International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

DECENTRALIZED APPLICATION FOR SECURE MESSAGING IN A TRUSTLESS ENVIRONMENT

RAHEELA TABASSUM[1], DR. MOHD.ABDUL BARI[2], MR.L.K SURESH KUMAR[3]
*1. ME Student, Department Of Computer Science, ISL ENGINEERING COLLEGE.
*2. Head Of Department of Computer Science, ISL ENGINEERING COLLEGE
*3. Professor Of CSE Dept MVSR ENGINEERING COLLEGE

Abstract

Block chain technology has been seeing widespread interest as a means to ensure the integrity, confidentiality and availability of data in a trustless environment. They are designed to protect data from both internal and external cyber attacks by utilizing the aggregated power of the network to resist malicious efforts. In this article, we will create our own decentralized messaging application utilizing the Ethereal Whisper protocol. Our application will be able to send encrypted messages both securely and anonymously. We will utilize the ethereal platform to deploy our block chain network. This application would be resistant to most suppression tactics due to its distributed nature and Adaptability of its communication protocol.

Keywords: Block chain, Ethereal, Whisper, Peer-to-Peer P2P) Networks, Distributed Messaging, Decentralized Application (dApp).

1.INTRODUCTION

In recent times, it is becoming increasingly vital that communication not only be secure but also anonymous. The presence of mass surveillance programs as well as cyber attacks focused on compromising messaging applications highlight the need for maintaining the anonymity of communicating parties. In this article, we developed a decentralized messenger application utilizing the Ethereal Whisper protocol. Using our application, two users can engage in secure and anonymous communication which is encrypted end-to-end and resistant to network traffic

Analysis.

1.1BACKGROUND AND DEFINITIONS

This section provides all the necessary background required for this work. We will briefly discuss secure messaging as well as block chain technology. This will be followed by an overview of the ethereal platform with emphasis made on the Whisper protocol.

A. Secure Messaging A significant amount of current electronic communication is still placed over a number of legacy

Protocols such as SMS/GSM, SMTP and centralized messengers which were not designed with end-to-end security as a requirement [1]. These methods routinely broadcast recipient and sender information and therefore provide limited anonymity capabilities. In addition, they are more prone to suppression due to the storage and transmission requirements by intermediate servers. Communication systems with a peer-to-peer (P2P) architecture attempt to exchange messages directly between the participants in the network rather than rely on centralized servers for the storage and forwarding of messages. These systems commonly utilize Distributed Hash Tables (DHTs) for mapping usernames to IP addresses without the need for a centralized authority [1, pp. 20]. However, these P2P solutions such as Kademlia [2] only partially anonymize the recipient and sender. In addition, they do not provide any capability to hide which participants Are engaged in a conversation, and do not prevent protocol messages from being associated to a particular conversation. Furthermore, it is possible for global network adversaries to view the flow of traffic between the participants of a conversation.

B. Block chain

A block chain is an append-only distributed database operating within a P2P network whereby each peer has a partial or full copy of the block chain [3]. Due to their distributed nature, block chains are highly available and fault tolerant even if a large scale attack is mounted on the network. Changes in the block chain are conducted through transactions which are broadcasted and verified by all the nodes in the network. After verification the change is appended to the block chain. To circumvent ‘double-spending’ a number of transactions are grouped into a block and are then verified simultaneously [4]. The block is then appended to the block chain usually through a proof-of-work (PoW) mechanism which requires computationally intensive work to be conducted by a ‘mining’ node. Mining nodes are incentivized through rewards while other nodes interested in only posting transactions provide a fee. In this way integrity of the data is ensured.

Ownership of accounts is maintained through asymmetric cryptography whereby a public key is shared with the network and the private key is known only to the holder. Only the holder of the private key can digitally sign transactions on the block chain, whereas the public key can be utilized as a personal address that other users can send assets digitally or interact with in some way. Furthermore, users can usually create public/private key pairs anonymously thus protecting their identities. Therefore, block chains provide an elegant means for handling the transmission of critical data or digital assets in untrusted distributed environments.

C. Ethereal

Ethereal is an open-source block chain platform with distributed computing that allows developers to run smart contracts [5]. These are collections of code that exist on the Ethereal block chain and are able to run without possibility of censorship, fraud, third-party interference or downtime. Data integrity is ensured through the use of Merkle Patricia tree which guarantees a data structure that is cryptographically authenticated [6].

The Ethereal technology stack is given in Figure 1:

Fig. 1.1. Ethereal technology stack.

1) Mist Browser: an interface to access various dApps.

2) Decentralized Applications:

3) Ethereal Virtual Machine (EVM): The EVM is an abstraction layer which sits above the hardware clients and handles internal computation and state [7]. All full nodes perform the same code execution on the EVM. The EVM is essentially a computer that can execute code and contain data with absolute availability and fault tolerance as long as the network is sufficiently large.

4) Whisper: it is Eternal’s P2P communication protocol for decentralized applications [8]. P2P

Communication between nodes in the Whisper network utilizes the DΞVp2p Wire Protocol. A dApp instance can create an identity within a node that is connected to Whisper. This identity is needed to send or receive messages. Once a message is sent, it is, in theory, supposed to be routed through every Whisper node. This makes it necessary to implement a PoW algorithm to prevent denial of- service (DoS) attacks. Messages are only processed and further routed if their PoW is found to exceed a predefined threshold. Furthermore, Whisper allows dApp developers to configure the anonymity and security of their messages. All messages on Whisper are initially encrypted and sent through a basic wire-protocol called DΞVp2p which Supports various sub-protocols such as Whisper. The message is further encrypted by the DΞVp2p protocol. Currently, all Whisper messages are required to be either asymmetrically encrypted using the Elliptic Curve Integrated Encryption Scheme (ECIES) together with the

SECP-256k1 public key or symmetrically encrypted using the Advanced Encryption Standard Galios/Counter Mode (AES-GCM) with a random 96-bit nonce [9]. Multiple asymmetric and symmetric keys may be owned by a single node. If a message is successfully decrypted it is then

Forwarded to its respective dApp. Using the web3-shh package, one can interact with the Whisper protocol. The following methods within the web3-shh package are used.

- web3.shh.newKeyPair (): This method generates a new private and public key pair used for message encryption and decryption.
- web3.shh.newSymKey (): This method randomly generates a symmetric key and stores the key under an ID. This key is shared between communicating parties and used to encrypt and decrypt messages.
- web3.shh.getPublicKey (kId): This method returns the associated public key for a given key pair ID.
- web3.shh.getSymKey (id): This method returns the symmetric key for a given ID.
- web3.shh.newMessageFilter (options): This method creates a new message filter within the node to be used for polling messages that satisfy a set of criteria given.
- web3.shh.post (object [, callback]): This method is called when Whisper message is to be posted to the network.

EXISTING SYSTEM

Suggestions have been made to use the block chain as a platform to store and transfer messages. The data to be stored on the block chain is encrypted to prevent unauthorized access in these systems. In [11], a decentralized personal management system which encrypted user data was suggested. Sending encrypted messages on payment platforms has also been suggested [12]. However, due to the open nature of block chain, there are difficulties in ensuring the anonymity of two users trying to conduct a transaction or communicate [13].Others have suggested using existing P2P communication protocols to route messages in a more anonymous fashion [14] [15]. Some of these protocols are unable to prevent the network from being attacked through flooding and do not provide guarantee anonymity for the sender [1, pp. 21].

The Whisper protocol is designed to provide complete anonymity and is resistant to certain attacks and network analysis. In [16], a messenger and coupon exchanging application was built using the Whisper protocol. Their implementation necessitated the exchange of a topic between the sender and the recipient for each session. Furthermore, this topic needed to be shared over some other secure channel other than the application. Therefore, this implementation was not designed for functioning in a trustless environment. In addition, the implementation relied on some features which do not exist in the newer versions of the protocol such as the ability to send unencrypted messages with a topic. In their implementation the unencrypted message could reveal information about the recipient and sender. For this reason, newer versions of the Whisper protocol require all messages to be encrypted to ensure anonymity by default. As far as we are aware, we are the first to implement a decentralized messaging application using the Whisper protocol that preserves the complete anonymity of the participants. Due to the Whisper protocol being an ongoing rapidly evolving project, certain portions of the official documentation and user guides were inconsistent, incorrect or outdated. This article brings together the most up-to-date information at the time of writing concerning the Whisper protocol through a direct analysis of the open source code.

Figure 1. Stakeholders from the traditional IdMS model.

Although the proposed blend CAC mechanism has demonstrated these attractive features, using

Block chain to enforce AC policy in space systems, it also incurs new challenges in performance and security. The transaction rate is associated with confirmation time of the block chain data, which depends on the block size and the time interval between the generations of new blocks. Thus, the latency for transaction validation may not be able to meet the requirement in real-time SSA scenarios. In addition, as the amount of transactions increases, the block chain becomes large. The continuously growing data introduces more overhead on storage and computing resources of each client, especially for resource constrained devices. Furthermore, the block chain is susceptible to majority attack (also known as 51% attacks), in which once an attacker takes over 51% computing power of network by colluding selfish miners, they are able to control the block chain and reverse the transactions. Finally, since the block chain data is open to all nodes joined the block chain network, such a property of transparency inevitably brings privacy leakage concerns. More research efforts are necessary to improve the trade-off when applying the Blend CAC in practical scenarios..

PROPOSED SYSTEMSYSTEM DESIGN:

This section is divided into four parts. Firstly, we will give a description of the problem. In the second part, we will discuss the design choices we made. Thirdly, we will explore the software architecture of the implemented solution. Lastly, we describe the operation of the application.

Problem Statement

Listed below are characteristics we deemed vital for an anonymoussecure messaging application expected to operate in entrusted environments:

***End-to-End encryption**:* Only the users should be able to decipher thedata being stored or communicated.

Anonymous Sender*:* The source of a particular message cannot be attributed to a specific needed. Therefore, it was decided to implement the 'account management' functionality locally. Possible future extensions could include backing-up encrypted user data on a P2P distributed file system. Once the application is connected to a local Geth client who serves as an ethereal node, it can communicate with other nodes that are its peers on the Whisper Network using the 'messaging function'. The 'messaging function' utilizes the Ethereal JavaScript API web3.js library and is capable of transmitting signed

encrypted messages to peers on the Whisper network. Since the Geth instance handles Whisper identities it would not be in the user's interest for the Gethinstance tobe run by a third-party.

Another issue was handling messaging between users that would best preserve their anonymity and allow for plausible deniability. Since messagesare able to encrypt both asymmetrically and symmetrically we first explored using signed encrypted messages for every message. However, this does not allow the sender to plausibly deny having sent a message in case of the loss of the key or coercion. Hence, in the 'messaging function'private keys are only used to establish the session while symmetric keys are used toencrypt/decrypt messages. The recipient can be certain of the identity of the sender since the symmetric key was signed by the sender using his private key. However, themessages cannot be used as conclusive evidence as they could have been sent by anyone with access to the symmetric key. It is only necessary for the symmetric keyto have been transmitted to the recipient for there to be plausible deniability.

LITERATURE REVIEW

N. Unger, S. Dockhand, J. Bonneau, S. Fahl, H. Perl, I. Goldberg and M. Smith, "SoK:Secure Messaging", ***2015 IEEE Symposium on Security and Privacy*****, pp. 22, 2015.**

Motivated by recent revelations of widespread state surveillance of personal communication, manyproducts now claim to offer secure and private messaging. This includes both a large number of new projects and many widely adopted tools that have added security features. The intense pressure in the past two years to deliver solutions quickly has resulted in varying threat models, incomplete objectives, dubious security claims, and a lack of broad perspective on the existing cryptographic

literature on secure communication. In this paper, we evaluate and systematize current secure messaging solutions and propose an evaluation framework for their security, usability, and ease-of-adoption properties. We consider solutions from academia, but also identifyinnovative and promising approaches used "in the wild" that are not considered by the academic literature. We identify three key challenges and map the design landscape for each: trust establishment, conversation security, and transport privacy. Trust establishment approaches offering strong security and privacy features perform poorly from a usability and adoption perspective, whereas some hybrid approaches that have not been well studied in the academic literature might provide better trade-offs in practice. In contrast, once trust is established, conversation security can be achieved without any user involvement in most two-party conversations, though conversations between larger groups still lack a good solution. Finally, transport privacy appears to be the most difficult problem to solve without paying significant performance penalties.

P. Maymounkov and D. Mazieres, "Kademlia: A Peer-to-Peer Information System Based on the XOR Metric," in *Peer-to-Peer Systems.* Springer, 2002, pp. 53–65.

We describe a peer-to-peer distributed hash table with provable consistency and performance in a fault-prone environment. Our system routes queries and locates nodes using a novel XOR-basedmetric topology that simplifies the algorithm and facilitates our proof. The topology has the property that every message exchanged conveys or reinforces useful contact information. The system exploits this information to send parallel, asynchronous query messages that tolerate nodefailureswithout imposing timeout delays on users.t DDDAS is a conceptual framework that synergistically combines models and data in orderto facilitate the analysis and prediction of physical phenomena. 6, 7 In a SSA applications, DDDAS is a variation of adaptive state estimation that uses computational feedback rather than physicalfeedback to enhance the information content of measurements.

The feedback loops in DDDASinclude a data assimilation loop and a sensor reconfiguration loop. The data assimilation loop calculates the physical system simulation by using sensor data error to ensure that the trajectory of the simulation more closely follows the trajectory of the physical system. As a fundamental aspect of DDDAS, the sensor reconfiguration loop seeks to manage the physical sensors in order to enhance the information content of the collected data. The simulation based on computational feedback process guides the sensor reconfiguration and the data collection, and in turn, improves 5 the accuracy of the physical system environmental assessment (e.g., space weather and Restacking). For sensor management, DDDAS develops runtime software methods for real- time control such as access control.

1. SYSTEM ANALYSIS

FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a verygeneral plan for the project and some cost estimates. During system analysis the feasibility study ofthe proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are,

ECONOMICALFEASIBILITY
TECHNICALFEASIBILITY
SOCIALFEASIBILITY

Economic Feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available.Only the customized products had to be purchased.

Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This willlead to high demands being placed on the client. The developed system must have a modest.

Social Feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends onthe methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

REQUIREMENTS ANALYSIS

Requirement Analysis is the first and important phase of the software developing activity in developing any kind of project effectively. I started to list out all the functionalities thatmy application should provide. There have been some minor changes with respect to the functionalities over the course of development. Following are the requirements that have been implemented in this project

The project involved analyzing the design of few applications so as to make the application more users friendly. To do so, it was really important to keep the navigationsfrom one screen to the other well ordered and at the same time reducing the amount of typing the user needs to do. In order to make the application more accessible, the browser version had to be chosen so that it is compatible with most of the Browsers.

REQUIREMENTS SPECIFICATION

Functional requirementsDigitalization

Secure Server connection

Software Requirements

For developing the application the following are the

Software Requirements:

Python Blockchain

Hardware Requirements

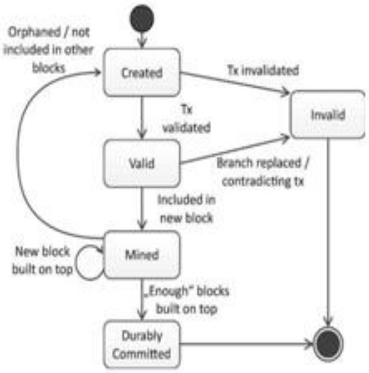
For developing the application the following are the

Hardware Requirements:

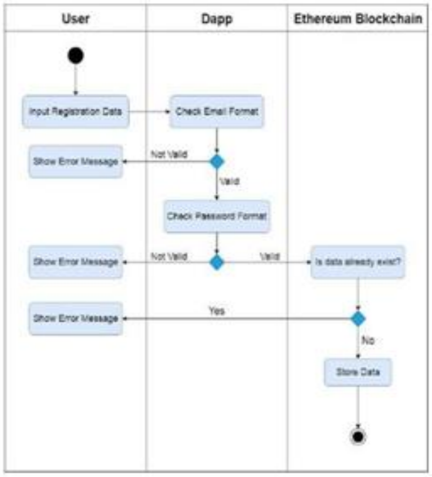
Space on Hard Disk: minimum 512MB

2. PRESENT WORK WITH DIAGRAMS

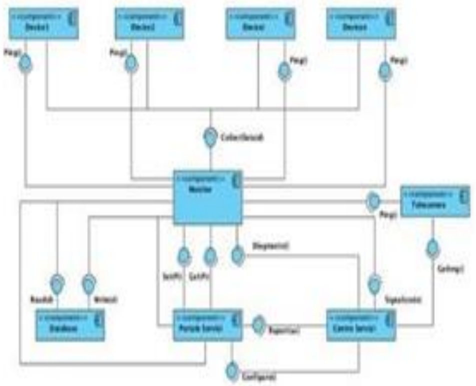
STATECHART DIAGRAM

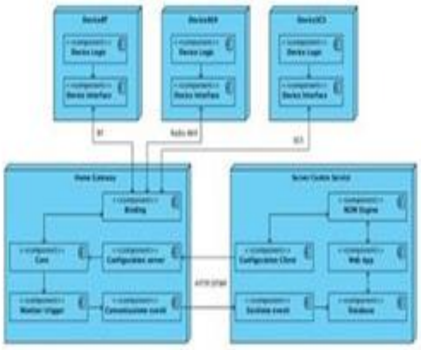


ACTIVITY DIAGRAM



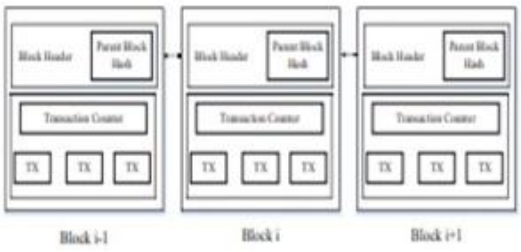
COMPONENT DIAGRAM



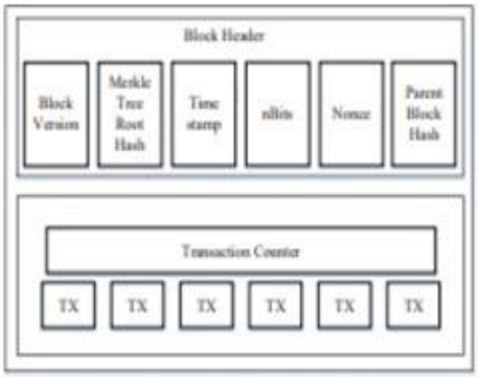


3. SYSTEM DESIGN

BLOCK CHAIN ARCHITECTURE



An example of block chain which consists of acontinuous sequence of blocks.



Block structure

Block chain is a sequence of blocks, which holds a complete list of transaction records like conventional public ledger. Figure 1 illustrates an example of a Block chain. With a previous blockhash contained in the block header, a block has only one parent block. It is worth noting that uncle blocks (children of the block's ancestors) hashes would also be stored in ethereal Block chain. Thefirst block of a Block chain is called genesis block which has no parent block. We then explain the internals of Block chain in details.

A. **Block** A block consists of the block header and the block body as shown in Figure. In particular, the block header includes:

(i) Block version: indicates which set of block validationrules to follow.
(ii) Merkle tree root hash: the hash value of all the transactions in the block.
(iii) Timestamp: current time as seconds in universal time since January 1, 1970.
(iv) NBits: target threshold of a valid block hash.
(v) Nonce: an 4-byte field, which usually starts with 0 and increases for every hash calculation (will be explained in details in Section III).

(vi) Parent block hash: a 256-bit hash values that points to the previous block. The block body is composed of a transaction counter and transactions. The maximum number of transactions that a block can contain depends on the block size and the size of each transaction. Block chain uses an asymmetric cryptography mechanism to validate the authentication of transactions. Digital signature based on asymmetric cryptography is used in an untrustworthy environment. We next briefly illustrate digital signature.

B. Digital Signature Each user owns a pair of private key and public key. The private key that shall be kept in confidentiality is used to sign the transactions. The digital signed transactions are broadcasted throughout the whole network. The typical digital signature is involved with two phases: signing phase and verification phase. For instance, an user Alice wants to send another user Bob a message.

(1) In the signing phase, Alice encrypts her data with her private key and sends Bob the encrypted result and original data.

(2) In the verification phase, Bob validates the value with Alice’s public key. In that way, Bob could easily check if the data has been tampered or not. The typical digital signature algorithm used in Block chains is the elliptic curve digital signature algorithm (ECDSA).

C. Key Characteristics of Block chain In summary, Block chain has following key characteristics.

- Decentralization. In conventional centralized transaction systems, each transaction needs to be validated through the central trusted agency (e.g., the central bank), inevitably resulting to the cost and the performance bottlenecks at the central servers. Contrast to the centralized mode, third party is no longer needed in Block chain. Consensus algorithms in Block chain are used to maintain data consistency in distributed network.

- Persistency. Transactions can be validated quickly and invalid transactions would not be admitted by honest miners. It is nearly impossible to delete or rollback transactions once they areincluded in the Block chain. Blocks that contain invalid transactions could be discovered immediately.

7. TEST CASES, SCREENS AND REPORTS

TEST CASES

This section is divided into three parts. In the first part, we review the encryption algorithm used. Secondly, an analysis on the application’s resistance to DoS attacks is made. Finally, the resistance of the application to network traffic analysis is examined.

A. Encryption Algorithm

The application utilizes the Elliptic Curve Integrated Encryption Scheme (ECIES) with the recipient’s SECP- 256k1 key to share the symmetric keys (AES-256). This initial message is signed by the sender. As the sender’s signature is part of the message and can only be accessed after decryption, the signature can only be accessed by the recipient. Subsequent messages between the sender and recipient are encrypted using the shared symmetric keys.

In this way, only the participants in a chat can decrypt the messages. After the session has been completed the symmetric keys are disposed.

B. Resistance to Denial of Service attacks

Since every Whisper message is routed to every node thatit is able to reach, the network might besusceptible to twotypes of attacks:

_ Expiry Attack: Messages are set to have a long Time- to-Live (TTL) on the envelope.

_ Flood Attack: The network is repeatedly sent messages. An expiry attack is averted by using a system that rates messages by taking into account the message size, TTL and POW. Messages that have a smaller size, lower TTL and higher POW are considered to have a higher rating. This ratingimpacts how long the message is stored as well as its forwarding priority. Lower rated messages would be removed first in the event of a DoS attack, thus preventing an expiry attack.

A Flood attack can be averted by requiring the sender to conduct POW computation and posting the result to the EnvNonce field within the message envelope. If the PoW is below the amount required, then the message is not further routed.

C. Resistance to Network Traffic Analysis

Since every message is routed to every node in the network, it is impossible to determine the identity of the recipient of a certain message. However, a global

message.

THE MODEL OF DECENTRALIZED ENERGY TRADING

The prominent example of a trading energy system that uses block chain is Brooklyn micro grid as shown in Figure 5 which is designed in the USA. It can be described as a solution that combines the security and transparency between the neighbors that is offered by block chain concept. The goal of the system is to measure the ability of block chain technology adapted in order to buy and sell the energy among the neighbors and how effective block chain technology is adopted.

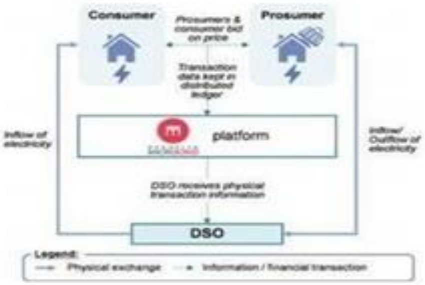

Consumer
Prosumers & consumer bid on price
Prosumer
Transaction data kept in distributed ledger
Inflow of electricity
platform
Inflow/ Outflow of electricity
DSO receives physical transaction information
DSO
Legend:
Physical exchange
Information / financial transaction

Fig 5. Brooklyn micro grid network

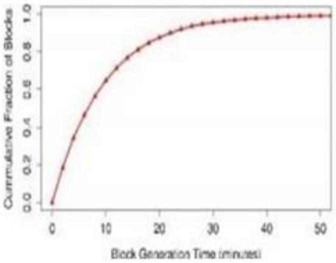


Fig 6. Block generation time in Bit coin

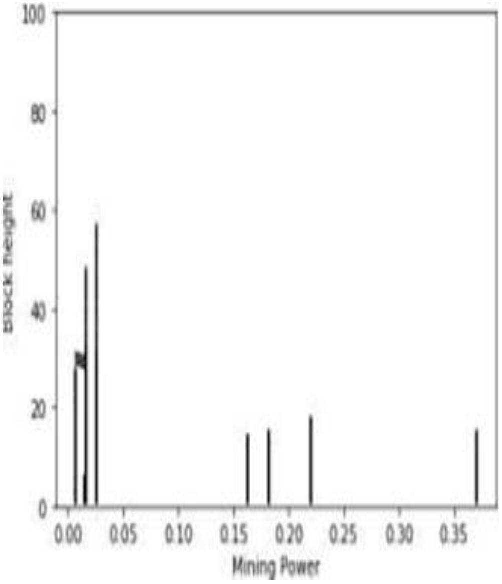
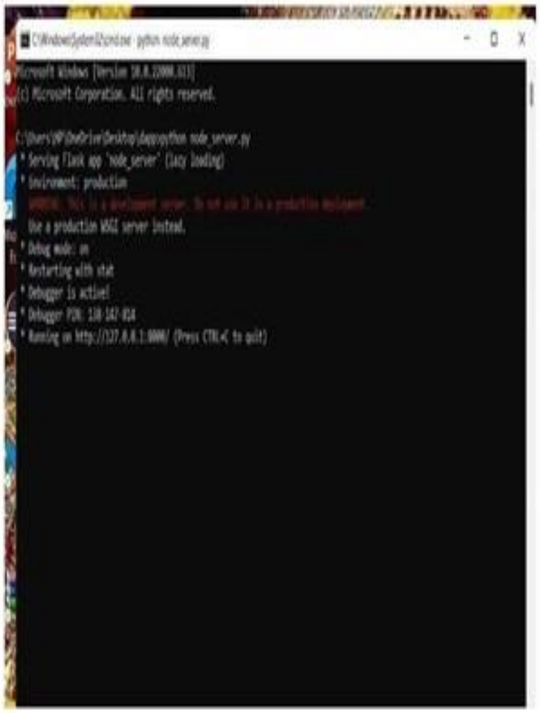


Fig 7. Performance of dishonest miner

SCREENS

Screen-1



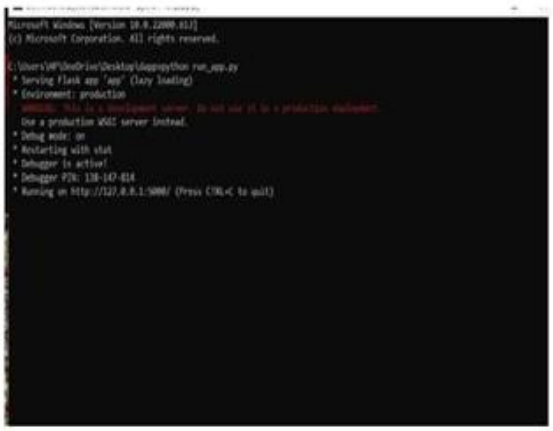
```
Microsoft Windows [Version 10.0.22000.613]
(c) Microsoft Corporation. All rights reserved.

C:\Users\HP\OneDrive\Desktop\dapp>python node_server.py
 * Serving Flask app 'node_server' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 130-147-014
 * Running on http://127.0.0.1:8000/ (Press CTRL+C to quit)
```

From the above screen we have run node app
First we have open the file explorer and run the node .py in command line it displays a link to server and the below screen depicts the node.py application and

it is a demo application for my project

Screen-2



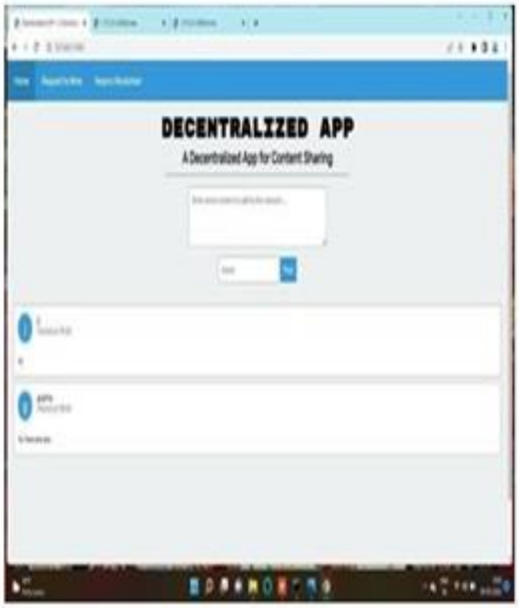
```
Microsoft Windows [Version 10.0.22000.613]
(c) Microsoft Corporation. All rights reserved.

C:\Users\HP\OneDrive\Desktop\dapp>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 deployment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 138-147-814
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

From the above screen we have run app

After the node.py application we have to run the app.py application it also run in command line it also gives a link to server and this was the actual link to the front end of my project

Screen-3



From the above screen we have seen the messages already shared

After the link to server is stored in specified browser thebelow page which depicts the frontend page of project

Screen-4

After selecting the request to mine then the above screen appear and displays the message saved to us in server

CONCLUSION

The concept of block chain technology is used for trading the renewable energy system inan environment among the neighbors in the peer-to-peer network. We discussed a model for trading energy in a small environment by using block chain technology and then we analyzed security issues that might be occurred. The performance of the attacker is also presented. Blockchain technology with cryptographic embedded to support the security issues can become a possible solution for the future to create a secure trading renewable energy system in the environment among the neighbors. The energy and commodity transaction life cycle, even for simple transactions, involves a multitude of processes within eachcompany and across market participants.

Block chain turns both currencies and commodities into a digital form without relying on middleman which allows one person totrade with another. For the future, the strategy is neededin order to prevent the various attacks, especially in the overlay network

FUTURE WORK

An unexplored issue was accounting for the possibility of the user being offline or an unexpected network failure. The messages intended for a specific user could expire during this interval which would mean that the user would not be able to retrieve those messages. One solution could be a decentralized mail server capable of resending the expired messages to the network with sufficientPOW. However,since the mail server would notknow whether the recipient had actually received the message it would have to continue doing so indefinitely while new messages would continue being added. It is unlikely that anymail server could be capable of managing the appropriate POWand in any case would result in a DoS attack on the Whisper Network. An alternative solution might be to send the expired message directly to the node upon reconnection however this would result in the exposure of their identity as a recipient. Even if this could be solved by routing the message through the Whisper network upon an anonymouslegitimate request by the recipient, thereis an issue with storage and incentivizing storage by the participants in the network. One possible implementation could be a ‘social storage’ scheme.

REFERENCES

1. N.Z. Aitzhan and D. Svetinovic, "Security and Privacy in Decentralized Energy Trading through Multi-Signature, Blockchain and Anonymous Messaging Streams", IEEE Transactions on Dependable and Secure Computing, 2016.

2. PwC Global Power, "Blockchain- an Opportunity for Energy Producers and Consumers?" PwCGlobal Power and Utilities, 2016. G. Karame, E. Androulaki, "Bitcoin and Blockchain Security", Information Security and Privacy Series, United States of America, Artech House, 2016.

3. G.Karame, E.Androulaki,”Bitcoin andBlockchain Security”, Information Security and PrivacySeries, United States of America, Artech House,2016.

4. P. Danzi, A. Marko, C. Stefanovic, and P. Popovski, “Distributed Proportional-FairnessControl in Microgrids via Blockchain Smart Contracts”, arXiv: 1705.01453v2 [cs.MA], 2017. [5]F. Imbault, M. Swiatek, R.de Beaufort and R. Plana, "The Green Blockchain Management Decentralized Energy Production and Consumption", IEEE International Conference on Environmentand Electrical Engineering(EEEIC/ I&CPS Europe), 2017.

5. S. Nakamoto, “Bitcoin: A Peer-to-peer Electronic CashSystem”, 2008.