END-TO-END ENCRYPTION USING BLOCKCHAIN

Vedasree Anusha - 20BCE0776
Devanshi Jajodia - 20BCE2749
Payal Maheshwari - 20BCE2759
Mohammed Shabbir-20BCE2719
Harsh Rajpal - 20BCI0271

J component - Review 1

CSE3502 - Information Security Management

Supervisor: Prof. Ruby D.

School of Computer Science and Engineering (SCOPE)
Vellore Institute of Technology, Vellore

April, 2023

Table of Contents

I. ABSTRACT **3**

II. INTRODUCTION **4**

a. Technologies used 4

b. Social Impact 5

c. Expected Result 5

III. LITERATURE SURVEY **7**

IV. SYSTEM DESIGN **29**

a. Architecture Diagram 29

b. Flow Diagram 30

d. Algorithm & techniques 31

V. IMPLEMENTATION & GOAL ACHIEVED **35**

VI. RESULT AND DISCUSSION **39**

VII. CONCLUSION AND FUTURE SCOPE **40**

VIII. REFERENCES **40**

I. ABSTRACT

Users are becoming increasingly aware of the data privacy risks linked with social media and messaging apps in this day and age. However, the present security measures used by various service providers are not genuine E2EE (End-to-end encryption) implementations and are having several vulnerabilities. At the moment, the majority of the E2EE mechanism is managed by the service provider's servers, which also hold the decryption keys in case of backup restoration. So the major flaw in having a centralized architecture is the risk of a single point of control or failure. These flaws erode users' trust in the privacy of their data when utilizing these apps.

As a result, we propose a blockchain-enabled E2EE platform capable of providing true end-to-end encryption. The backup technique used by WhatsApp poses a significant risk to the secrecy of those messages. The WhatsApp server will always be free to decode messages, but hackers can duplicate the backup and deceive the WhatsApp server into sending them the decryption key. This is also an issue of single point of failure and control. As a result, the centralized architecture of WhatsApp and comparable older apps is to blame for this issue. A decentralized architecture proposed by us will make the data immutable, and not only provide better confidentiality and authentication. For attacking our system, an attacker would need to gain access to the vast majority of nodes on the network to collect any specific user's conversation data which is much more challenging than just hacking into a central server. We demonstrate the working of the proposed system by implementing each component in python language and flask web framework.

Keywords: End to end encryption, Blockchain implementation, authentication and authorization with decentralized architecture, Proof of work, Immutability.

II. INTRODUCTION

A blockchain is a decentralized, peer-to-peer database that may hold an ever-increasing number of transactions. No single person or institution has complete control over the information blocks because they are exchanged and stored throughout a network of nodes managed by people. This eliminates the possibility of a single person or entity changing records without the knowledge of others. Existing messaging apps use their own servers as trusted organizations to provide end-to-end encryption. If users use commercial messaging services, the encryption keys are stored on the organization's server. As a result, contemporary messaging systems may be vulnerable to unauthorized individuals accessing them in a group.

Messages will be stored inside blocks in our proposed decentralized system, and each block will contain the hash of the preceding block saved with it. As a result, data is dispersed, eliminating single point failures. Furthermore, because the data is unchangeable within the blockchain, attackers in the middle cannot edit it. Because the communications will be kept on a distributed network over the blockchain rather than on the user's local device, it will save storage and our decentralized chatting application will allow users to interact with one another without the need for a central authority's intervention.

a. Technologies used

The technologies that we will be implementing are:

1. Encryption with RSA (Preserves Data Confidentiality)
2. Storing transactions into blocks
3. Adding digital fingerprints to the blocks (Authentication)
4. Implementation of Proof of work algorithm
5. Establish consensus and decentralization
6. Solidity
7. React

The project will be created using flask as backend and blockchain for storing the encrypted messages. The messages will be encrypted using the RSA algorithm.

RSA Algorithm:

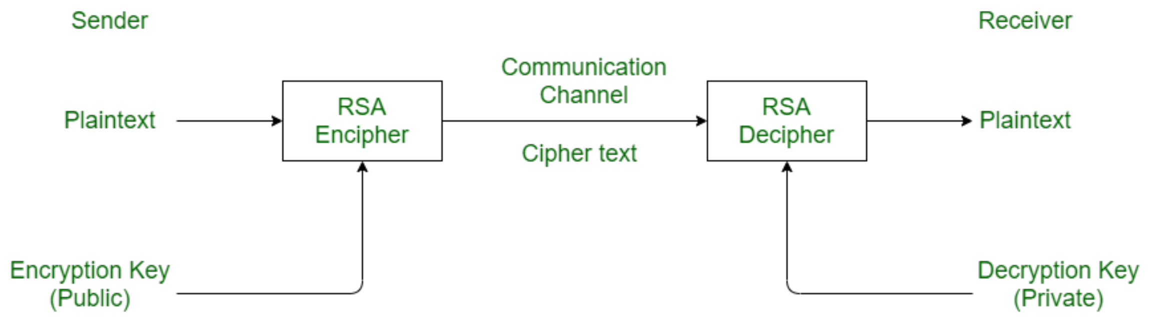


Fig 1: RSA Algorithm

b. Social Impact

Existing messaging programmes use their own servers as trusted entities to provide end-to-end encryption. If users are making use of corporate messaging services, the encryption keys are kept on the organization's server. The present messaging systems are open to illegal individuals accessing them collectively. The lack of reliable user authentication by third parties is the main cause of these vulnerabilities.

To prevent such vulnerabilities and third party attacks from happening, we are aiming to create a project on end-to-end encryption of messages using blockchain technology.

c. Expected Result

The communications will be kept in blocks in our decentralized system proposal, and each block will also keep a copy of the previous block's hash. Since the encryption decryption process will happen at the client side rather than on centralized servers, which gives authorization, this design will make the data immutable and allow it to be backed up and restored.

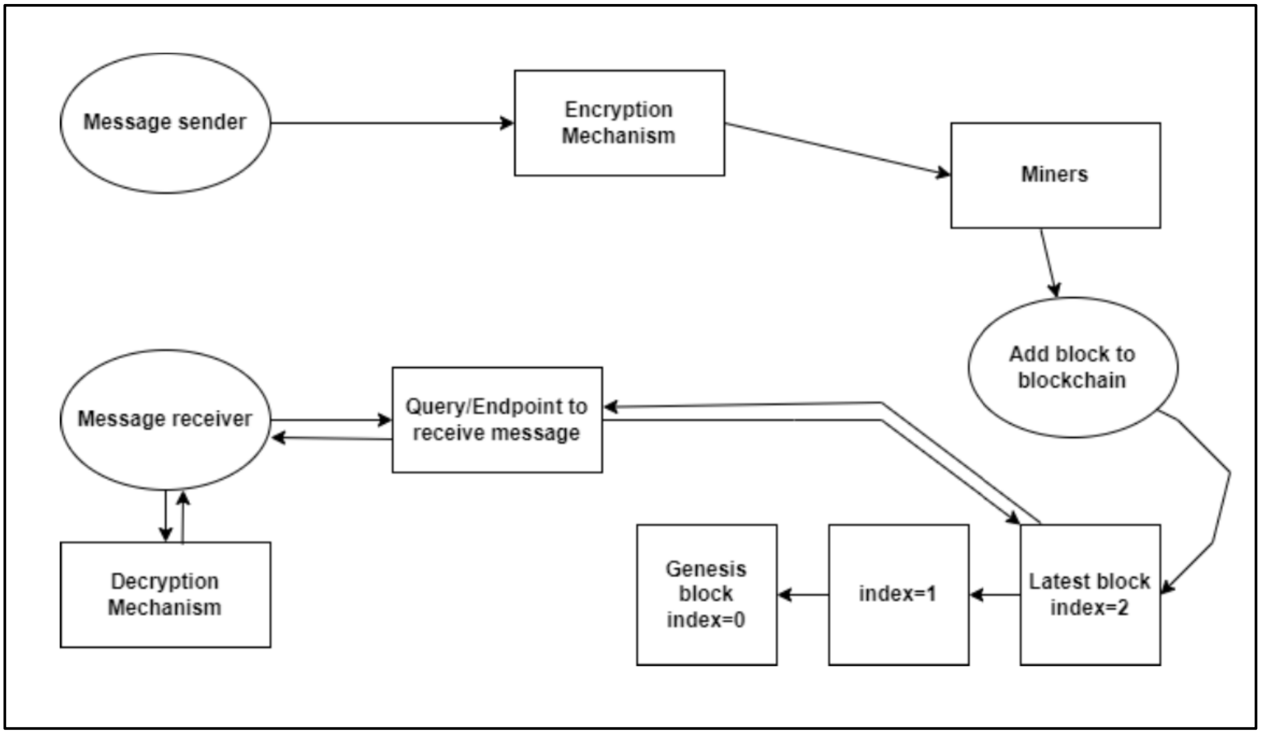


Fig 2: Expected result

III. LITERATURE SURVEY

1.
Name of the paper: Enabling (End-to-End) Encrypted Cloud Emails With Practical Forward Secrecy
Link of the paper:
https://www.computer.org/csdl/journal/tq/5555/01/09340272/1qMKylFI4g0
Authors: Jianghong Wei, Xiaofeng Chen, Jianfeng Wang, Xuexian Hu, Jianfeng Ma
Published on: July - August, 2022
Summary: In a paper titled "Enabling (End-to-End) Encrypted Cloud Emails With Practical Forward Secrecy," a technique for secure email transmission in the cloud is presented. End-to-end encryption and forward secrecy are intended to be provided, meaning that even if the encryption keys are stolen, previous emails are still safe. The authors suggest a system for safely exchanging encryption keys that combines public key encryption with key-exchange protocols. The system is functional and simple to use, making it a viable option for cloud-based encrypted email communication.

2.
Name of the paper: New Lightweight Hybrid Encryption Algorithm for Cloud Computing by using new 5D hyperchaos system
Link of the paper: https://turcomat.org/index.php/turkbilmat/article/view/4865
Authors: Muhned Hussam
Published: 2021
Summary: A new encryption method created specifically for cloud computing is described in the paper "New Lightweight Hybrid Encryption Algorithm for Cloud Computing (LMGHA-128bit) by employing new 5-D hyperchaos system." The LMGHA-128bit method combines aspects of symmetric and asymmetric encryption to maximise security while preserving speedy encryption and decryption procedures. The technique, according to the authors, offers 128-bit security, and the inclusion of a 5-D hyperchaos system improves encryption performance. The method may be used to secure data in cloud computing systems because it is also compact and appropriate for usage in contexts with limited resources.

3.
Name of the paper: Performance evaluation and analysis of lightweight symmetric encryption algorithms for internet of things
Link of the paper:
https://www.inderscienceonline.com/doi/abs/10.1504/IJRIS.2016.080072
Authors: Yue Li and Yanqin Cao
Published on: November 1, 2016
Summary: In the study "Performance Evaluation and Analysis of Lightweight Symmetric Encryption Algorithms for the Internet of Things (IoT)," the effectiveness of several symmetric encryption algorithms for IoT devices is assessed and contrasted. AES, DES, and Blowfish are just a few of the algorithms that the authors look at and contrast in terms of computation time, memory requirements, and key size. The objective is to choose the best encryption method for usage in IoT devices with limited resources. The authors come to the conclusion that Blowfish may be a better alternative for applications demanding stronger security, whereas AES is the best option for securing data in IoT devices due to its quick calculation times and small key size. The study sheds light on the trade-offs related to selecting an IoT encryption technique.

4.
Name of the paper: A new lightweight cryptographic algorithm for enhancing data security in cloud computing
Link of the paper:
https://www.sciencedirect.com/science/article/pii/S2666285X21000133
Authors: Fursan Thabita Sharaf Alhomdyb Abdulrazzaq H.A.Al-Ahdalc Prof Dr Sudhir Jagtapa
Published on: June 2021
Summary: In a paper titled "A New Lightweight Cryptographic Technique for Enhancing Data Security in Cloud Computing," a brand-new encryption algorithm is suggested. The method is designed to be quick and effective, making it appropriate for usage in cloud computing systems with limited resources. The technique, according to the inventors, offers a high level of security and benefits from quick encryption and decryption times due to its lightweight design. The algorithm's performance and security are assessed, and the results are contrasted with those of other widely used encryption techniques. The proposed technique is a potentially effective way to improve data security in cloud computing systems, according to the paper's conclusion.

5.
Name of the paper: Towards an End-to-End IoT Data Privacy-Preserving Framework Using Blockchain Technology
Link of the paper: https://link.springer.com/chapter/10.1007/978-3-030-02922-7_5
Authors: Faiza Loukil, Chirine Ghedira-Guegan, Khouloud Boukadi & Aïcha Nabila Benharkat
Published on: 20 October, 2018
Summary: A framework for protecting data privacy in Internet of Things (IoT) systems is presented in the paper "Towards an End-to-End IoT Data Privacy-Preserving Framework Using Blockchain Technology." The authors suggest implementing end-to-end encryption and privacy in IoT networks using blockchain technology. The framework consists of a decentralized network for data transmission and storage that is secure, as well as a technique for key management that makes data encryption and decryption secure. The performance of the suggested framework is assessed by the authors, who also compare it to other widely employed privacy-preserving techniques. The suggested framework offers a high level of security and privacy for data in IoT systems, and the research concludes that it is a potential method for secure communication in such systems.

6.
Name of the paper: Secure Peer-to-Peer communication based on Blockchain.
Link of the paper: https://www.researchgate.net/publication/354058627_Secure_Peer-to-Peer_communication_based_on_Blockchain
Authors: Kahina Khacef, Guy Pujolle
Published on: March 2019
Summary: Authentication plays a vital role in electronic communication. PKI and S/MIME are the most commonly used approaches for authentication. But those encryption protocols are facing threats like MITM and EFAIL attacks. Blockchain overcomes these challenges by allowing decentralized operations with a high level of security. Blockchain uses smart contracts to validate identities and associated public keys. Pretty Good Privacy(PGP) uses PKI services for authentication, which is centralized. Thus results in a single point of failure. Ethereum is a platform for decentralized applications called smart contacts. Ethereum addresses are unique whose ownership can't be changed, their activity can be tracked and analyzed. In this paper, the proposed model describes the application of the decentralization property of blockchain for secure communication. A new key pair will be generated for each user using ECDSA algorithm. Each user's identity will be registered by their public keys. Each time when

users exchange information, smart contacts verifies the identity, timestamp and signature of the transaction. Messages will be encrypted using Elliptic curve Diffie Hellman algorithm by using shared keys. The sequential execution of smart contracts affects the performance of blockchain. This methodology offers confidentiality, authentication, reliability, integrity, transparency, redundancy, fault tolerance. .

7.

Name of the paper: A Decentralized Application for Secure Messaging in a Trustless Environment

Link of the paper: https://ieeexplore.ieee.org/document/8625362

Authors: Mohamed Abdulaziz; Davut Çulha; Ali Yazici

Published on: 24 January 2019

Summary:

The protocol followed in blockchain helps to prevent internal and external cyber attacks by utilizing the decentralization principle. In this article the creation of a decentralized chat application using Ethereum Whisper protocol has been demonstrated. Messages can be sent both securely and anonymously in this application. The communication channel is secured and end-end encrypted and resistant to network attacks. Current electronic chat applications follow SMTP or SMS/GSM protocols which are centralized and were not designed with end-end encryption requirements. These methods do not have anonymity capabilities. Blockchain uses hash tables by mapping usernames to ip addresses without the need of central authority. In this application, both the sender and receiver should generate an asymmetric key pair. The sender transmits the message after encrypting using the sender's public key. The recipient decrypts the message using a private key and verifies using a digital signature. Message numbers are included along with messages to order messages. Upon completion of session symmetric keys will be deleted. In this application users can also subscribe to particular topic related posts, where the identity of each user is anonymous. Elliptic Curve Integrated algorithm has been used for encryption. The messages will not be stored in the blocks after the completion of the session. Messages having smaller size, lower TTL and higher PoW are considered to have higher rating and priority of forwarding.

8.

Name of the paper: DECENTRALISED CHAT APPLICATION

Link of the paper: https://www.irjet.net/archives/V9/i2/IRJET-V9I244.pdf

Authors: Uma Thakur, Abhishek Chichmalkar, Aditya Sambhare, Aman Chaturvedi, Chinmay Khuspare, Nikhil Tembhe

Published on: February 2022
Summary: Communication became a part of everyone's lifestyle. Each person communicates with others through the internet and different chat systems are working on centralized servers. There is a possibility of a single point of failure and data leakage in centralized servers. Decentralization is a way to resolve all these problems. In this system, the information will be distributed and stored across multiple devices. Gun.js, a decentralized datagraph has been used to develop this application. All the computers are connected with each other with equal permissions. The proposed system uses Svelte app with 2 major features, which are email password user authentication and a chat room in which anybody can join. Security Encryption and Authorization(SEA) and Advanced Exchange Equation(AXE) have been used for authentication and connection of peers respectively. Sign in, sign out and avatar creation options have been implemented to better user interaction.

9.
Name of the paper: Secure Communications Using Blockchain Technology
Link of the paper: https://ieeexplore.ieee.org/abstract/document/8599771
Authors: Peter Menegay; Jason Salyers; Griffin College
Published on: 3 January 2019
Summary: In this paper, a secure communication infrastructure using chat, email and MIPR applications has been developed. Proof of Concept has been used to store contents using blockchain databases. Email applications which use SMTP protocols are extended by broadcasting the messages using blockchain client applications such as MS outlook and ThunderBird. Blockchain is used as a backend to provide enhanced security in chat applications. Email and chat applications use BitShares blockchain technology to transfer information. This methodology uses memos in which users can write a short information and transfer. These messages will be automatically encrypted. The proposed chat application has been developed using steem blockchain. Logging into the application is done by providing Steem username and private key. The messages are signed locally using SteamJ Java Library. The proposed MIPR application supports multiple party signatures for transfer of currencies in secured channels.

10.
Name of the paper: Text Encryption in Android Chat Applications using Elliptical Curve Cryptography (ECC)
Link of the paper:
https://www.sciencedirect.com/science/article/pii/S1877050918314650

Authors: DimasNatanael, Faisal, Dewi, Suryani
Published on: 29 August 2018
Summary: Current chat applications are more convenient as there is no message limit, the highest number of people use chat applications, feasible. Due to the increase in use of chat applications, security demands also increased. A proper cryptographic scheme is needed to protect messages. In this paper, Elliptic Curve Cryptography has been proposed for security and protection of messages. Elliptic curve cryptography encrypts the message at sender side using public key and decrypts the message using receiver's private key. The proposed application has several features such as adding friends, end-end encrypted messages etc. The detailed explanation of accuracy of decryption, time complexity etc has been given.

11.
Name of the paper: Server Security in Cloud Computing using Blockchain
Link of the paper: https://ieeexplore.ieee.org/document/9785060
Authors: Asheesh Tiwari, Yasha Aggarwal, Vanshika Aggarwal, Unnati Srivastva
Published on: 25 March 2022
Summary: In this paper, it is suggested that blockchain development and cloud computing development be combined to provide the highest level of security possible in the cloud environment.
By displaying exploratory findings produced by the computer program using the Java programming language, this study verifies the integration of cloud development with blockchain advancement. Integration of blockchain technology with cloud security differs in several ways to increase the result's security and specificity for users and clients.

12.
Name of the paper: A centralized Blockchain based Data Security for Electrical Energy against Attacks
Link of the paper: https://ieeexplore.ieee.ortg/document/9484898
Authors: Shiela David, Dr. Aroul Canessane
Published on:
Summary: The triggers described in this paper may be resolved by keeping crucial data on a decentralized ledger that increases the stability of the electric power system by employing block chain technology to provide high protection to the current power system. The essential operational data is compressed into blocks and hashed to provide a singular value. Information gathering, where the fundamental data is gathered, is where

the paper flow starts. The SHA-256 method is then used to encrypt and decrypt the information. Additionally, there is a technique for data validation that makes use of the Byzantine Fault Tolerance algorithm and enables consensus agreement among all nodes. Additionally, when the blocks have been successfully validated by the miners using a mathematical problem, linking of the blocks takes place to get the nonce, which is nothing more than the answer to adding up the successive blocks. The daily electricity consumption dataset is pre-processed, conditioned, and reviewed before attack categorization occurs.

13.
Name of the paper: A Survey on Blockchain Technology: Evolution, Architecture and Security
Link of the paper: https://ieeexplore.ieee.org/document/9402747
Authors: Muhammad Nasir Mumtaz Bhutta, Amir A khwaja, Adnan Nadeem, Hafiz Farooq, Houbing Song, Yue Cao
Published on: 13 April 2021
Summary: This study has offered a review of pertinent works and commented on their contributions and limits with a critical comparison analysis. The survey included the evolution of the blockchain, frameworks, architectures, security and privacy features. In regard to cryptocurrencies, smart contracts, and other applications, the article offers a viewpoint on how Blockchain architectures work. A few important application and development frameworks serve to illustrate the research advancements in consensus algorithms. Additionally, a thorough discussion of open research topics is conducted, which may assist academics get started on tackling the most difficult problems in the field of blockchain technology.

14.
Name of the paper: Secure Internet Voting using Blockchain Technology
Link of the paper: https://ieeexplore.ieee.org/document/9703027
Authors: Muhammad Ali Khan and Hafiz Shahbaz Rasheed
Published on:
Summary: In order to assure security, this study presents an electronic voting system that uses the Ethereum Blockchain and employs Pallier homomorphic encryption and the private key shifting approach. The system leverages Ethereum Blockchain's smart contracts to keep track of each user and vote. We use HTML and the 256-bit integer-only Solidity language to create the system. Due to the need for significantly larger numbers,

Paillier homomorphic encryption is implemented via an external server. Suggested approach is reliable and suitable for online voting.

15.
Name of the paper: Secure End to End VoIP System Based on Ethereum Blockchain
Link of the paper:
http://ce.sc.edu/cyberinfra/docs/publications/20180817033654165.pdf
Authors: Elie F Kfoury and David J Khoury
Published on: 8 August 2018
Summary: The complexity of SIP security and the problems with third-party trust have been addressed in this research by presenting a unique method for trustless key distribution management based on the Ethereum Blockchain. In a different study effort, they suggested a strategy for improving trust built on the combination of GBA and CA.] The Ethereum Testnet was used to test the implementation. The solution has a good call setup time, according to the results. Additionally, it little affects the entire VoIP architecture.

16.
Name of the paper: On Ends-to-Ends Encryption: Asynchronous Group Messaging with Strong Security Guarantees
Link of the paper: https://dl.acm.org/doi/abs/10.1145/3243734.3243747
Published on: 2018
Summary: With over a billion active users of end-to-end encryption protocols like Signal in the last few years, secure messaging has entered the mainstream. Users of the Signal Protocol can benefit from a powerful feature known as post-compromise security. It turns out, though, that many of its implementations offer a weaker feature for group messaging without giving users any warning: an adversary who manages to corrupt only one member of the group can read and inject messages indefinitely. For the first time, we demonstrate how realistic, asynchronous group messaging systems can accomplish post-compromise security. A group of users can generate a shared symmetric key even if no two are ever online at the same time by using a scheme we call Asynchronous Ratcheting Trees (ART), which leverages tree-based Diffie-Hellman key exchange. ART scalable to groups with tens of thousands of participants while maintaining verifiable security assurances.

17.

Name of the paper: A post-quantum end-to-end encryption over smart contract-based blockchain for defeating man-in-the-middle and interception attacks
Link of the paper:https://link.springer.com/article/10.1007/s12083-020-00901-w
Authors: Amir Hassani Karbasi, Siyamak Shahpasand
Published on: 28 February,2020
Summary: End-to-End Encryption (EEE) and SSL/TLS are both vulnerable to Man-In-The-Middle (MITM) and Compelled Certificate Creation attacks. IP/ARP poisoning and phishing attacks make up a subset of the actual attacks on SSL/TLS and end-to-end encryption. Whether or not the client is connected to a particular authenticated and protected connection, MITM attacks make the client difficult to understand.Due to the immutability and irreversibility of the blockchain, Ethereum smart contracts are used to manage, monitor, and provide detectability and visibility into the history of digital data from its inception to its most recent variant, in a way that is decentralised and accessible from anywhere in the world with high integrity, resiliency, and transparency.

18.
Name of the paper: Improving End-to-End Verifiable Voting Systems with Blockchain Technologies
Link of the paper: https://ieeexplore.ieee.org/abstract/document/8726502
Authors: Anthony J. Perez ,Ebrima N. Ceesay
Published on: 03 June,2019
Summary: The goal of end-to-end verifiable (E2E) voting systems is to provide elections where each voter has direct confirmation that their vote was cast correctly and where anybody can verify that each ballot was accurately counted as part of the final vote total. Sadly, placing the bulletin board on a centralised server exposes this feature to denial of service attacks or data manipulation, both of which could erode the public's confidence in the outcome of the election. This study illustrates the replacement of the bulletin board concept with decentralised storage and a blockchain to reduce these concerns. By doing this, the election audit data benefit from a system that makes the data irreversible, replicated, and available to the general public.

19.
Name of the paper: A secure end-to-end verifiable e-voting system using zero knowledge based blockchain
Link of the paper: https://eprint.iacr.org/2018/466.pdf
Authors: Somnath Panja, Bimal Kumar Roy
Published on: 2018

Summary: In this paper, a cryptographic method for an election that is authenticated, end-to-end verifiable, and secret ballot is provided. Voters should be given confirmation that their ballot was properly cast, recorded, and counted. Even when voters are prepared to be swayed, the voting system as a whole should make it improbable that votes will be coerced. Currently, the counting procedure is carried out by trusted authorities in practically all verifiable electronic voting systems. The DRE-i and DRE-ip systems are an exception. By encrypting ballots so that the election tally may be publicly confirmed without decrypting cast ballots, the DRE-ip method does away with the need for tallying authorities.

20.
Name of the paper: SEEMless: Secure End-to-End Encrypted Messaging with less Trust
Link of the paper: https://dl.acm.org/doi/abs/10.1145/3319535.3363202
Authors: Melissa Chase ,Apoorvaa Deshpande , Esha Ghosh ,Harjasleen Malvai
Published on: 2019
Summary: The security of end-to-end encrypted messaging (E2E) depends on participants' ability to locate the correct recipient's public key. However, users must be able to replace their keys in these systems for them to be useful (for instance, when they lose or reset their devices or reinstall their apps), and we cannot assume any cryptographic method of authenticating the new keys. A compromised or forced service provider can introduce their own keys and carry out a man-in-the-middle attack since in existing E2E systems, the service provider administers the directory of public keys of its registered customers. In this paper, the authors define the idea of a Privacy-Preserving Verifiable Key Directory by building on the strategy of CONIKS (Melara et al., USENIX Security '15). (VKD)

21.
Name of the paper: More is Less: On the End-to-End Security of Group Chats in Signal, WhatsApp, and Threema
Link of the paper: https://ieeexplore.ieee.org/abstract/document/8406614
Authors: Paul Rösler ; Christian Mainka ; Jörg Schwenk
Published: 2018
Summary: Both one-to-one communication and group communication use secure instant messaging. Little is known about the cryptographic techniques and security assurances of secure group communication via instant messaging, despite the fact that the first variation has recently drawn a lot of interest. We first offer an extensive and realistic security model to frame how we will approach an evaluation of group instant messaging

technologies. To capture pertinent characteristics for communication in dynamic groups, this model incorporates security and reliability objectives from other related literature. Thus, the definitions take into account their suitability for quick message transmission. They examine Signal, WhatsApp, and Threema, three commonly used real-world protocols, to demonstrate its applicability.Our systematic research demonstrates that (1) end-to-end protection is not provided for the integrity of the communications, which is represented by the integrity of all exchanged messages, and (2) the closeness of the groups, which is represented by the members' capacity to manage the group.

22.
Name of the paper: SINGLETON: A lightweight and secure end-to-end encryption protocol for the sensor networks in the Internet of Things based on cryptographic ratchets
Link of the paper: https://link.springer.com/article/10.1007/s11227-020-03411-x
Authors: Amir Hassani Karbasi & Siyamak Shahpasand
Published: 2020
Summary: Here, several threats are defined along with protocols that offer secure communication for a range of IoT applications are discussed. In this work, they provide a thorough and portable security methodology based on Cryptographic Ratchets to safeguard IoT devices and sensors. In other words, the Double Ratchet Algorithm-based encrypted messaging protocol Singleton is defined, and its implementation is tested and compared to the IoT standard protocols and a postquantum version of the protocol. Additionally, several cryptographic primitives are analysed and tested to see whether they can be used in the protocol. The findings demonstrate that the protocol serves as a foundation for enhanced and scalable IoT sensors in addition to efficient resource-wise protocols and architectures.Our design and analysis show that the Singleton security architecture is simple to integrate into current network protocols like IEEE 802.15.4 or OMA LWM2M, which has significant advantages over existing methods that cannot provide both performance and crucial security services. An end-to-end encryption, forward secrecy, backward secrecy, authentication, and deniability are all provided by a cryptographic ratchet-based protocol for chat programmes like WhatsApp, Skype, Facebook Private Messenger, Google Allo, and Signal.

23.
Name of the paper: How Secure is TextSecure?
Link of the paper: https://ieeexplore.ieee.org/abstract/document/7467371

Authors: Tilman Frosch; Christian Mainka; Christoph Bader; Florian Bergsma; Jörg Schwenk;Thorsten Holz
Published: 2016
Summary: TextSecure is a texting programme that has gained a lot of attention and makes the claim to offer safe instant chat. In addition to several direct instals, Cyanogen-Mod, the most well-liked aftermarket firmware for Android, includes its protocol. The text messaging protocol is still used by TextSecure's replacement, Signal. The sophisticated cryptographic protocol used by TextSecure is described in detail for the first time in this work. Its three major components—key exchange, key derivation, and authenticated encryption—are also examined in terms of their security, and TextSecure's primary security claims are covered. Additionally, they explicitly demonstrate that TextSecure's push messaging can actually accomplish the majority of the stated security requirements assuming key registration is believed to be safe.

24.
Name of the paper: Enabling (End-to-End) Encrypted Cloud Emails With Practical Forward Secrecy
Link of the paper: https://ieeexplore.ieee.org/abstract/document/9340272
Authors: Jianghong Wei; Xiaofeng Chen; Jianfeng Wang; Xuexian Hu; Jianfeng Ma
Published: 2022
Summary: They introduce a new cryptographic primitive called forward-secure puncturable identity-based encryption (fs-PIBE), which enables an email user to perform fine-grained revocation of decryption capacity, in order to capture forward secrecy of encrypted cloud email systems without sacrificing the practicability. They develop a framework for encrypted cloud email systems based on such a basic and implement it using a concrete fs-PIBE design with constant ciphertext size and standard model-provable security. Additionally, they enhance the suggested fs-PIBE scheme to allow end-to-end encryption and outsourced decryption, improving the security and effectiveness of the given system.

25.
Name of the paper: Improving Non-Experts' Understanding of End-to-End Encryption: An Exploratory Study
Link of the paper:https://ieeexplore.ieee.org/abstract/document/9229664
Authors: Wei Bai, Michael Pearson, Patrick Gage Kelley, and Michelle L. Mazurek
Published: 2020

Summary: End-to-end encryption (E2EE), which has been used by a number of well-known messaging applications, may efficiently secure the privacy of online communication. However, past research suggests that many users struggle, in part because of erroneous mental models, to use E2EE products effectively and confidently as well as to recognise their security benefits. In order to give non-experts high-level, roughly accurate information regarding end-to-end encryption, this document is a first attempt. Participants in a lab research were questioned about their knowledge of E2EE before and after creating a lesson. They were also asked which material they found most helpful and unexpected. Participants' comprehension of the advantages and restrictions of E2EE has generally improved. They discovered that sharing information on confidentiality, dangers, and vulnerabilities was the most beneficial, unexpected, and persuasive.

LITERATURE SURVEY - BASE PAPER

S.NO	Name of the transaction/journal/conference with year	Major technologies used	Results/Outcome of their research	Drawbacks if any
1.	Chatterjee, Runa et al. “Design of Cryptographic model for End-to-End Encryption in FPGA based systems.” *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)* (2019): 459-465.	Symmetric Key Cryptography - This is used to secure the encryption of data transmission over communication channels. Common symmetric key algorithms used in FPGA-based	In this study a cryptographic model for end-to-end encryption in FPGA systems is designed. To attain a high level of security and performance, the authors	Complexity - Cryptographic models for end-to-end encryption in FPGA-based systems can be complex to design and implement, especially when integrating multiple

		systems are AES and DES. Public Key Cryptography - This is used for secure communication over insecure channels. Common public key algorithms used in FPGA-based systems are RSA, Elliptic Curve Cryptography (ECC), and Diffie-Hellman. Hardware Security Modules (HSMs) - These are specialized cryptographic processors that provide secure key storage, encryption and decryption capabilities. Field-Programmable Gate Arrays (FPGAs) - These are programmable	suggest using FPGAs to construct encryption and decryption procedures. A performance evaluation of the algorithms on FPGA-based systems is part of the design, which focuses on the use of symmetric encryption techniques like AES and DES. According to the evaluation's findings, a high level of security is maintained while the FPGA-based encryption model offers quick encryption and decryption times. For designers and developers of FPGA-based	cryptographic algorithms and technologies.

		integrated circuits that can be reconfigured to perform specific functions, such as encryption and decryption, efficiently. Trusted Platform Module (TPM) - This is a specialized microcontroller that provides secure storage of cryptographic keys and performs cryptographic operations such as encryption or decryption.	systems, the paper offers insights into the design and implementation of encryption in these systems.	
2.	U. P. Ellewala, W. D. H. U. Amarasena, H. V. S. Lakmali, L. M. K. Senanayaka and A. N. Senarathne, "Secure Messaging Platform Based on Blockchain," 2020 2nd International Conference on Advancements in Computing (ICAC), Malabe, Sri Lanka, 2020, pp. 317-322, doi: 10.1109/ICAC51239.2020	Smart contracts- These inherit the underlying blockchain properties and make the records immutable. Smart contracts mitigates single point of failure. Encryption -	In this study, the proposed model uses a more secure channel of communication along with advanced auditing features. Proposed model consists	Messages cannot be modified once transferred. Time Complexity because of implementing many algorithms at each level. Requires more storage and is not as scalable

	.9357306.	AES and RSA algorithms have been used for end to end encryption to provide confidentiality. These algorithms generate public and private keys for encryption and description. These algorithms also generate digital signatures to verify the identity of the sender. Authentication- Multi factor authentication by generating OTP using timestamp values, HMAC-SHA-256 algorithm- Used to generate OTP by creating HOTP value. Input is key and timestamp. Output is of 160 bits which will be truncated later	of a message authentication model, end to end encryption to protect the privacy of the user, cryptographic hash function to verify the integrity of messages and smart contracts. Each block is time stamped and arranged in chronological order. The proposed smart contract can be changed according to organizations policies and procedures. This model consists of 3 authentication stages to provide more user confidentiality. First stage is generating OTP using timestamping	as centralized applications.

		for the use. Ping pong stream cipher algorithm- To generate secure OTP based on bit separation which changes each and every time a user logins. Timeout mechanism- To prevent DoS attacks, the application uses a timeout mechanism after 15 minutes of inactive user login. Data loss prevention model- Implements DLP model using machine learning algorithm to prevent leakage of data. Consensus Algorithm- Used to validate each block of	based on the stream cipher algorithm. Second stage is to sign up using a 6 digit user's passcode. Third stage is using captcha. To protect from DoS and brute force attacks, a timeout mechanism has been implemented. Consensus algorithm in blockchain validates each block and maintains data consistency in a decentralized network. This paper has provided a roadmap for designing a secure chat application using blockchain.	

		transactions and form a global ledger.		
3.	T. Sureshkumar and M. Vijayakumar, "Blockchain based Security System for the Devices in Internet of Things," 2020 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 2020, pp. 1-4, doi: 10.1109/ICCCI48352.2020.9104092.	In an IoT system, the network may be divided into four levels: the physical layer, the application layer, the database layer, and the communication layer. On each of these layers, security rules are implemented. Devices like sensors and actuators are found on the physical layer and are employed in the physical environment of an IoT system. Because of their weak access control and encryption, these devices are vulnerable to assaults. Data transfer between the	A high level of security in terms of confidentiality, integrity, dependability, and other aspects is required by the IoT's development. The security measures used for typical networks cannot be implemented due to hardware limitations. For these reasons, a distributed record-based technology known as the blockchain (BC) approach is employed instead of a centralised one. When compared to the current security methods, this delivers superior	This technique involves modification of IoT devices and to do that we need to replace all the device which are currently in the market. This process will consume a lot of time and might be a costly solution than available options.

		physical layer and the top layer is handled by the communication layer. Blockchain protocols, which offer security and privacy principles for data transfer, are applied to this layer. The database layer is the most crucial layer, and it requires the right security standards. A decentralised database is maintained using records that include a timestamp, a cryptographic hash function, and other information.	outcomes in terms of security considerations like security, privacy, etc In order to enable protected data communication for IoT networks, this research suggests a blockchain-based security framework. The fact that the blockchain is resistant to numerous security risks is one of its main advantages. It offers a lot of security characteristics, including enhanced consistency, enhanced fault tolerance, high efficiency, and enhanced scalability. . As a result, the	

			integration of blockchain technology with IoT network devices will produce a broad platform where all the devices will be able to safely exchange data in a dispersed network.	
4.	R. Singh, A. N. S. Chauhan and H. Tewari, "Blockchain-enabled End-to-End Encryption for Instant Messaging Applications," 2022 IEEE 23rd International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), Belfast, United Kingdom, 2022, pp. 501-506, doi: 10.1109/WoWMoM54355.2022.00078.	The Ethereum blockchain is implemented on the Docker platform to implement the blockchain capabilities. Our investigation looks at how well AES256 in CBC mode performs for encryption and decryption, HMAC with SHA256 performs for message verification codes, and how long the E2EE process takes overall on Android emulators. The	The suggested system can be considered a true end-to-end encryption technique because it does not rely on an IM server for encryption and decryption. currently improvements can be done on two further crucial components of the suggested framework: the first is to provide secure "group communications," and the second is to evaluate the framework's	

		length of the input strings has been used to estimate how long certain application procedures will take. The trials were carried out utilising the Google Pixel 4 XL emulator and the 1536 MB RAM and 384 MB heap size Intel Atom(x86) system image.	"scalability" in a blockchain-based setting. For key exchange between users, the group messaging protocol uses one-to-one encrypted channels.	
5.	Cohn-Gordon, K., Cremers, C., Dowling, B. *et al.* A Formal Security Analysis of the Signal Messaging Protocol. *J Cryptol* 33, 1914–1983 (2020). https://doi.org/10.1007/s00145-020-09360-1	“post-compromise security”-In the event that the key for current communications is compromised, post-compromise security protects upcoming messages. By utilising the key update procedure, it fixes itself. Key update method uses asymetric	Non-Signal library components- The open-source libraries contain various sections of code which are not considered part of the Signal protocol. Out-of-band key verification- Signal supports a fingerprint mechanism for verifying	The investigation demonstrates that Signal's cryptographic core has valuable security features. These complicated but inherent features are represented in our security model, and they demonstrate that Signal meets them under common

		key exchange, such as DHE, and chain of key derivation function to generate new key for next messages. “triple Diffie–Hellman (X3DH)”X3DH establishes a shared secret key between two parties who mutually authenticate each other based on public keys. X3DH provides forward secrecy and cryptographic deniability. “Double Ratchet ” The Double Ratchet algorithm is designed to provide security against an attacker who records encrypted messages and then compromises the sender or receiver at a	public keys through an out-of-band channel. They simply assume that long-term and medium-term public key distribution is honest. They do not analyse the out- of-band verification channel. Out-of-order decryption-To decrypt out-of-order messages, users must store message keys until the messages arrive, reducing their forward security. Simultaneous session initiation	cryptographic presumptions. Even in a variety of hostile compromise situations, such as forward security, they assume secrecy and authentication of the message keys that Signal derives. Forward secrecy has several disadvantages over post-compromise security, which Signal may be able to accomplish when implemented properly.

		later time.		

Overall Technology Gap

Technology gap that can be observed is the focus on practicality and usability of security solutions. Some papers focus on developing encryption algorithms and systems that are lightweight, not so efficient, and can be implemented in real-world scenarios. While our project aims to design user-friendly and decentralized applications that can provide secure communication and data exchange, our approach integrates blockchain and encryption techniques to address security and privacy concerns in instant messaging applications. This represents a step forward in the development of blockchain-based secure communication systems.

IV. SYSTEM DESIGN

a. Architecture Diagram

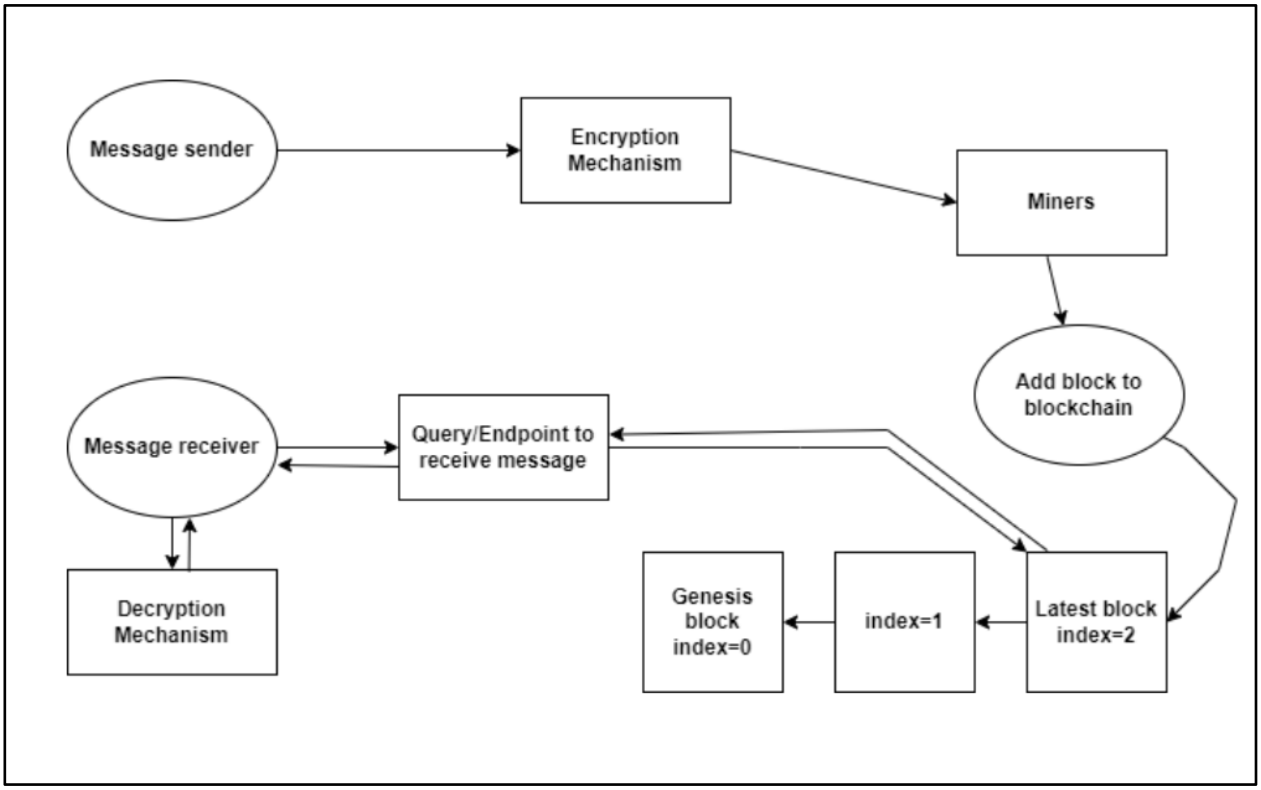


Fig 3: Architecture diagram

b. Flow Diagram

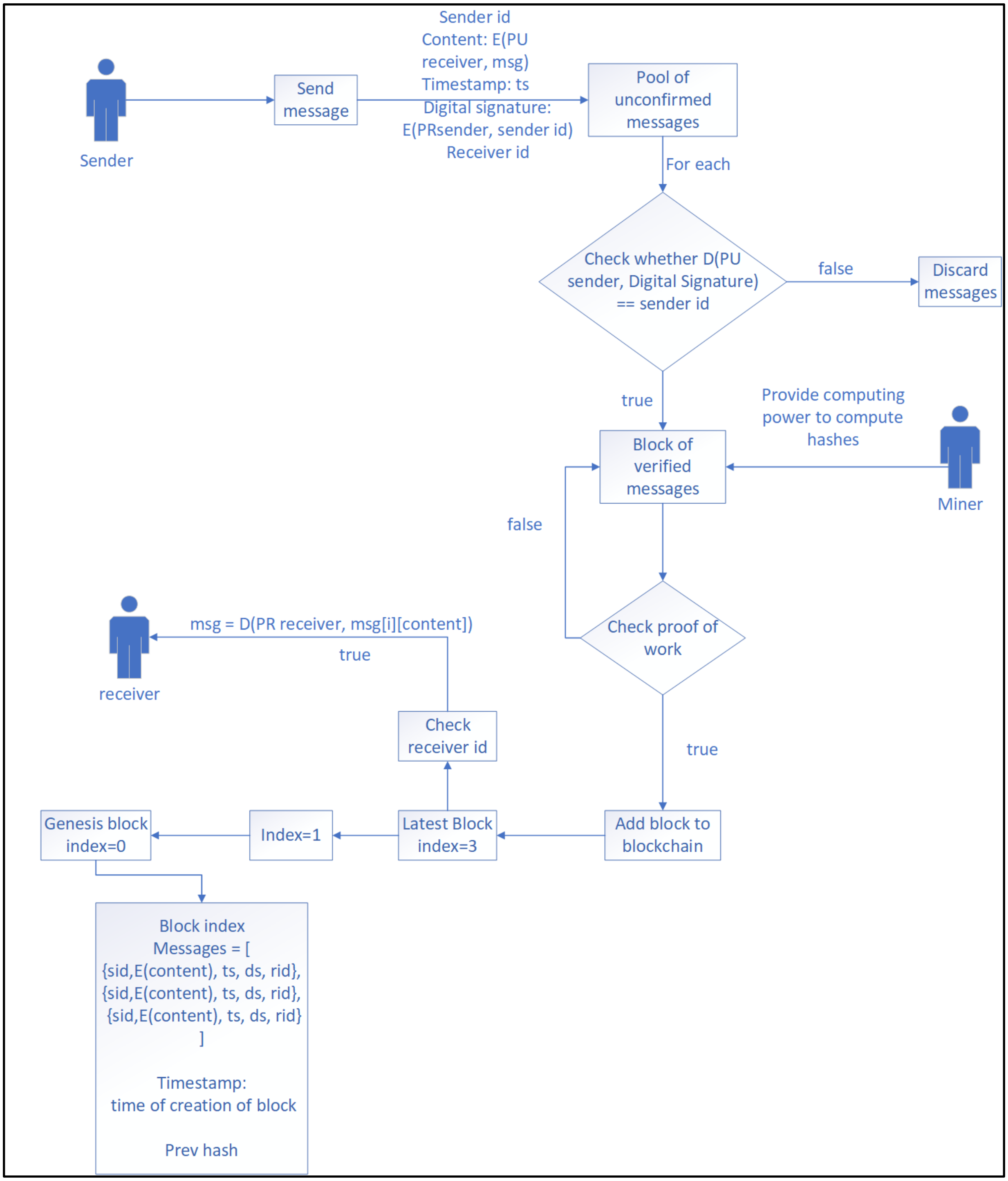

Sender
Send message
Sender id
Content: E(PU receiver, msg)
Timestamp: ts
Digital signature: E(PRsender, sender id)
Receiver id
Pool of unconfirmed messages
For each
Check whether D(PU sender, Digital Signature) == sender id
false
Discard messages
true
Provide computing power to compute hashes
Miner
Block of verified messages
false
Check proof of work
true
msg = D(PR receiver, msg[i][content])
true
receiver
Check receiver id
Genesis block index=0
Index=1
Latest Block index=3
Add block to blockchain
Block index
Messages = [
{sid,E(content), ts, ds, rid},
{sid,E(content), ts, ds, rid},
{sid,E(content), ts, ds, rid}
]
Timestamp:
time of creation of block
Prev hash

Fig 4: Flow diagram

c. Sequence Diagram

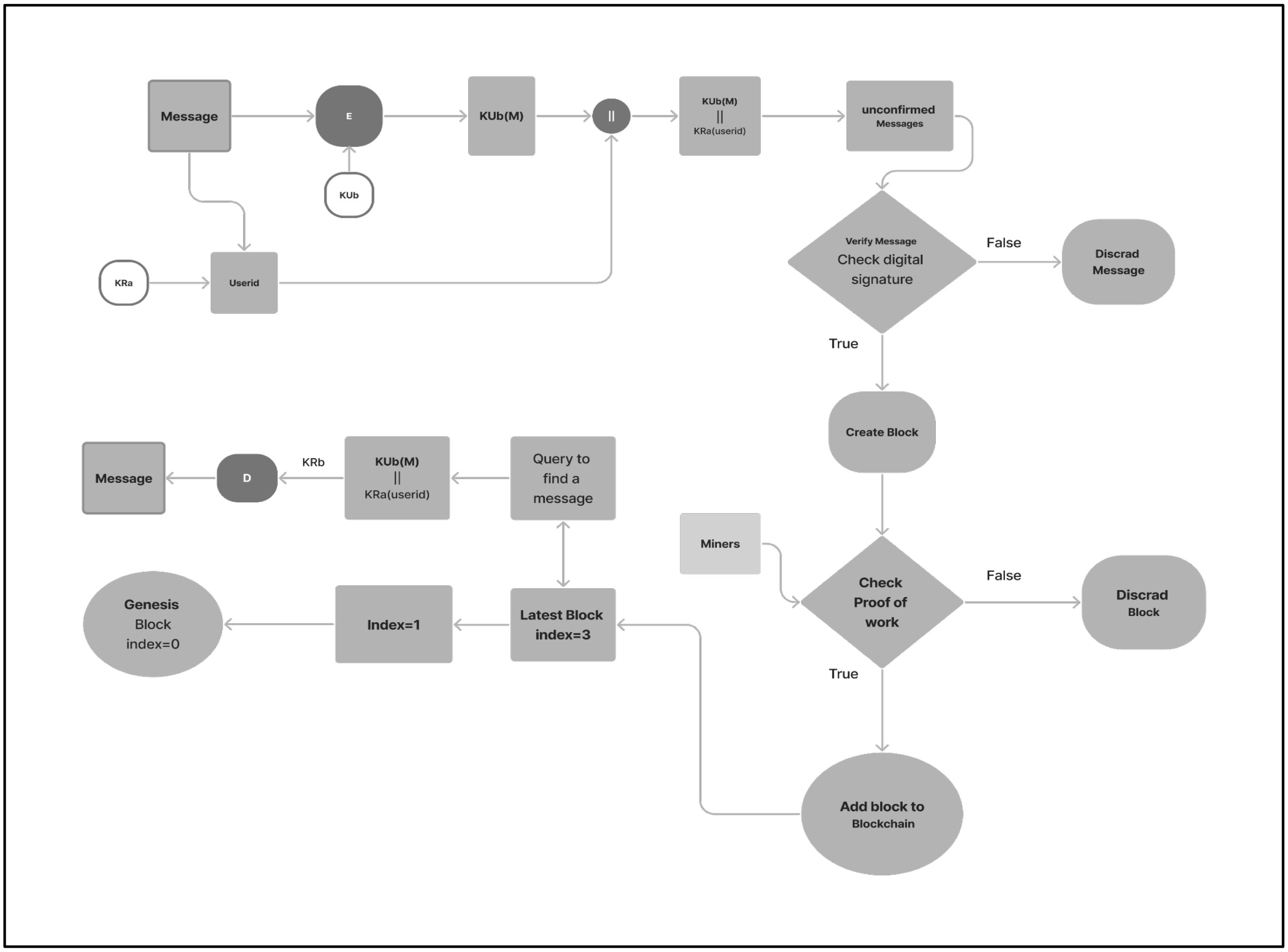


Fig 5: Sequence diagram

d. Algorithm & techniques

Life Cycle of a Message

1. Messages will be encrypted by the private key of the receiver.
2. Digital signature will be prepared by encrypting user identifier with the private key of the sender.
3. The merged output will be added to a set of unconfirmed messages.
4. Among these messages the ones with the correct digital signature will be considered for the block creation. Here the digital signature verification process takes place.
5. The created block will be verified by proof of work algorithm through miners.

6. If the block hash is strong enough and verified by proof of work algorithm then it will be added to the chain.
7. Now a receiver can query his/her messages from the blockchain.
8. The encrypted message is decrypted with the private key of the receiver.

Encryption with RSA

To achieve confidentiality, we have used RSA public key cryptography to encrypt and decrypt the messages kept inside the blockchain. Obtaining authentication also involves the use of digital fingerprints. Due to the blockchain's architecture, which consists of sequential blocks with previous hashes, immutability has already been accomplished.

Storing transactions into blocks

We are storing data in our blockchain in a format that's widely used: JSON. Example of a message:

```
msg= {
'author': author,
'content': E(KUb, msg),
'receiver': receiver,
'digital_signature': E(KRa, author_id), 'pub_key_author': (e, n),
'timestamp': ts,
}
```

Blocks are used to organize messages. There could be one word or many messages in a block. The blocks containing the communications are consistently created and updated on the blockchain. There could be a lot of pieces, so each has its own unique id.

Adding digital fingerprints to the blocks

The first stage in preventing any tampering with the data stored inside the block is detection. If the data in the block has been altered, we can tell using cryptographic hash methods.

A hash function is a mathematical formula that takes in data of any size and returns data of a specific size, known as a hash, which is frequently used to identify the input.

The qualities of a perfect hash function are as follows:

1. From a computational perspective, it ought to be straightforward to calculate.
2. It should always generate the same hash when given the same data, or it should be deterministic.
3. Should always be random, meaning that even a single bit shift in the data should significantly alter the hash.

This has the following effects:

1. Given the hash, it is impossible to predict the input data (the only way is to try all the possible input combinations).
2. To check the given hash, you can just pass the input through the hash function if you know both the input and the provided hash.

Blockchain uses this sort of effort imbalance between determining the hash from an input (easy) and determining the input from a hash (almost impossible) to achieve the desired properties.

Implementation of Proof of work algorithm

If we change the block before it, we can quickly recompute the hashes of all the blocks after it and create a new, valid blockchain. We use the asymmetry in hash function attempts to make the task of computing the hash challenging and unpredictable in order to prevent this. We put some constraints on the block instead of allowing any hash. We additionally stipulate that our hash must contain "n preceding zeros," where "n" can be any positive integer.

Since we don't want to alter the current data, we know that the hash won't change until we update the block's data. We add some fictitious data that can be changed. We expand our block by including the nonce entry. Until we discover a hash that satisfies our limitation, a nonce is a number that will be altered. The fact that a nonce meets the restriction shows that calculations were made. The constraint's number of zeros determines how "difficult" our Proof of Work technique is (more the number of zeroes, harder it is to figure out the nonce).

Additionally, because of the asymmetry, Proof of Work is challenging to calculate but simple to validate once we know the nonce (you just need to run the hash function again):

Establish consensus and decentralization

We develop a method to enable a new node to learn about other peers in the network in order to move from a single node to a peer-to-peer network.

To register with existing nodes in the network, a new node can use the register with existing node function (via the /register with endpoint).

The following will benefit from this:

1. Request that a new peer be added to the remote node's inventory of recognised peers.

2. Initialize the distant node's blockchain with that of the new node's.

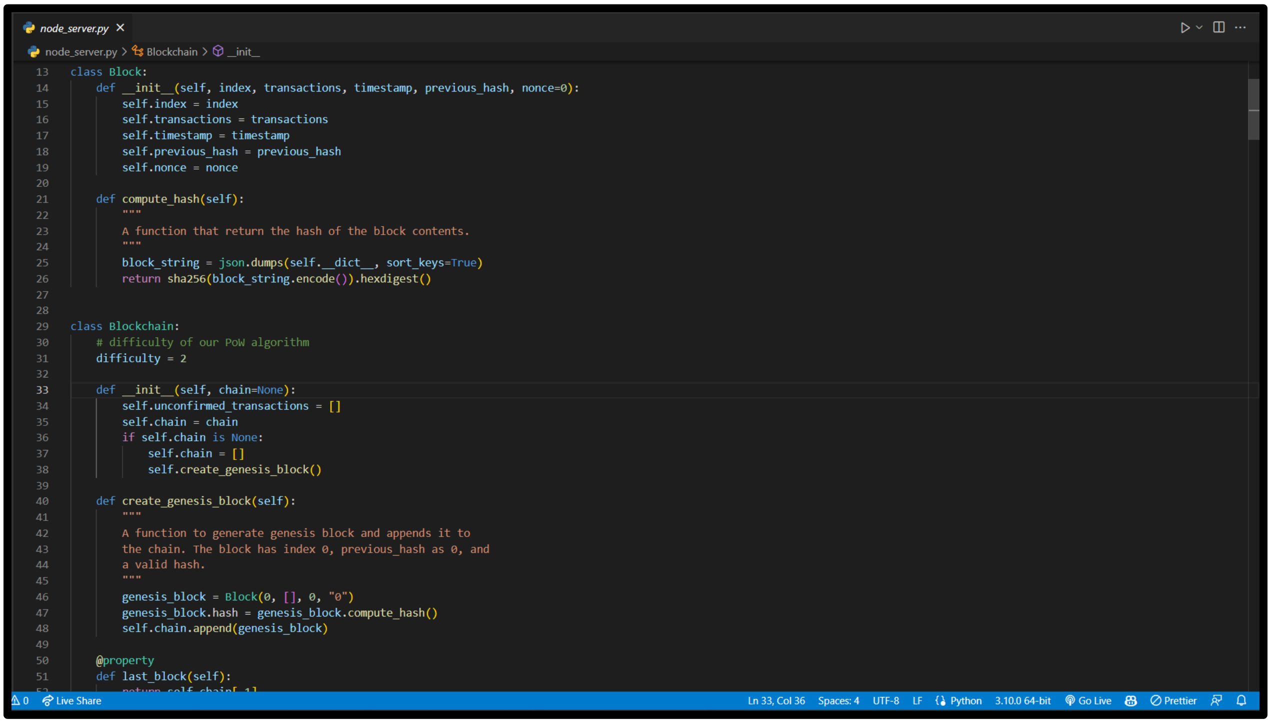
3. In case the node moves off-grid, resynchronize the blockchain with the network.

Multiple nodes do pose a challenge, though. The copy of chains of a few nodes can vary due to deliberate manipulation or unintentional causes (like network latency). To keep the integrity of the entire system in that situation, nodes must agree on some version of the chain. To put it another way, we must reach "agreement."

When the chains of various participating nodes in the network appear to diverge, a straightforward consensus algorithm could be to concur on the longest valid chain. This strategy is justified by the fact that the longest chain provides a reasonable approximation of the amount of work that has been completed (remember that it is challenging to calculate proof of work).

V. IMPLEMENTATION & GOAL ACHIEVED

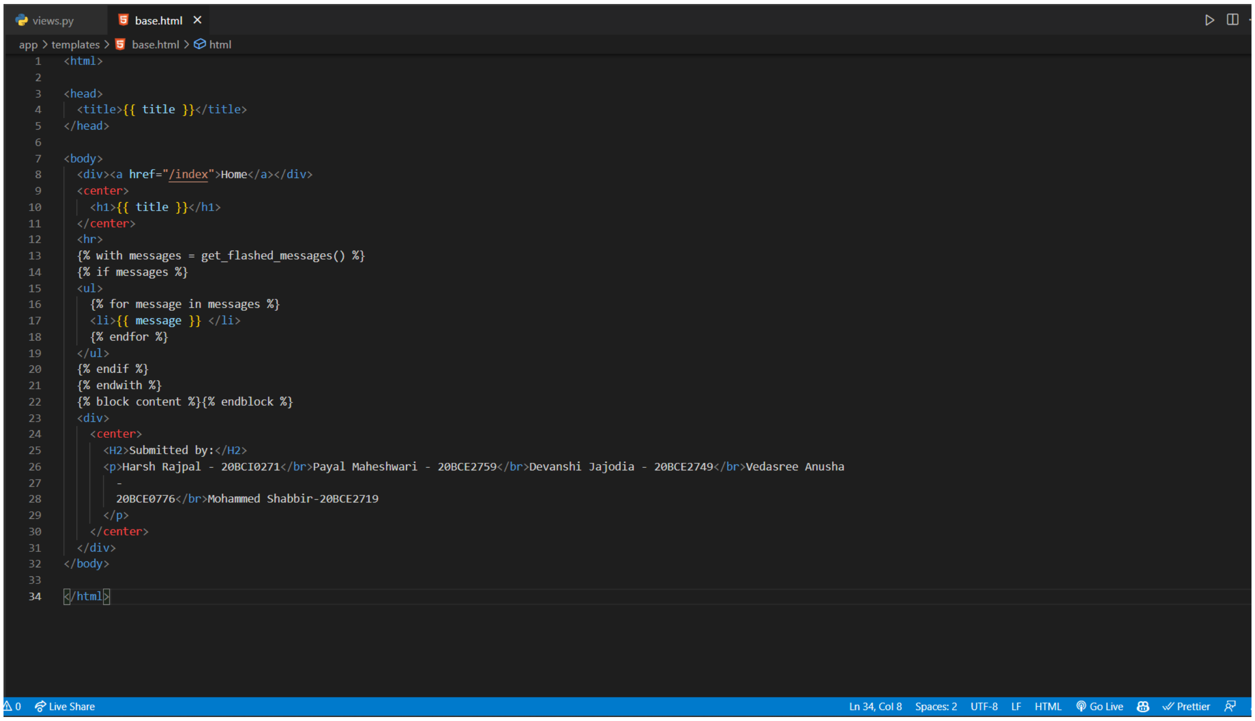
Backend server:



```python
class Block:
    def __init__(self, index, transactions, timestamp, previous_hash, nonce=0):
        self.index = index
        self.transactions = transactions
        self.timestamp = timestamp
        self.previous_hash = previous_hash
        self.nonce = nonce

    def compute_hash(self):
        """
        A function that return the hash of the block contents.
        """
        block_string = json.dumps(self.__dict__, sort_keys=True)
        return sha256(block_string.encode()).hexdigest()


class Blockchain:
    # difficulty of our PoW algorithm
    difficulty = 2

    def __init__(self, chain=None):
        self.unconfirmed_transactions = []
        self.chain = chain
        if self.chain is None:
            self.chain = []
            self.create_genesis_block()

    def create_genesis_block(self):
        """
        A function to generate genesis block and appends it to
        the chain. The block has index 0, previous_hash as 0, and
        a valid hash.
        """
        genesis_block = Block(0, [], 0, "0")
        genesis_block.hash = genesis_block.compute_hash()
        self.chain.append(genesis_block)

    @property
    def last_block(self):
        return self.chain[-1]
```

Fig 6: Sample code of Backend server

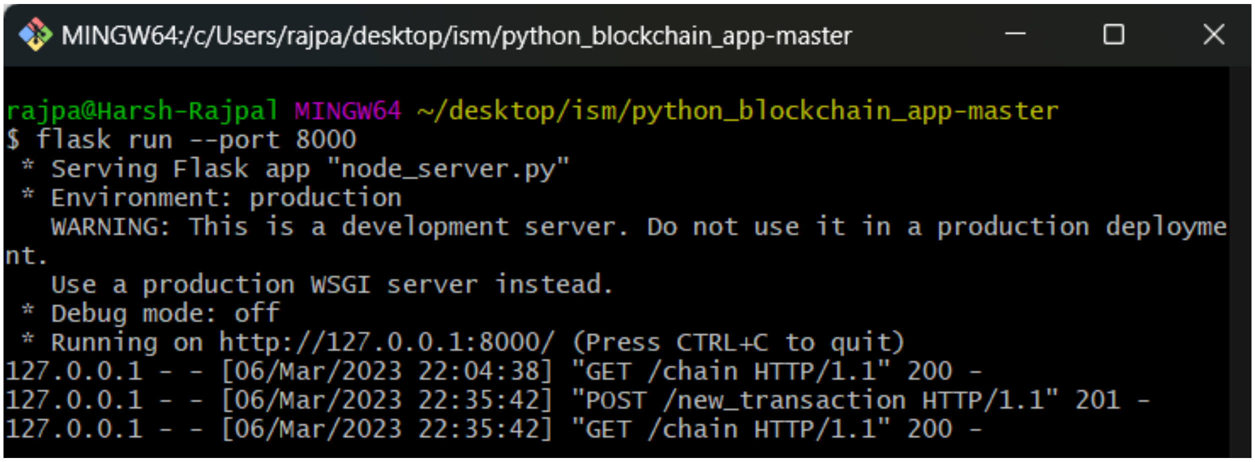
Homepage:



```html
<html>

<head>
  <title>{{ title }}</title>
</head>

<body>
  <div><a href="/index">Home</a></div>
  <center>
    <h1>{{ title }}</h1>
  </center>
  <hr>
  {% with messages = get_flashed_messages() %}
  {% if messages %}
  <ul>
    {% for message in messages %}
    <li>{{ message }} </li>
    {% endfor %}
  </ul>
  {% endif %}
  {% endwith %}
  {% block content %}{% endblock %}
  <div>
    <center>
      <H2>Submitted by:</H2>
      <p>Harsh Rajpal - 20BCI0271</br>Payal Maheshwari - 20BCE2759</br>Devanshi Jajodia - 20BCE2749</br>Vedasree Anusha
        -
        20BCE0776</br>Mohammed Shabbir-20BCE2719
      </p>
    </center>
  </div>
</body>

</html>
```

Fig 7: Sample code for homepage

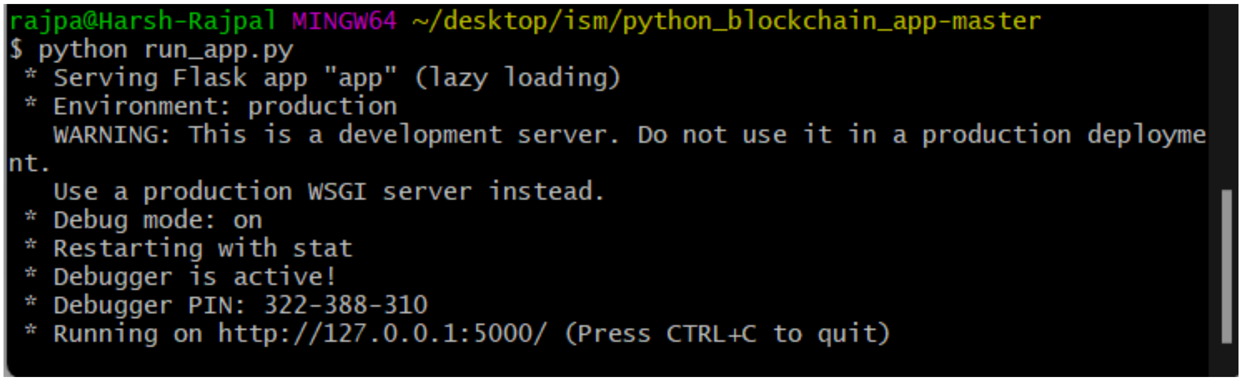
Starting backend server:


MINGW64:/c/Users/rajpa/desktop/ism/python_blockchain_app-master

```
rajpa@Harsh-Rajpal MINGW64 ~/desktop/ism/python_blockchain_app-master
$ flask run --port 8000
 * Serving Flask app "node_server.py"
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployme
nt.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:8000/ (Press CTRL+C to quit)
127.0.0.1 - - [06/Mar/2023 22:04:38] "GET /chain HTTP/1.1" 200 -
127.0.0.1 - - [06/Mar/2023 22:35:42] "POST /new_transaction HTTP/1.1" 201 -
127.0.0.1 - - [06/Mar/2023 22:35:42] "GET /chain HTTP/1.1" 200 -
```

Fig 8: Starting backend server

Running development server(Front End):

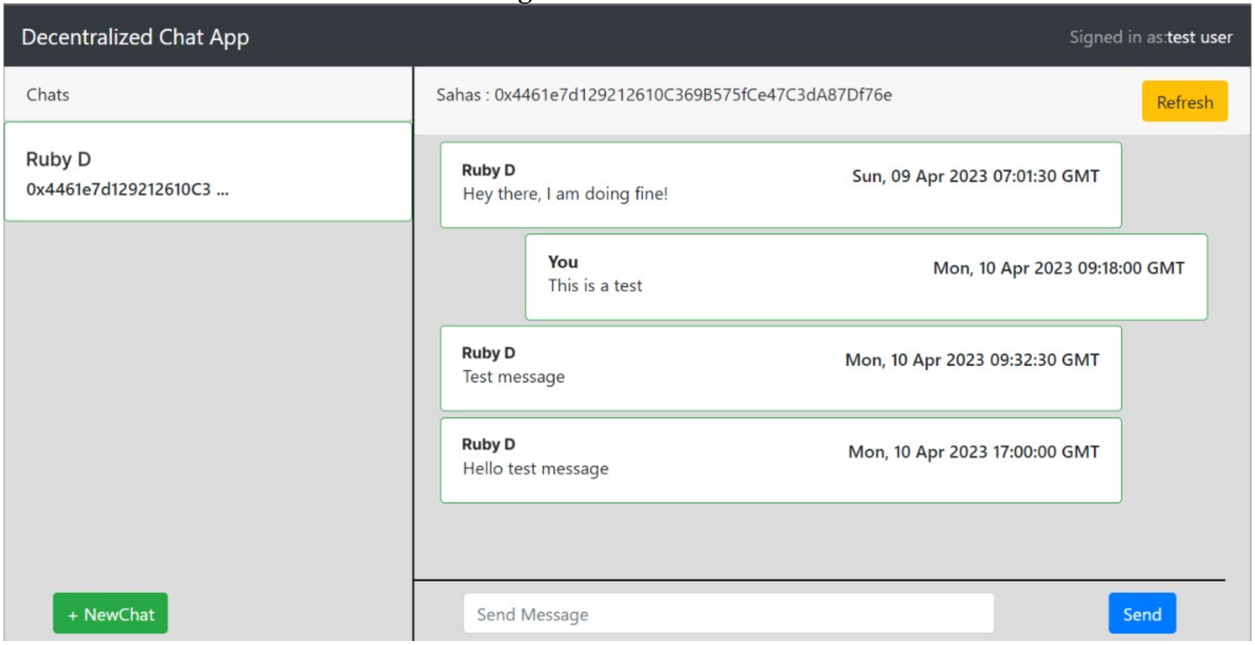


```
rajpa@Harsh-Rajpal MINGW64 ~/desktop/ism/python_blockchain_app-master
$ python run_app.py
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployme
nt.
   Use a production WSGI server instead.
 * Debug mode: on
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 322-388-310
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Fig 9: Running development server

Final Output:

Login first as Test user



Ruby D
0x4461e7d129212610C3 ...

Sahas : 0x4461e7d129212610C369B575fCe47C3dA87Df76e

Ruby D Sun, 09 Apr 2023 07:01:30 GMT
Hey there, I am doing fine!

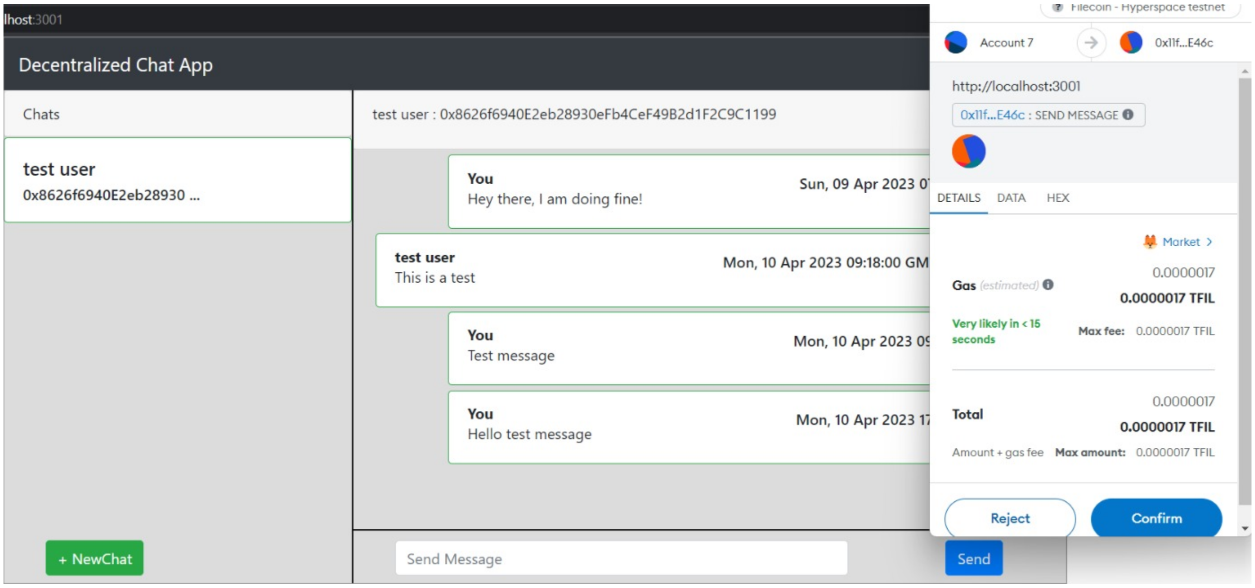
You Mon, 10 Apr 2023 09:18:00 GMT
This is a test

Ruby D Mon, 10 Apr 2023 09:32:30 GMT
Test message

Ruby D Mon, 10 Apr 2023 17:00:00 GMT
Hello test message

Fig 10: Login as test user

Login second as Ruby D



test user
0x8626f6940E2eb28930 ...

test user : 0x8626f6940E2eb28930eFb4CeF49B2d1F2C9C1199

You Sun, 09 Apr 2023 07
Hey there, I am doing fine!

test user Mon, 10 Apr 2023 09:18:00 GM
This is a test

You Mon, 10 Apr 2023 09
Test message

You Mon, 10 Apr 2023 17
Hello test message

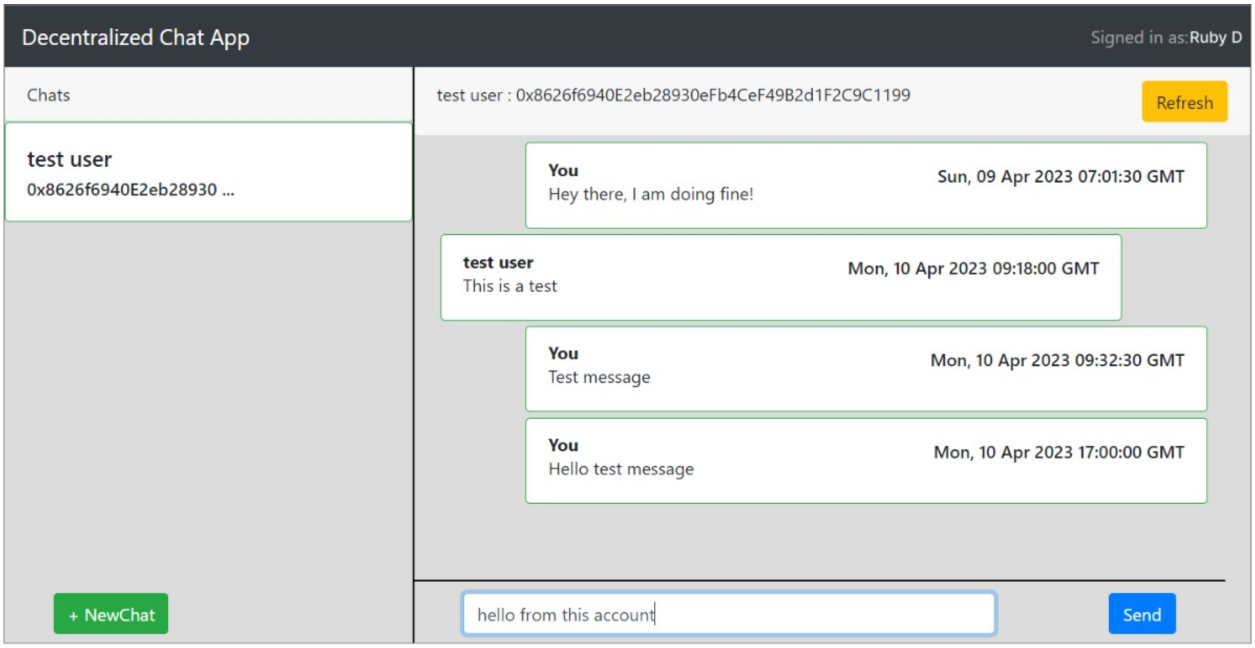
http://localhost:3001
0x11f...E46c : SEND MESSAGE

Gas *(estimated)* 0.0000017
0.0000017 TFIL
Very likely in < 15 seconds **Max fee:** 0.0000017 TFIL

Total 0.0000017
0.0000017 TFIL
Amount + gas fee **Max amount:** 0.0000017 TFIL

Fig 11: Login as a user

Send a message from Ruby D



Signed in as: **Ruby D**

test user
0x8626f6940E2eb28930 ...

test user : 0x8626f6940E2eb28930eFb4CeF49B2d1F2C9C1199

You Sun, 09 Apr 2023 07:01:30 GMT
Hey there, I am doing fine!

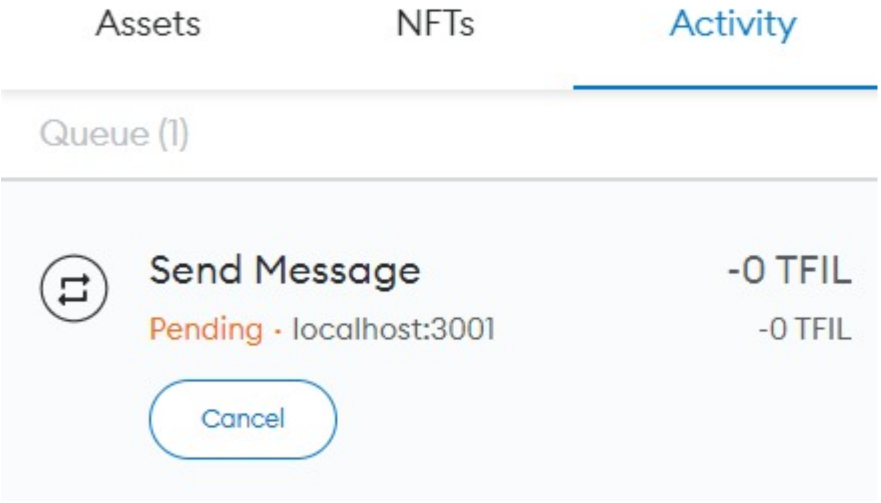
test user Mon, 10 Apr 2023 09:18:00 GMT
This is a test

You Mon, 10 Apr 2023 09:32:30 GMT
Test message

You Mon, 10 Apr 2023 17:00:00 GMT
Hello test message

hello from this account

Fig 12: Send message from user

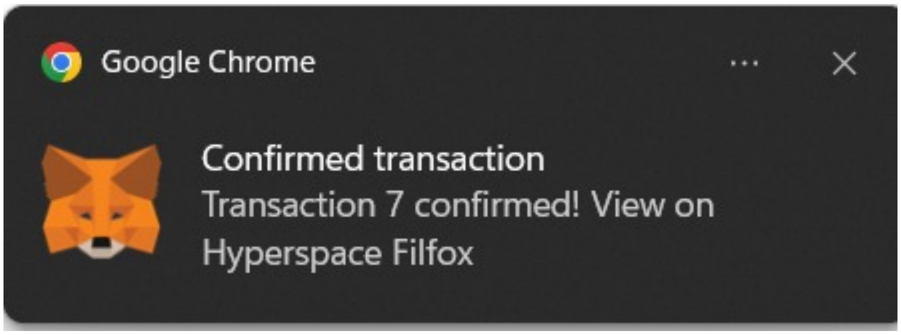


Send Message -0 TFIL

Pending · localhost:3001 -0 TFIL

Fig 13: Message added to pending pool

User gets a notification when the message received

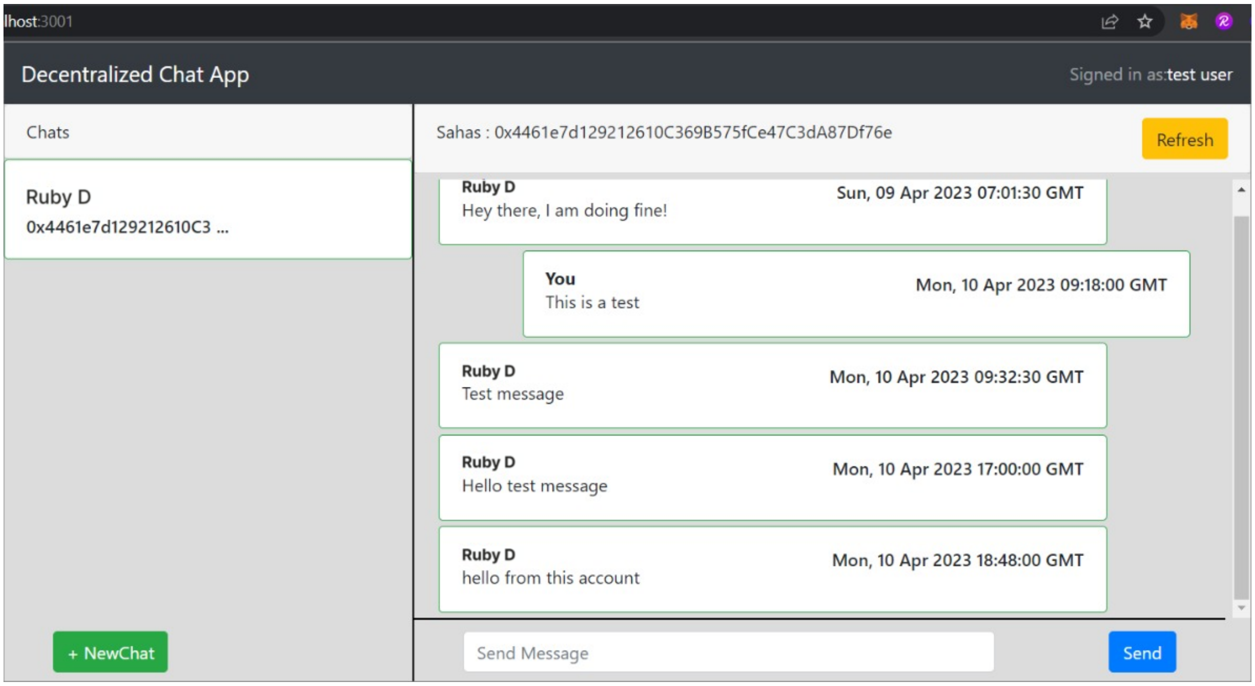


Confirmed transaction

Transaction 7 confirmed! View on Hyperspace Filfox

Fig 14: User gets notification when message is received

Sign in again as test user to receive and view the message



Ruby D
0x4461e7d129212610C3 ...

Sahas : 0x4461e7d129212610C369B575fCe47C3dA87Df76e

Ruby D Sun, 09 Apr 2023 07:01:30 GMT
Hey there, I am doing fine!

You Mon, 10 Apr 2023 09:18:00 GMT
This is a test

Ruby D Mon, 10 Apr 2023 09:32:30 GMT
Test message

Ruby D Mon, 10 Apr 2023 17:00:00 GMT
Hello test message

Ruby D Mon, 10 Apr 2023 18:48:00 GMT
hello from this account

Fig 15:View the received message

VI. RESULT AND DISCUSSION

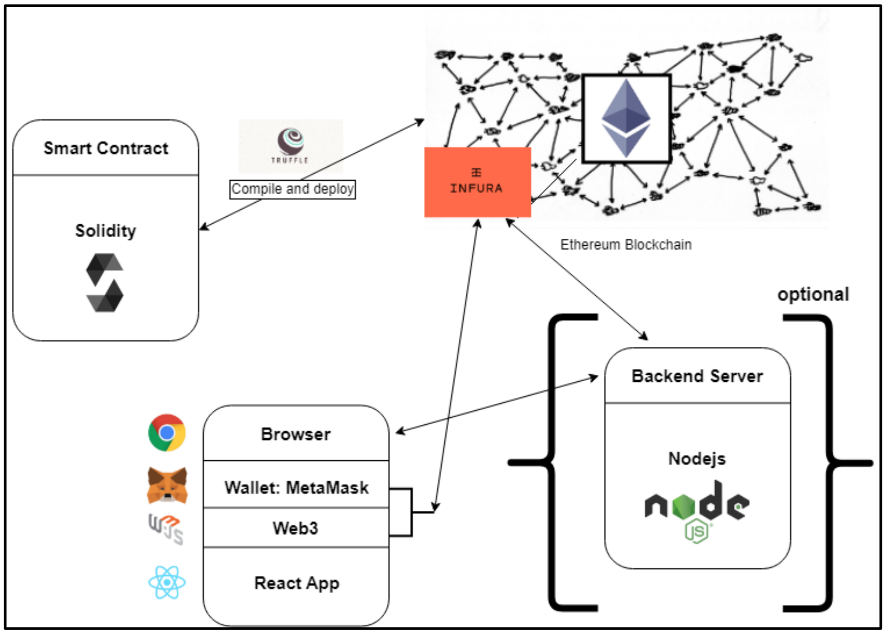


Fig 16: Overall working of system

This project on blockchain was used to create the decentralized messaging platform, which intends to give users secure and private communication without the use of centralized middlemen. Messages are stored in typical messaging applications on centralized servers that are vulnerable to hacking, putting users' private information at risk. This issue is resolved by this project using the decentralized, tamper-proof network provided by the Avalanche blockchain.

After the chat is in place, users have the option to make accounts, add contacts, send messages, and see their message history. Message notifications and conversation bubbles are among the characteristics of the user-friendly and intuitive design. Web3-enabled browsers like Metamask and the Avalanche Wallet both allow access to the chat application.

This Chat provides end-to-end encryption making sure that only the sender and receiver can see the message, which is one of its primary benefits. This implies that the communication cannot be read even if it is intercepted by a third party. All communication is kept secure because the encryption covers the user's contact list and message history as well.

This Chat app also incorporates the Avalanche wallet, which simplifies user transactions, in addition to its security features. As a result, users can immediately utilize the messaging app to send and receive the tokens as payment for goods and services. The app is now more adaptable and practical for users who want to use it for both payments and conversation.

Ultimately, the Chat app offers a secure, decentralized replacement for conventional messaging services. It makes use of the Avalanche blockchain to guarantee the confidentiality and security of user communication and makes easy use of the Avalanche wallet for frictionless transactions. The Chat App offers a contemporary solution that gives the user authority over their own data in light of the growing concern about data privacy and the requirement for secure communication.

VII. CONCLUSION AND FUTURE SCOPE

A decentralized chat application that we successfully created may be installed on the Avalanche blockchain or any other EVM-compatible blockchain.
Both developers and users found the experience of creating a chat app with Solidity and ReactJS to be beneficial. These two technologies were combined to create a decentralized chat network that is efficient, safe, and transparent. The app's chat data may be saved on the blockchain using Solidity's smart contract technology, guaranteeing that discussions are unchangeable and impenetrable. In the meantime, ReactJS offered a clear and user-friendly interface that allowed for frictionless user interaction. The software is a great option for people who value confidentiality and independence from centralized control due to its decentralized design and capacity to deliver a high level of privacy and security. This chat software has the potential to completely transform the way we communicate and share information online by utilizing the strength of blockchain technology and the adaptability of ReactJS.

Our app's functionality is quite constrained. By introducing features to remove messages, block users, or make buddy groups, we can make it better.
We might reduce the cost of utilizing our web app by limiting the maximum number of messages that can be sent or by employing an event log for brief communications.

VIII. REFERENCES

1. Wei, J., Chen, X., Wang, J., Hu, X., & Ma, J. (2021). Enabling (end-to-end) encrypted cloud emails with practical forward secrecy. IEEE Transactions on Dependable and Secure Computing, 19(4), 2318-2332.
2. Hussam, M. (2021). New lightweight hybrid encryption algorithm for cloud computing (LMGHA-128bit) by using new 5-D hyperchaos system. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 12(10), 2531-2540.
3. Li, Y., & Cao, Y. (2016). Performance evaluation and analysis of lightweight symmetric encryption algorithms for internet of things. International Journal of Reasoning-based Intelligent Systems, 8(1-2), 84-90.
4. Thabit, F., Alhomdy, S., Al-Ahdal, A. H., & Jagtap, S. (2021). A new lightweight cryptographic algorithm for enhancing data security in cloud computing. Global Transitions Proceedings, 2(1), 91-99.
5. Loukil, F., Ghedira-Guegan, C., Boukadi, K., & Benharkat, A. N. (2018). Towards an end-to-end IoT data privacy-preserving framework using blockchain technology. In Web Information Systems Engineering–WISE 2018: 19th International Conference, Dubai, United Arab Emirates, November 12-15, 2018, Proceedings, Part I 19 (pp. 68-78). Springer International Publishing.
6. Khacef, K., & Pujolle, G. (2019). Secure Peer-to-Peer communication based on Blockchain. In Web, Artificial Intelligence and Network Applications: Proceedings of the Workshops of the 33rd International Conference on Advanced Information Networking and Applications (WAINA-2019) 33 (pp. 662-672). Springer International Publishing.
7. Abdulaziz, M., Çulha, D., & Yazici, A. (2018, December). A decentralized application for secure messaging in a trustless environment. In 2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT) (pp. 1-5). IEEE.
8. Thakur, U., Chichmalkar, A., Sambhare, A., Chaturvedi, A., Khuspare, C., & Tembhe, N. (2022). DECENTRALISED CHAT APPLICATION.
9. Menegay, P., Salyers, J., & College, G. (2018, October). Secure communications using blockchain technology. In MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM) (pp. 599-604). IEEE.

10. Natanael, D., & Suryani, D. (2018). Text encryption in android chat applications using elliptical curve cryptography (ECC). Procedia Computer Science, 135, 283-291.
11. Tiwari, A., Agarwal, V., Aggarwal, Y., & Srivastava, U. (2022, March). Server Security in Cloud Computing Using Blockchain. In 2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS) (Vol. 1, pp. 963-966). IEEE.
12. David, S., & Canessane, A. (2021, June). A Centralized Blockchain-based Data Security System for Electrical Energy against Attacks. In 2021 International Conference on Communication, Control and Information Sciences (ICCISc) (Vol. 1, pp. 1-4). IEEE.
13. Bhutta, M. N. M., Khwaja, A. A., Nadeem, A., Ahmad, H. F., Khan, M. K., Hanif, M. A., ... & Cao, Y. (2021). A survey on blockchain technology: Evolution, architecture and security. Ieee Access, 9, 61048-61073.
14. Khan, M. A., & Rasheed, H. S. (2021, November). Secure Internet Voting using Blockchain Technology. In 2021 International Conference on Cyber Warfare and Security (ICCWS) (pp. 82-86). IEEE.
15. Kfoury, E. F., & Khoury, D. J. (2018). Secure End-to-End VoIP System Based on Ethereum Blockchain. J. Commun., 13(8), 450-455.
16. Cohn-Gordon, K., Cremers, C., Garratt, L., Millican, J., & Milner, K. (2018, October). On ends-to-ends encryption: Asynchronous group messaging with strong security guarantees. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security* (pp. 1802-1819).
17. Karbasi, A. H., & Shahpasand, S. (2020). A post-quantum end-to-end encryption over smart contract-based blockchain for defeating man-in-the-middle and interception attacks. *Peer-to-peer networking and applications*, *13*, 1423-1441.
18. Perez, A. J., & Ceesay, E. N. (2018, July). Improving end-to-end verifiable voting systems with blockchain technologies. In *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)* (pp. 1108-1115). IEEE.
19. Panja, S., & Roy, B. K. (2018). A secure end-to-end verifiable e-voting system using zero knowledge based blockchain. *Cryptology ePrint Archive*.
20. Chase, M., Deshpande, A., Ghosh, E., & Malvai, H. (2019, November). Seemless: Secure end-to-end encrypted messaging with less trust. In *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security* (pp. 1639-1656).

21. Rösler, P., Mainka, C., & Schwenk, J. (2018, April). More is less: on the end-to-end security of group chats in Signal, WhatsApp, and Threema. In *2018 IEEE European Symposium on Security and Privacy (EuroS&P)* (pp. 415-429). IEEE.
22. Hassani Karbasi, A., & Shahpasand, S. (2021). SINGLETON: A lightweight and secure end-to-end encryption protocol for the sensor networks in the Internet of Things based on cryptographic ratchets. *The Journal of Supercomputing*, *77*, 3516-3554.
23. Frosch, T., Mainka, C., Bader, C., Bergsma, F., Schwenk, J., & Holz, T. (2016, March). How secure is TextSecure?. In *2016 IEEE European Symposium on Security and Privacy (EuroS&P)* (pp. 457-472). IEEE.
24. Wei, J., Chen, X., Wang, J., Hu, X., & Ma, J. (2021). Enabling (end-to-end) encrypted cloud emails with practical forward secrecy. *IEEE Transactions on Dependable and Secure Computing*, *19*(4), 2318-2332.
25. Bai, W., Pearson, M., Kelley, P. G., & Mazurek, M. L. (2020, September). Improving non-experts' understanding of end-to-end encryption: an exploratory study. In *2020 IEEE european symposium on security and privacy workshops (EuroS&PW)* (pp. 210-219). IEEE.