figure 13 show the Comparison between Traditional and Veilcomm method.

5 CONCLUSIONS

The challenge of enhancing messaging security by dynamically adapting encryption methods to counter evolving cybersecurity threats, performance issues, and regulatory pressures has been the central focus of this paper. Existing systems often rely on static encryption techniques, making them vulnerable to brute force attacks, quantum computing advancements, and legal mandates for backdoor access. The proposed system addresses these challenges through its innovative design, featuring dynamic encryption that randomly selects an encryption algorithm for each session, making communications significantly harder to compromise. Ephemeral key exchange mechanisms generate secure, temporary keys that cannot be reused or exploited to decrypt past communications, while the absence of session data storage minimizes the risk of breaches and unauthorized access. The platform integrates cutting-edge cryptographic techniques, including Elliptic Curve Cryptography (ECC), AES, Blowfish, Twofish, and Quantum Key Distribution (QKD), positioning it to withstand emerging quantum threats. Its modular architecture enables seamless integration of new encryption algorithms and security features, while horizontal scalability ensures consistent performance under increasing user bases and data volumes. The system prioritizes user privacy by avoiding centralized metadata storage and aligning with regulatory frameworks like GDPR and CCPA. This has far-reaching benefits across various sectors, enabling private communication for individuals in sensitive contexts such as legal consultations, medical advice, or personal conversations, while also facilitating secure exchanges of confidential information in business and enterprise environments, as well as government institutions handling national security data.

REFERENCES

Dabola, S., Tomer, V., Singh, N., Madan, P., & Jhinkwan, A. (2024). Chat Secure-Messaging Application Based on Secure Encryption Algorithm. International *Journal for Research in Applied Science & Engineering Technology* (IJRASET), 12(1). DOI: 10.22214/ijraset.2024.58817.

Pandey, D., Gupta, R. R., & Choudhary, A. (2023). A Novel Development of Blockchain-based Messaging Application. *2023 International Conference on Advances in Electronics*, Communication, Computing and Intelligent Information Systems (ICAECIS). DOI: 10.1109/ICAECIS58353.2023.10170701.

Gour, A., Malhi, S. S., Singh, G., & Kaur, G. (2024). Hybrid Cryptographic Approach: For Secure Data Communication using Block Cipher Techniques. *E3S Web of Conferences,* 556, 01048. DOI: 10.1051/e3sconf/202455601048

Shamsi, S. E., & Nasrat, B. W. (2024). Securing Messages and Files Using Integrated Steganography and Asymmetric Encryption Techniques. *Cognizance Journal of Multidisciplinary Studies*, 4(4), 272-282. DOI: 10.47760/cognizance. 2024.v04i04.019

AlSideiri, A., AlShamsi, S., AlBreiki, H., AlMoqbali, M., AlMaamari, M., & AlSaadi, S. (2024). Cybersecurity Enhancement through Hybrid Encryption: Combining RSA and Vigenère Algorithms in the Cypher-X System. Baghdad Science *Journal, 21(5 Special Issue),* 1765-1774. DOI: 10.21123/bsj.2024.10539.

Shuhan, M. K. B., Islam, T., Shuvo, E. A., Bappy, F. H., Hasan, K., & Caicedo, C. (2023). Quarks: A Secure and Decentralized Blockchain-Based Messaging Network. *IEEE International Conference on Cyber Security and Cloud Computing* (CSCloud 2023). DOI: 10.48550/arXiv.2308.04452.

Yeung, M. (2024). SendingNetwork: Advancing the Future of Decentralized Messaging Networks. arXiv preprint arXiv:2401.09102v2. DOI: 10.48550/arXiv.2401.09102v2

Cui, Li, Qianqian, Xing, Yi, Wang, Baosheng, Wang, Jing, Tao, Liu, Liu, FSEE: A Forward Secure End-to-End Encrypted Message Transmission System for IoT, *Security and Communication Networks*, 2022, 2644716, 18 pages, 2022.

Len, J., Ghosh, E., Grubbs, P., & Rösler, P. Interoperability in End-to-End Encrypted Messaging. 2023.

N. Unger et al., "SoK: Secure Messaging," *2015 IEEE Symposium on Security and Privacy, San Jose*, CA, USA, 2015, pp. 232-249, doi: 10.1109/SP.2015.22.

Chopra, Aakanksha. (2015). Comparative Analysis of Key Exchange Algorithms in Cryptography and its Implementation. IMS Manthan *(The Journal of Innovations)*. 8. 10.18701/imsmanthan. v8i2.5126.

Mashari Alatawi and Nitesh Saxena. 2023. SoK: An Analysis of End-to-End Encryption and Authentication Ceremonies in Secure Messaging Systems. In Proceedings of the *16th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec '23). Association for Computing Machinery*, New York, NY, USA, 187–201.

K. Cohn-Gordon, C. Cremers, B. Dowling, L. Garratt and D. Stebila, "A Formal Security Analysis of the Signal Messaging Protocol," *2017 IEEE European Symposium on Security and Privacy* (EuroS&P), Paris, France, 2017, pp. 451-466, doi: 10.1109/EuroSP.2017.27.

Puyosa, I. (2023). Protecting Point-to-Point Messaging Apps: Understanding Telegram, WeChat, and WhatsApp in the United States and in Turkey. Digital Forensic Research Lab (DFRLab), Atlantic Council

Mishra, M. R., & Kar, J. (2017). A study on diffie-hellman key exchange protocols. *International Journal of Pure and Applied Mathematics*, 114(2), 179-189. DOI: 10.12732/ijpam. v114i2.2.

Phan, Duong & Pointcheval, David. (2004). OAEP 3-Round: *A Generic and Secure Asymmetric Encryption Padding*. 3329. 63-77. 10.1007/978-3-540-30539-2_5.

Hoffstein, J., Pipher, J., Silverman, J.H. (2001). NSS: An NTRU Lattice-Based Signature Scheme. In: Pfitzmann, B. (eds) Advances in Cryptology EUROCRYPT 2001. EUROCRYPT 2001. Lecture Notes in Computer Science, vol 2045. Springer, Berlin, Heidelberg.

Alwen, J., Coretti, S., Dodis, Y. (2019). The Double Ratchet: Security Notions, Proofs, and Modularization for the Signal Protocol. In: Ishai, Y., Rijmen, V. (eds) Advances in Cryptology EUROCRYPT 2019. EUROCRYPT 2019. Lecture Notes in Computer Science (), vol 11476. Springer, Cham.

Serrano, R., Duran, C., Sarmiento, M., Pham, C.-K., & Hoang, T.-T. (2022). ChaCha20–Poly1305 Authenticated Encryption with Additional Data for Transport Layer Security 1.3. Cryptography, 6(2), 30.

Kim, Kyungho, Seungju Choi, Hyeokdong Kwon, Hyunjun Kim, Zhe Liu, and Hwajeong Seo. 2020. "PAGEPractical AES-GCM Encryption for Low-End Microcontrollers" *Applied Sciences* 10, no. 9: 3131.

Leon Botros, Matthias J. Kannwischer, and Peter Schwabe. 2019. Memory-Efficient High-Speed Implementation of Kyber on Cortex-M4. In Progress in Cryptology – AFRICACRYPT 2019: 11th International Conference on Cryptology in Africa, Rabat, Morocco, July 9–11, 2019, Proceedings. Springer-Verlag, Berlin, Heidelberg, 209–228.

A. Ruggeri, A. Galletta, A. Celesti, M. Fazio and M. Villari, "An Innovative Blockchain Based Application of the Extended Triple Diffie-Hellman Protocol for IoT," 2021 *8th International Conference on Future Internet of Things and Cloud* (FiCloud), Rome, Italy, 2021, pp. 278-284, doi: 10.1109/FiCloud49777.2021.00047.

J. Dong, F. Zheng, J. Cheng, J. Lin, W. Pan and Z. Wang, "Towards High-performance X25519/448 Key Agreement in General Purpose GPUs," *2018 IEEE Conference on Communications and Network Security (CNS),* Beijing, China, 2018, pp. 1-9, doi: 10.1109/CNS.2018.8433161.