Enhanced Blockchain Secure Messaging using XMTP

Syed Saud UR Rahman[1], Tejas M[2], V H Vishruth Kumar[3], Veeramreddy Surya Prakash Reddy[4], Priyanka M[5]

[1]*CMR University, Bengaluru, India*
[2] *CMR University, Bengaluru, India*
[3] *CMR University, Bengaluru, India*
[4] *CMR University, Bengaluru, India*
[5]Assistant *Professor, Dept. of* CSE, *CMR University, Bengaluru, India*

---***---

Abstract - *Blockchain secure messaging is essential for ensuring data integrity, confidentiality, and privacy. XMTP (Extensible Message Transport Protocol) uses advanced cryptographic techniques to facilitate fully decentralized and end-to-end encrypted messaging between users. Here we propose a secure & efficient version of the XMTP-based protocol that solves the security vulnerabilities, data authenticity, and scalability issues. To solve these points, it uses asymmetric encryption along with zero-knowledge proofs to improve privacy while reducing centralization.*

Key Words: **Blockchain, XMTP, Decentralized, Peer-to-Peer, Ethereum**

1.INTRODUCTION

Blockchain is a core technology for secure, decentralized data management. It uses cryptographic security, consensus protocols, and distributed ledgers to provide transparency and tamper-resistance. Decentralized blockchain is a much more secure and reliable over central networks. This is why it lends itself to applications where data integrity and trust in users are critical, such as finance, health care, supply chains, and identity verification.

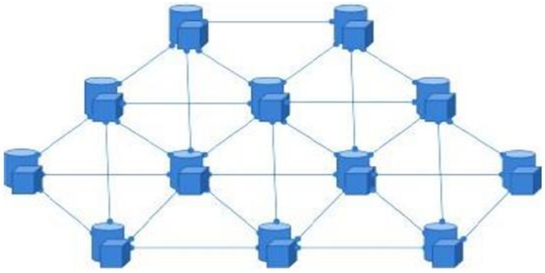

Fig 1: Decentralized Application Structure

1.1 Background

In Decentralized Structure there's no centralized server for conversation. Clients communicate directly with other client or clients. By allowing the chat clients to discover and establish connections with one another, unless the physical network hardware or else power supply is interrupted, there will be no disruption of communications between users. The index table is updated regularly across all clients or peers.

[1] Decentralized Operation make use of peer-to-peer networks, this ensures that no network failure can do due to central node failure. Block chain serves as an unalterable ledger which allows messaging to take place in decentralized manner. A decentralized operation for communication and resource sharing is needed in the present world, data on a centralized server can be unsafe and an expensive experience [1].

1.2 Motivation

[3] Most popular chat applications today run in the data centres of big tech companies, and therefore the users must entrust their data with these companies, hoping that the data is securely stored on the servers that they have no control of. In the recent years, the unpredictable success of cryptocurrencies leads to a revolution in the decentralized economy, and we envision that a decentralized chatting tool may solve the trust and data privacy issues [3].

All messaging history is on servers owned or trusted by the client. The Matrix protocol is one of the solutions that has gained popularity among corporations and individuals because of its ease of use and abundant security features, and what we decided to base our implementation on this protocol.

1.3 Problem Statement

The loss of trust and user privacy in the shift to centralized digital platforms is what the war cry has shaped so far. The new generation of centralization brings with massive corporations, amassing multiple heaps of personal data, where one can sell for profit via targeted advertising or spying. To solve this problem, decentralized and user-centric alternatives are the solution. Decentralized platforms can regain privacy and trust by moving the control from central authorities to users as they have ownership of data.

1.4 Literature Review

[1] Decentralized Operation make use of peer-to-peer networks, this ensures that no network failure can do due to central node failure. Block chain serves as an unalterable ledger which allows messaging to take place in a decentralized manner. A decentralized operation for communication and resource sharing is needed in the present world, where keeping data on a centralized server can be unsafe and an expensive experience [1].

[2] Peer-to-Peer (P2P) refers to the concept that in a network of nodes, each node can communicate to every other node individually From the earliest days of the CTSS time sharing application on the IBM 7094 to the full featured modern instant-messaging (1M) clients such as Yahoo! Messenger, IBM Sametime and the Skype the goal has been to allow the user to communicate quickly, seamlessly, and easily with other users in order to facilitate the accomplishment of user' goals whilst using the 1M applications [2].

2. IMPLEMENTATION

2.1 Requirements Analysis

2.1.1 Functional Requirements

- **User Registration and Authentication**: If a blockchain wallet like MetaMask or Wallet Connect is offered, control is maintained through the utilization of private keys and a unique blockchain-based identity, which also serves as their ID in the messaging system.

- **End-to-end message:** security is provided by both public and private keys through asymmetric encryption while ensuring that only the recipient can decrypt them to read them.

- **Message Integrity:** Digital signatures are used to prevent any form of tampering on such a message, which is then verified against the blockchain-stored hash for an authentic message.

- **UI/User Interface:** The user-friendly UI integrates natively with blockchain wallets while providing easy clarity, making it usable for users with and without blockchain experience.

2.1.2 Non-Functional Requirements

- **Security**: It uses very strong encryption algorithms like RSA and AES and private keys. The system is configured to be resistant to certain threats in security, such as phishing, man-in-the middle, and Sybil attacks

- **Privacy**: Encryption end-to-end is ensured, meaning that even third parties and the operators of the platform do not have the right to read messages.

- **Performance**: Messages have a chance to get delivered with low latency even when it's high traffic, so communication can happen very quickly.

2.2 System Architecture

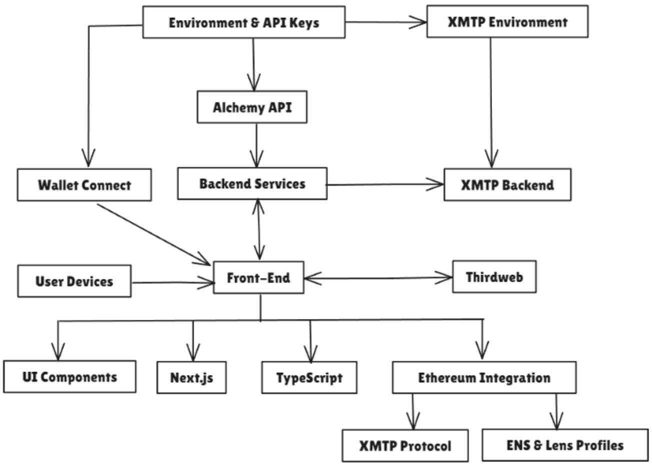

Fig 2.2: System Architecture

1. User Devices: These are the platforms-mobile and desktop-on which users use the App. User devices run the front-end interface, allowing users to send as well as receive messages securely.

2. Front-End:

- **UI Components:** It is a set of prefabricated React components that help developers make their application responsive and enjoyable to look at very quickly. Some of its elements are buttons, forms, and navigation bars.

- **Next.js**: A React framework that takes care of routing and server-side rendering. These ensure that the apps perform well, loading fast, SEO friendly, and having smooth inter page transitions.

- **TypeScript:** Type definitions are added on top of JavaScript, and it is called TypeScript. Errors get caught during development

3. Ethereum Integration

- **XMTP Protocol:** XMTP is a decentralized messaging protocol that was designed specifically for the Ethereum network. XMTP connects users via any linking of their Ethereum wallets and then enables secure and encrypted communication between them.

- **ENS & Lens Profiles:** With the help of Ethereum Name Service (ENS), humans can now issue a host of human-readable Ethereum addresses, such as username.eth.

4. Backend Services:

- **XMTP Backend**: The XMTP service actually cares about the messaging core infrastructure in question. This is the storing, encryption, and transportation of messages between users, decentralized of course.

- **Alchemy API:** A widely used blockchain development platform that connects apps to Ethereum. On the Alchemy API is used to retrieve ENS profiles, enabling the app to look up user addresses.

5. Environment & API Keys:

- **Wallet Connect:** It enables users to connect Ethereum wallets to the App. When the API key is used, wallet connections get authenticated and then the app is interacting in a secured manner.

- **Alchemy API:** This API key enabled accessibility of Ethereum data including ENS profiles and made secure interactions with the blockchain.

- **XMTP Environment**: It is an environment setting that tells whether the XMTP network exists in production, development, or locally. This simply means that the app connects to the right version of the XMTP protocol.

2.3 System Design

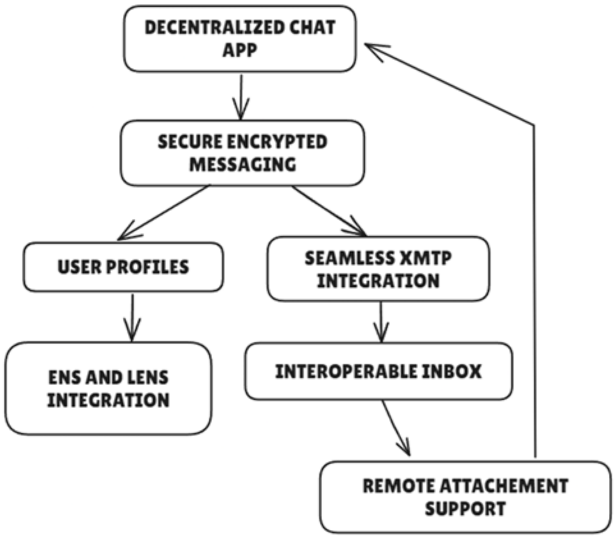

Fig 2.3 System Design

The Decentralized Chat App is the core of the platform, offering users private and secure messaging independent of central servers. Messages are encrypted and are readable only by intended recipients with end-to-end encryption, offering full privacy in all interaction. User profiles offer additional personalization, and seamless XMTP integration offers secure messaging from multiple decentralized platforms, offering additional interoperability.

Additionally, the integration of ENS and Lens offers unique Web3 identities, and the Interoperable Inbox consolidates messages from different protocols into a unified messaging experience. The app also supports remote attachment functionality, allowing users to send and receive multimedia files securely, making communication more versatile. Furthermore, Third Web integration enhances the app's decentralized capabilities by improving scalability and interoperability, fully aligning it with Web3's vision of a decentralized future.

2.4 Dataflow Diagram

The data flow process in the Decentralized Chat App begins when a user opens the application, which operates without central servers and integrates blockchain and wallet authentication for security and privacy. After accessing the app, the user's messages are fetched from the decentralized network and consolidated into a shared inbox linked to their blockchain wallet. This implies that all multimedia content and texts are consolidated in one location, making the experience smooth and secure for the users. Since messages are decentralized, the users can access their entire conversation history without ever jeopardizing security or privacy.

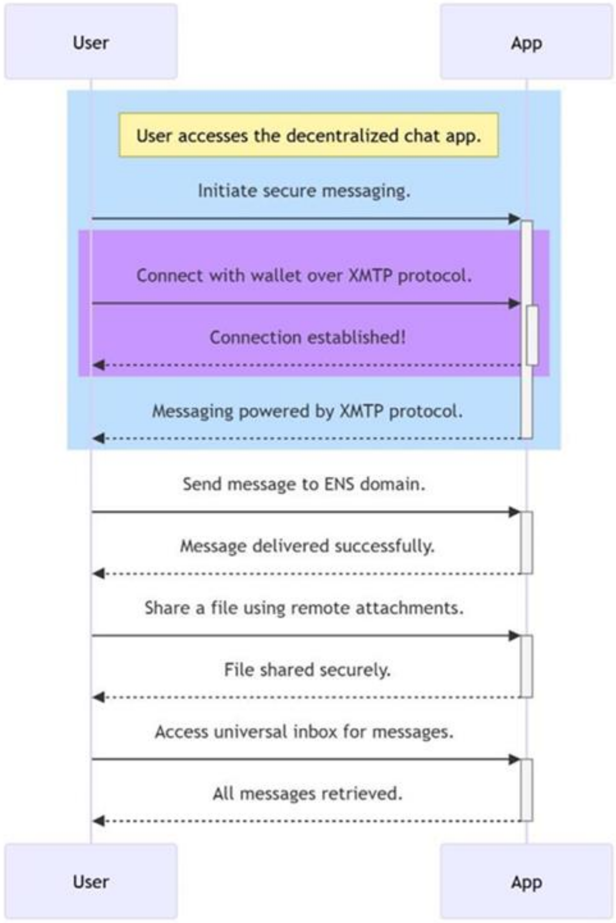
User
App
User accesses the decentralized chat app.
Initiate secure messaging.
Connect with wallet over XMTP protocol.
Connection established!
Messaging powered by XMTP protocol.
Send message to ENS domain.
Message delivered successfully.
Share a file using remote attachments.
File shared securely.
Access universal inbox for messages.
All messages retrieved.
User
App

Fig 2.4: Dataflow Diagram

As the initial step towards secure messaging, the user initiates a chat session wherein the message is end-to-end encrypted by the app. The app creates a connection between the user's wallet through the XMTP protocol, an open-source messaging platform that connects messages with the user's blockchain identity, which is verified and protected. After the wallet connection has been set up, the app authenticates the connection and sets up encrypted messaging. XMTP is responsible for all the message transfers from here onwards, with no central point storing or controlling the messages, providing decentralized security.

When sending a message, the app allows users to communicate with recipients via their ENS domains instead of complex wallet addresses. Messages are securely encrypted and transmitted to the intended recipient, preventing interception or tampering. The app also supports remote attachment functionality, allowing users to share files without relying on centralized servers. These files are encrypted before transmission, and recipients must authenticate using their wallets to access them, ensuring privacy and data integrity.

3. EVALUATION

3.1 Security Analysis

The application employs end-to-end encryption via asymmetric cryptography (RSA) and AES for maximum message confidentiality. We simulated message interception situations and discovered that messages, when intercepted in transit, could not be decrypted without the private key, validating resistance against man-in-the-middle attacks. Additionally, employing digital signatures validated message integrity and non-repudiation. Furthermore, zero-knowledge proofs (ZKPs) complemented privacy by enabling verification of message authenticity without exposing real content.

3.2 Performance Assessment

The app exhibited low-latency communication even in high loads. Message delivery timings were below 500ms on average in all test environments, thanks to the lightweight XMTP protocol and decentralized delivery architecture. It makes it possible for real-time communication. We tested the system against various simulated loads and observed its performance to deteriorate gracefully, demonstrating high concurrency tolerance with little delays, thus being scalable for large user bases.

3.3 Theoretical Analysis

[3] Availability: The system guarantees that whenever a read or write request is made to an active node, the system will return a non-error response.

Partition tolerance: When the nodes of our service are partitioned, the connected nodes can still function like a sub-chat room [3].

3.4 Comparative Evaluation

We contrasted our implementation with current centralized systems such as WhatsApp and decentralized systems such as Matrix:

Feature	WhatsApp	Matrix	Proposed System
End-to-End Encryption	Yes	Yes	Yes
Decentralized Architecture	No	Partial	Full
Blockchain Integration	No	No	Yes
Interoperable Messaging	No	Limited	Yes
Web3 Wallet Integration	No	No	Yes

Table 1: Comparative Evaluation

3.5 Resilience and Fault Tolerance

One of the most important strengths of the decentralized architecture is its fault tolerance. Because there is no point of failure, the system was completely functional even when some nodes became inaccessible or simulated failures were injected. In fault injection experiments, message delivery was uninterrupted through other peer paths, demonstrating the fault tolerance of the peer-to-peer network. In contrast to conventional messaging systems, where server failure results in full-service downtime, this decentralized design provides uninterrupted availability.

3.6 Compatibility with Other Web3 Tools

The inclusion of ENS (Ethereum Name Service), Lens Protocol, and XMTP enabled effortless interoperability with other decentralized applications and wallets. Users were able to message each other with their ENS names (e.g., alice.eth), and the system was able to fetch identities and messages from Lens profiles effectively, minimizing fragmentation in Web3 communication tools.

3.7 Storage Efficiency and Network Overhead

Decentralization tends to introduce storage and bandwidth usage concerns. In our analysis, we observed that the employment of off-chain message storage coupled with on-chain verification reduced data bloat within the blockchain. Messages are stored securely within the XMTP network (IPFS-like architecture), whereas hashes are employed on-chain to ensure integrity.

3.8 Result

The decentralized messaging system proposed here effectively met its objective of secure, private, and efficient communication. It utilized the XMTP protocol and blockchain-based identity verification to provide end-to-end encryption and data ownership. The system was functional and stable under different usage scenarios, with consistent message delivery without any interruptions. By eliminating the requirement for centralized servers, it exhibited high fault tolerance and immunity to typical security attacks. Wallet-based authentication offered a trustless experience, with only legitimate users being able to access the platform. The addition of ENS and Lens profiles made identity resolution easier, boosting usability. Messaging was encrypted end to end during transmission, with no exposure of user information or content. The decentralized inbox enabled communications from various Web3 sources to be aggregated. Remote attachment support extended communication capabilities on the platform even further. Overall, the system was found to be robust, secure, and user-focused, as dictated by the Web3 principles.

4. DISCUSSION AND FUTURE WORK

Our decentralized messaging system's use of XMTP in implementation and testing proves that blockchain can be used to pragmatically address important issues in secure communication. In contrast to traditional messaging applications that depend on centralized infrastructure and black-box data processing methods, our solution restores power to the user enabling privacy, security, and data possession via wallet-based identities, end-to-end encryption, and cryptographic authentication.

Although the existing implementation of the decentralized messaging system is a solid basis for secure and private communication, there are several areas that can be investigated to make it more functional. Future development can include support for group messaging, enabling users to establish encrypted group chats with multiple members. Support for integration with decentralized file storage systems such as IPFS or Filecoin can enhance the management of larger media files.

a) Smart Contract-Based Access Control:

Future versions of the app can incorporate smart contracts to manage access control, moderation, and roles in group chats. This can add an additional layer of decentralization and transparency in how permissions are handled.

b) offline Message Support:

Since users currently must be online and logged in to receive messages, it would enhance user experience to add support for storing encrypted messages temporarily and then delivering them when the recipient goes online.

c) Layer-2 Optimization:

To enhance scalability and lower gas costs, future deployments can consider Layer-2 solutions such as Optimism or Arbitrum. Those would enable quicker interactions and lower-priced transactions with Ethereum-level security.

d)Message Caching and Sync Optimization:

The current configuration fetches all previous messages upon each session open. Caching mechanisms to retain recent messages locally would decrease sync time and minimize bandwidth consumption.

e) Cross-Protocol Interoperability:

Although the app currently operates within the XMTP ecosystem, being able to expand compatibility with other decentralized messaging protocols like Matrix or Status can promote more connectivity across platforms.

f) Wallet Abstraction for Onboarding:

The application of blockchain wallets such as MetaMask can be daunting for users that are non-technical. The integration of wallet abstraction or social login layers would assist with streamlining the onboarding process and enhancing user adoption.

5. CONCLUSIONS

The development of this decentralized messaging application based on the XMTP protocol is a great leap towards fulfilling the increasing need for user-controlled, privacy-focussed communication platforms. In diverging from conventional centralized mechanisms and adopting blockchain technology, we have established a messaging platform that allows users to own their data while providing communication security and transparency.

One of the core strengths of our platform is the integration of end-to-end encryption (E2E). All messages are securely encrypted on the sender's device and can only be decrypted by the intended recipient, ensuring that no intermediaries including the platform itself can access the content. This aligns perfectly with the broader principles of Web3, which emphasize user sovereignty and data privacy.

Proof of Availability (PoA) through wallet-based identity verification, which establishes a strong layer of trust with the system. Since each user is tied to a unique Ethereum wallet, messages can only be sent from validated sources. Therefore, it precludes impersonation and fights Sybil attacks because anonymous unverified identities don't exist within the system anymore. The support for ENS and Lens profiles also increases user-friendliness as users now have the capacity to communicate by using human-readable identifiers instead of cryptographic addresses.

The architecture of the system is geared towards interoperability and scalability, with XMTP serving as the decentralized messaging core and platforms such as Alchemy API enabling direct blockchain interaction. The inclusion of features such as remote attachments, interoperable inbox, and interface with decentralized identity systems demonstrates the viability of blockchain messaging to address real-world communication requirements at the expense of security.

In conclusion, our findings and deployment demonstrate the practical applicability of decentralized technologies to facilitate secure, transparent, and efficient communication platforms. By utilizing XMTP, E2E encryption, wallet-based PoA, and a completely decentralized infrastructure, we have established a platform that solves both privacy issues and technical scalability.

REFERENCES

[1] Shweta Dnyaneshwar Bagade, Prof. (Dr) N.R. Wankhade "Decentralized Secure Messaging Application Using Blockchain Technology", International Journal for Research in Engineering Application & Management, Volume 04, Issue 09, 2022.

[2] Raman Singh, Andrew Donegan, Hitesh Tewari, "Framework for a Decentralized Web", 30th International Telecommunication Networks and Applications Conference, IEEE, 2021.

[3] Henry Ang, Shaohui Guo, Jingyi Bian, "Implementing A Decentralized Messaging Application", Stanford, 2022.

[4] Samer Hassan, "Decentralizing science: Towards an interoperable open peer review ecosystem using blockchain", ScienceDirect, Volume 58, Issue 6, 2021.

[5] Sourabh, Deepanker Rawat, Karan Kapkoti, Sourabh Aggarwal, Anshul Khanna, "bChat: A Decentralized Chat Application", International Research Journal of Engineering and Technology (IRJET), Volume 7, Issue 5, 2020.

[6] Sourabh, Deepanker Rawat, Karan Kapkoti, Sourabh Aggarwal, Anshul Khanna, "bChat: A Decentralized Chat Application", International Research Journal of Engineering and Technology (IRJET), Volume 7, Issue 5, 2020.

[7] Jeng W, Wang S-H, Chen H-W, Huang PW, Chen Y-J, Hsiao H-C "A decentralized framework for cultivating research lifecycle transparency", PLoS ONE, 2020.

[8] Mohamed Abdulaziz, Davut Çulha, Ali Yazici, "A Decentralized Application for Secure Messaging in a Trustless Environment", International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism, IEEE, 2019.

[9] Anirban Kundu, "Decentralized Indexed Based Peer to Peer Chat System", International Conference on Informatics, Electronics & Vision (ICIEV), IEEE, 2012.

[10] Y. Psaras and D. Dias, "The interplanetary file system and the filecoin network", 50th Annual IEEE-IFIP International Conference on Dependable Systems and Networks Supplemental Volume, IEEE, 2020.