A Decentralized Application for

Secure Messaging

**In A Trustless Environment**

# Abstract

Blockchain 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 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 blockchain network. This application would be resistant to most suppression tactics due to its distributed nature and Adaptability of its communication protocol.

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

2) Decentralized Applications.

3) Whisper: it is Ethereum’s P2P communication protocol for decentralized applications. P2P communication between nodes in the Whisper network utilizes the D Vp2p 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.

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

**CHAPTER-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.

**1.1 BACKGROUND AND DEFINITIONS**

This section provides all the necessary background required for this work. We will briefly discuss secure messaging as well as blockchain 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 several legacies

Protocols such as SMS/GSM, SMTP, and centralized messengers 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. Besides, 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. Besides, they do not provide any capability to hide which participants

Are engaged in a conversation, and do not prevent protocol messages from being associated with a particular conversation. Furthermore, global network adversaries can view the flow of traffic between the participants of a conversation.

**B. Blockchain**

A blockchain is an append-only distributed database operating within a P2P network whereby each peer has a partial or full copy of the blockchain [3]. Due to their distributed nature, blockchains are highly available and fault-tolerant even if a large scale attack is mounted on the network. Changes in the blockchain are conducted through transactions that are broadcasted and verified by all the nodes in the network. After verification, the change is appended to the blockchain. To circumvent ‘double-spending’ several transactions are grouped into a block and are then verified simultaneously [4]. The block is then appended to the blockchain 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, the 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 blockchain, 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, blockchains 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 blockchain platform with distributed computing that allows developers to run smart contracts [5]. These are collections of code that exist on the Ethereal blockchain and can run without the 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:

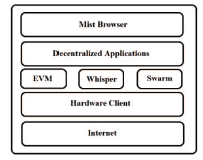
****

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 that 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 Galois/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.

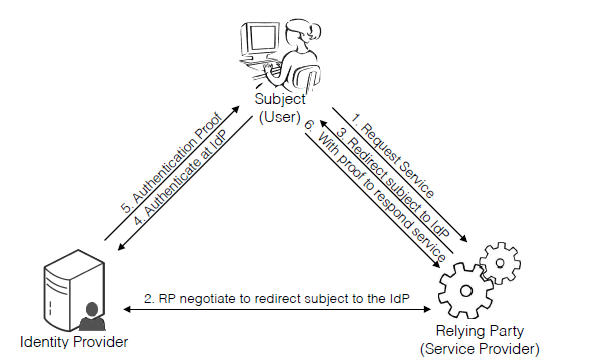
The following methods within the web3-shh package are used.

* 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 blockchain as a platform to store and transfer messages. The data to be stored on the blockchain is encrypted to prevent unauthorized access in these systems. In [11], a decentralized personal management system in which encrypted user data was suggested. Sending encrypted messages on payment platforms has also been suggested [12]. However, due to the open nature of blockchain, 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 were 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. Also, 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

Blockchain to enforce AC policy in space systems, it also incurs new challenges in performance and security. The transaction rate is associated with the confirmation time of the blockchain 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. Besides, as the amount of transactions increases, the blockchain becomes large. The continuously growing data introduces more overhead on the storage and computing resources of each client, especially for resource-constrained devices. Furthermore, the blockchain is susceptible to majority attack (also known as 51% attacks), in which once an attacker takes over 51% of the computing power of the network by colluding selfish miners, they can control the blockchain and reverse the transactions. Finally, since the blockchain data is open to all nodes joined the blockchain 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.

#### CHAPTER-2

# LITERATURE SURVEY

1. **N. Unger, S. Dechand, 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, many products 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 identify innovative 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.

1. **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 query and locates nodes using a novel XOR-based metric 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 node failures without imposing timeout delays on users.

DDDAS is a conceptual framework that synergistically combines models and data to facilitate the analysis and prediction of physical phenomena.6, 9 In an SSA application, DDDAS is a variation of adaptive state estimation that uses computational feedback rather than physical feedback to enhance the information content of measurements. The feedback loops in DDDAS include 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 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 RSO tracking). For sensor management, DDDAS develops runtime software methods for real-time control such as access control.

1. **Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, “An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends,” *2017 IEEE International Congress on Big Data (Big Data Congress)*, pp. 558–560, 2017.**

The Blockchain, the foundation of Bitcoin, has received extensive attention recently. The Blockchain serves as an immutable ledger that allows transactions to take place in a decentralized manner. Block chain-based applications are springing up, covering numerous fields including financial services, reputation system and Internet of Things (IoT), and so on. However, there are still many challenges of blockchain technology such as scalability and security problems waiting to be overcome. This paper presents a comprehensive overview of blockchain technology. We provide an overview of blockchain architecture firstly and compare some typical consensus algorithms used in different blockchains. Furthermore, technical challenges and recent advances are briefly listed. We also layout possible future trends for blockchain.

The blockchain protocol has been recognized as the potential candidate to revolutionize the fundamentals of IT technology because of its many attractive features and characteristics, such as supporting decentralization and anonymity maintenance,10 as well as a fundamental protocol of Bitcoin,11 the first digital currency. In this paper, a Blockchain-Enabled, Decentralized, Capability-based Access Control (Blend CAC) scheme is proposed to enhance the security of space applications. Blend CAC provides a decentralized, scalable, fine-grained and lightweight authentication and access control mechanism to protect devices, services, and information in space networks. To achieve secure identity authentication, a decentralized authentication mechanism is implemented on the blockchain and aims at creating virtual trust zones to allow all distributed entities to identify each other and communicate securely in the trustless network environment. An identity-based capability token management strategy is presented and the federated authorization delegation mechanism is illustrated. A proof-of-concept prototype has been developed and evaluated on a private Ethereal blockchain network, and the experimental results demonstrate the feasibility and effectiveness of the proposed Blend CAC scheme.

1. **V. Buterin, “A next-generation smart contract and decentralized application platform,” *white paper*, 2014.**

Blockchain technologies paired with smart contracts exhibit the potential to transform the global insurance industry. The recent evolution of smart contracts and their fast adoption allow to rethink processes and to challenge traditional structures. Therefore, a special focus is on the analysis of the underlying technology and recent improvements. Further, we provide an overview of how the insurance sector may be affected by blockchain technology. We emphasize current challenges and limitations by analyzing two promising use cases in this area. We find that realizing the full potential of blockchain technology requires overcoming several challenges including scalability, the incorporation of external information, flexibility, and giving access to the schemes.

Capability-based Access Control (Cap AC) utilizes the concept of capability that contains rights granted to the entity holding it.18 The capability is defined as tokens, tickets, or keys that give the possessor permission to access an entity or object in a computer system.27 The Cap AC has been implemented in many large-scale projects, like IoT@Work.28 However, the direct application of the original concept of the Cap AC model in a distributed network environment has raised several issues, like capability propagation and revocation. To tackle these challenges, a Secure Identity-Based Capability (SICAP) System17 was proposed to provide a prospective capability-based AC mechanism in distributed networks. By using an exception list, the SICAP enables monitoring, mediating, and recording capability propagations to enforce security policies as well as achieving rapid revocation capability.17 By introducing a delegation mechanism for the capability generation and propagation process, a Capability-based Context-Aware Access Control (CCAAC) model was proposed to enable contextual awareness in federated devices.29 A federated delegation mechanism enables the CCAAC model has great advantages in terms of addressing scalability and heterogeneity issues in IoT networks. As a user-driven AC model, the Cap AC supports machine-to-machine (M2M) communication and presents great scalability and flexibility to deal with spontaneous and dynamic changes in a distributed network environment. However, management of capability propagation becomes difficult without a proper delegation and revocation mechanism.

1. **Sandi Rahmadika , Diena Rauda Ramdania., “Security Analysis on the Decentralized Energy Trading System Using Blockchain Technology” 2016**

Blockchain turns both currencies and commodities into a digital form without relying on middleman which allows one person to trade with another include trading renewable energy. Blockchain technology as a secure and low-cost platform to track the billions of eventual transactions in a distributed energy economy has attracted the attention of experts in various fields of science. The current form of centralized energy trading system is still suffering from security concerns, quality of service, and to name a few. A decentralized energy system using blockchain technology allows the parties to create a trading energy transaction via micro-grid. The blockchain technology offers the promise of an immutable, single source of truth from multiple sources without third-party involvement. In this paper, we describe, explore and analyze the prominent implementation of blockchain technology in the energy sector. Furthermore, we analyze the security issues and highlight the performance of several attacks that might be occurred in the proposed system.

The concept of blockchain technology is used for trading the renewable energy system in an environment among the neighbors in the peer-to-peer network. We discussed a model for trading energy in a small environment by using blockchain 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 each company and across market participants. Blockchain turns both currencies and commodities into a digital form without relying on middleman which allows one person to trade with another. For the future, the strategy is needed to prevent the various attacks, especially in the overlay network.

**CHAPTER-3**

**PROPOSED SYSTEM**

**3.1 SYSTEM DESIGN:**

This section is divided into four parts. Firstly, we will describe 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.

**A. Problem Statement**

Listed below are characteristics we deemed vital for an anonymous secure messaging application expected to operate in untrusted environments:

* ***End-to-End encryption****:* Only the users should be able to decipher the data being stored or communicated.
* ***Anonymous Sender****:* The source of a particular message cannot be attributed to a specific entity.
* ***Anonymous recipient:***The destination of a particular message cannot be attributed to a

Specific entity.

* ***Anonymous Participants:***The set of participants engaged in a conversation cannot be determined.
* ***Unlinkability*:** Only the participants engaged in a conversation can associate two or more protocol messages as belonging to the same conversation.
* ***Resistant to Attacks by a Global Adversary*:** The anonymity of the protocol should not be threatened by global adversaries.
* ***Resistant to Flood and Spam attacks*:** Bulk messaging and DoS attacks should not be able to significantly impact the availability of the system.
* ***Plausible deniability*:** It should be possible for any entity to deny having sent a particular message. To ensure our application would satisfy these various security and anonymity features we have decided to build a decentralized messaging application that utilizes Whisper as its communication protocol.

**B. Design Considerations**

First, it is necessary to explore whether there is a need for the major functions of the application “account management” and “messaging” function to be moved to a smart contract. The “account management” function contains methods to create an account as well as manage friends. Due to the open nature of smart contracts on Ethereal, it is possible for the contact information to be viewed by anyone if stored ‘on-chain’ or on the blockchain without encryption. Even with encryption, there are performance and cost concerns on storing, updating and retrieving the data 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 Geth instance to be run by a third-party. Another issue was handling messaging between users that would best preserve their anonymity and allow for plausible deniability. Since messages can 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 to encrypt/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, the messages 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 key to having been transmitted to the recipient for there to be plausible deniability.

***C. Software Architecture***

The client uses Geth, an ethereal client, to run a node and to serve as an interface to interact with the Whisper network. The front end consists of a web application built on Node.js and was chosen due to the abundant support for web3.js, the Ethereal JavaScript API. This makes it

possible to make calls through the local Ethereal client using the JavaScript API on either the backend or frontend to interact with the Ethereal and Whisper network. The application UI is rendered using React.js and the frontend state management and JavaScript API requests are handled by Redux.js. The user data is stored locally.

Details of all of the parts are covered below:

***1) Geth:***the command-line interface for running a Go implementation of a full ethereal node.

***2) Node.js:***a JavaScript runtime environment used for building and executing event-driven, scalable applications.

***3) React.js:***a JavaScript library used for building user interfaces.

*4) Redux.js:* a JavaScript library used for managing application state.

*5) Application*

***a) Account Management:***included various functions such as creating accounts and adding/removing friends.

***b) Messaging****:* allows the user to send encrypted messages to peers in the Whisper network.

***6) Web3.js:***is an Ethereal JavaScript API used to interact with the Ethereal and Whisper Network through a local Ethereal or Whisper node.

***7) JSON-RPC****:* is a light-weight, stateless remote procedure calls (RPC) protocol.

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

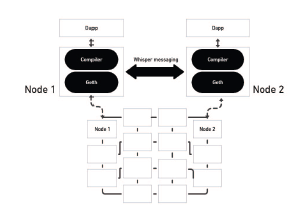
****

Fig. 3.1 Software network architecture.

***D. Operation***

Both the sender and recipient should have a generated asymmetric key pair. The sender initially transmits a message which is asymmetrically encrypted with the recipient’s public key. The message contains a randomly generated symmetric key and a partial Topic for each participant in the chat. Topics are probabilistic filters known as bloom filters used to partially classify the message without compromising security. Nodes can filter for messages that are probably meant for them using the partial topic.

After the recipient decrypts the message using their private key. The recipient verifies the identity of the sender and then retrieves the symmetric keys in the message. The receiver subscribes for messages with the first partial Topic and the sender subscribes for messages with the second partial Topic. The sender encrypts outgoing messages using the first symmetric key while the receiver encrypts outgoing messages with the latter symmetric key. After successfully sharing the symmetric keys and partial Topics, the sender and receiver can send each other secure messages with plausible deniability as all the participants have access to the encryption keys. Besides, the message number is included in each outgoing message to properly order messages and check whether any messages are missing. Upon completion of the session, the symmetric keys are deleted and the corresponding subscriptions are discontinued.

The process of establishing a session and messaging between two parties is displayed in the following figure:

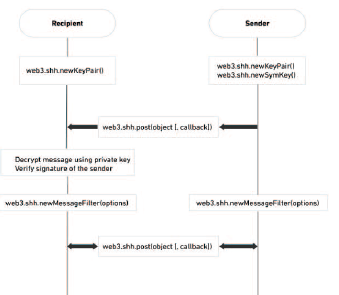
****

Fig. 3.2 Message transmission process.

### FUNDAMENTAL OF BLOCKCHAIN

The blockchain network can be described as a data structure used to create the ledgers that contain a lot of information related to the transaction [6]. As it has for centuries, commerce relies on trust and verified identity with the cryptography protocol module embedded in the system to make sure the credibility of the data and the other security manners. The timestamp as shown in Figure 2 is used in digital documents to prevent the tamper-proof by the attacker. The block in the blockchain is like a seal, if the attacker tries to break the seal, everyone allows to know the action. Each owner transfers the coin to the next by digitally signing a hash of the previous transaction and the public key of the next owner and adding these to the end of the coin [7]. The hash is produced by running contents of the block in question through a cryptographic hash function e.g. Bitcoin uses SHA-256. An ideal cryptographic hash function can easily produce a hash for any input, but it is difficult to derive the input.

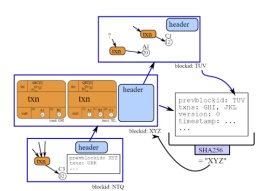
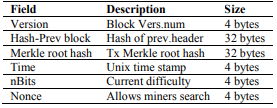


Fig 2. The information a block (transaction)

Table 1. Bitcoin block header format



In various sectors, blockchain is being used to build some purposes such as financial registries, operational registries and one of the most popular ones is smart contracts: a. Financial registries: cryptocurrencies such as Litecoin, Bitcoin, and Dogecoin can be used as an alternative for the real currencies in the blockchain system. b. Operational registries: blockchain allow tracking and certification of specific products or assets, including renting contracts, land registers and notary deals or votes. c. Smart contracts (automated actions on the blockchain): is account holding objects and contain several code functions to make decisions, store data and send the cryptocurrencies to the next owner. The smart contract can execute the code (self-executing). Proof-of-work in bitcoin, proof-of-stake and so on are various consensus protocols used to keep the blockchain secure [8]. It depends on the consensus protocol; the blocks are created and added to the blockchain differently. In proof of work, blocks are created by a procedure called mining, which keeps the blockchain safe. A probability of finding nNonce of proof H for given target T is:



The disadvantage of proof-work is related to efficiency that wastes too many computational resources to find the target value (hash puzzle). The hash in PoW begins with many zero bits hash (SHA-256) and involves kind of scanning for value when hashing a data.

**RESILIENT OVERLAY NETWORK**

An overlay network capable of delivering content, applications, and services to a global audience is a large distributed system [9]. Chord algorithm in overlay network provides a fast-distributed computation of hash function mapping keys to nodes responsible for them. It uses consistent hashing [10] which has several good properties. With high probability, when a Nth node joins (or leaves) the network, only an O(1/N) fraction of the keys are moved to a different location, this is the minimum necessary to maintain a balanced load [11]. Fig. 3 shows a possible three-layered software structure for a cooperative mirror system. The highest layer would provide an interface to users, including names.

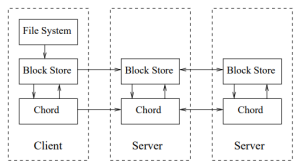


Fig 3. Chord-based distributed system

As shown in Fig. 3 a basic structure of the chord algorithm from client to server. From the client's side, there is a block such as a file system, block store and the chord itself. Whilst, from the server, there are block store and chord that connected to another server and the client. The main usage of the chord protocol is a query value [8] from a client to find a successor (k). This refers to an O(N) query time, and the N is referred to the number of rings applied. In our system model, the chord-based distributed system is used to know the location of the node among the neighbors in the decentralized trading system.

**CHAPTER-4**

**BLOCKCHAIN**

## BLOCKCHAIN ARCHITECTURE

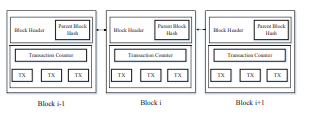
****

Fig. 4.1: An example of blockchain which consists of a continuous sequence of blocks.

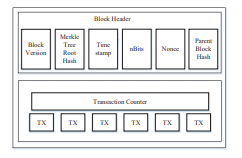
****

Fig.4. 2: Block structure

A Blockchain is a sequence of blocks, which holds a complete list of transaction records like conventional public ledger [14]. Figure 1 illustrates an example of a Blockchain. With a previous block hash 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 the ethereal Blockchain [15]. The first block of a Blockchain is called genesis block which has no parent block. We then explain the internals of Blockchain in detail.

1. **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 validation rules to follow.

(ii) Merkle tree root hash: the hash value of all the transactions in the block.

(iii) Timestamp: current time as seconds in the universal time since January 1, 1970.

(iv) NBits: target threshold of a valid block hash.

(v) Nonce: a 4-byte field, which usually starts with 0 and increases for every hash calculation (will be explained in detail in Section III).

(vi) Parent block hash: a 256-bit hash values that point 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. The Blockchain uses an asymmetric cryptography mechanism to validate the authentication of transactions [13]. Digital signature based on asymmetric cryptography is used in an untrustworthy environment. Next, briefly illustrate the 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 digitally signed transactions are broadcasted throughout the whole network. The typical digital signature is involved with two phases: signing phase and verification phase. For instance, a 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 Blockchains is the elliptic curve digital signature algorithm (ECDSA) [16].

**C. Key Characteristics** **of Blockchain**:

In summary, Blockchain has the 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 in the cost and the performance bottlenecks at the central servers. In contrast to the centralized model, the third party is no longer needed in Blockchain. Consensus algorithms in Blockchain are used to maintain data consistency in a 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 are included in the Blockchain. Blocks that contain invalid transactions could be discovered immediately.

**• Anonymity:** Each user can interact with the Blockchain with a generated address, which does not reveal the real identity of the user. Note that Blockchain cannot guarantee the perfect privacy preservation due to the intrinsic constraint.

**• Audit ability:** Bitcoin Blockchain stores data about user balances based on the Unspent Transaction Output (UTXO) model [2]: Any transaction has to refer to some previous unspent transactions. Once the current transaction is recorded into the Blockchain, the state of those referred unspent transactions switches from unspent to spent. So transactions could be easily verified and tracked.

**D. Taxonomy of Blockchain systems** Current Blockchain systems is categorized roughly into three types: public Blockchain, private Blockchain, and consortium Blockchain [17]. In the public Blockchain, all records are visible to the public and everyone could take part in the consensus process. Differently, only a group of pre-selected nodes would participate in the consensus process of a consortium Blockchain. As for the private Blockchain, only those nodes that come from one specific organization would be allowed to join the consensus process. A private Blockchain is regarded as a centralized network since it is fully controlled by one organization. The consortium Blockchain constructed by several organizations is partially decentralized since only a small portion of nodes would be selected to determine the consensus. The comparison among the three types of Blockchains is listed in.

• **Consensus determination:** In the public Blockchain, each node could take part in the consensus process. And only a selected set of nodes is responsible for validating the block in the consortium Blockchain. As for the private chain, it is fully controlled by one organization and the organization could determine the final consensus.

• **Read permission:** Transactions in a public Blockchain are visible to the public while it depends when it comes to a private Blockchain or a consortium Blockchain.

• **Immutability:** Since records are stored on a large number of participants, it is nearly impossible to tamper transactions in a public Blockchain. Differently, transactions in a private Blockchain or a consortium Blockchain could have tampered easily as there is only a limited number of participants.

• **Efficiency:** It takes plenty of time to propagate transactions and blocks as there are a large number of nodes on the public Blockchain network. As a result, transaction throughput is limited and the latency is high. With fewer validates, consortium Blockchain and private Blockchain could be more efficient.

• **Centralized**: The main difference among the three types of Blockchains is that the public Blockchain is decentralized; the consortium Blockchain is partially centralized and the private Blockchain is fully centralized as it is controlled by a single group.

• **Consensus process**: Everyone in the world could join the consensus process of public Blockchain. Different from the public Blockchain, both consortium Blockchain and Private Blockchain are permission. Since the public Blockchain is open to the world, it can attract many users and communities are active. Many public Blockchains emerge day by day. As for the consortium Blockchain, it could be applied to many business applications. Currently, Hyper ledger [18] is developing business consortium Blockchain frameworks. Ethereal also has provided tools for building consortium Blockchains [19].

**The Rise of Blockchain Identity Solutions**

Over the past decade, Blockchain technology originated from Bitcoin [37], which has raised a lot of attention due to the ability to eliminate intermediaries in transactions. The Blockchain trend of thought, especially from Ethereal [38] in 2014, has spread to many realms from monetary and finance to governance, copyright, and even IoT. As cybersecurity threats are becoming major challenges in IoT [39], Blockchain technology is emerging as a prominent perspective to develop IoT security solutions in decentralized and trustless environments. Blockchains can remove the intermediaries, and allow users and devices to manage their own identities without relying on third parties.

**4.1. Blockchain Technology**

Blockchain technology, which was initially introduced by the Bitcoin crypto-currency, opens opportunities to multiple initiatives and research topics in the context of IoT security. Moreover, Blockchains, like distributed ledgers, keep permanent records of all transactions used to transfer bitcoin values between members, participating in the Bitcoin peer-to-peer network. They also define the structure of how transactions should be organized into blocks, mined, confirmed and stored. Mining is the mechanism that allows the Blockchain to be decentralized and secure. Nakamoto introduced the concept of Proof of Work (PoW) as a mining process to ensure consistency of transactions and solve the double-spending problem in decentralized networks. With the PoW Blockchain, however, there is no need for any kind of a trusted authority, such as a bank, to keep track of the money transfer, all members have their tamper-proof copy of the Blockchain ledger.

Each node in the Bitcoin peer-to-peer network maintains a copy of the Blockchain. Besides, the Blockchain is simultaneously updated through the peer-to-peer network so all members can validate any transaction instantly. Since 2014, Blockchain entered the 2.0 era led by Ethereal, which is a decentralized platform based on Blockchain technology. It aims at creating a general-purpose decentralized computer via the Turing-completeness smart contract concept, which allows writing and deploying all kinds of decentralized applications (Dapps) without any possibility of downtime, censorship or fraud. Buterin explained Ethereal as: “combining the cryptographic algorithms with the economic incentives to create a decentralized network with memory.” To sum up, Blockchain provides us a new perspective to reconstruct the Internet in distributed P2P networks without any unnecessary intermediaries. It is the concept of eliminating central authorities and intermediaries through Blockchain technology that opens opportunities for multiple initiatives and research topics in the context of IoT security. Therefore, we provide a survey of how Blockchain could transform identity management systems.

**4.2. Elucidation of Identity and Naming Systems**

The distinction between identity management systems and naming systems (i.e., Domain Name

Service (DNS), active directory or URLs) is blurry in the context of the Internet. At first sight, there is a slight difference between these types of systems. Identity management systems are coupled with service providers and used to identify resources or users in a particular domain, whereas the naming systems are designed to identify computers across networks, resources and user accounts in companies or social networks. DNS records, URLs and user accounts are somehow identities at the same time. Blockchains are already used to build identifier or naming systems. The Namecoin [43], for example, is a fork of the Bitcoin Blockchain that provides domain naming systems functionality via binding the human-readable name and IP address. It is the first solution for the naming trilemma of the Zooko’s Triangle [44] on building a secure, decentralized and human-meaningful naming system. By modifying Namecoin, Certcoin [45] builds a decentralized authentication system (PKI), which defines a set of key operations like registering, updating, verifying and revoking. Based on Certain, Authcoin [46] proposes a new alternative protocol for authentication using a flexible challenge-response schema to PGP in the context of the Web of Things. Its successor, the Block stack [47] attempts to redesign the naming system and PKI authentication features using state machines.

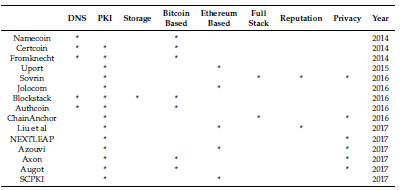
It also added the storage aspect to its Blockchain-based system to construct a kind of new type of Internet resource identification, preserving privacy and including property rights. Knecht et al. [48] tuned the certain parameters to ensure the retention of identities where users could not register the same already-registered identity again. Even though naming systems could be exploited as identity providers for individuals in specific domains, the goals of identity systems and naming systems are completely different. The former attempts to find a way to uniquely define individuals in cyberspace, whereas the latter is responsible for routing via assigning a unique identifier to retrieve a user or object in the service domain. Rather than working on naming systems, many research teams and recent projects as introduced in the following part, are working on the digital identity problem based on Blockchains. These Blockchain-based identity solutions are promising of becoming the critical component of future digital world infrastructures.

**4.3. Blockchain-Based IdMS**

From an academic research perspective, Blockchain-based IdM systems are gaining a lot of attention to propose new solutions for digital identities: Bassam [49] introduced a Blockchain-based PKI and implemented the solution based on Ethereal smart contracts. In his work, he defined several identity-related operations such as adding attributes, signing attributes and revoking signatures. More importantly, he also calculated the cost of different operations in the Ethereal platform. Liu et al. [50] developed an identity management system based on Ethereal smart contracts through binding public key and user’s entity information. Besides the identity management part, they also redefined the token to fit their proposed reputation model to reflect the reputation of users. Axon [51] analyzed privacy requirements when designing decentralized PKI systems and proposed a Blockchain-based PKI with privacy awareness. In addition to a set of operations such as registering, revocation and recovery, they introduced the concept of the neighbor group to enhance the performance of privacy-preserving. Augot et al. [52,53] modified the Bitcoin stack to build an identity management solution and introduce a zero-knowledge proof called Brands selective disclosure scheme [54] to ensure the anonymity of the identity at the same time. Hard Jono [55] introduced a Blockchain-based privacy-preserving identity solution called Chain Anchor using zero-knowledge proof in a permission Blockchain environment.

In Chain Anchor, verified nodes have the privileges to write or process transactions and others could only read and verify transactions. All verified nodes are built on the tamper-resistant hardwires and form the privacy-preserving layer to provide privacy protection services to users. Halpin [56] designed NEXT LEAP, a decentralized identity framework with privacy-preserving features using blind signatures. Moreover, they used authentication services provided by their identity solution to build a more secure messaging application. Azouvi et al. [57] also proposed a privacy-preserving identity solution using blind signatures. They set up a threat model, performed a security analysis and implemented their solution in Ethereal. In Table 2, we chronologically list these Blockchain-based identity management solutions, most of which are from academia, and give the main features of them.

Table 1. Blockchain identity management solutions.

****

Besides, several startups and IT players are focusing on the development of identity systems,

such as Uport [58], Shocard [59], Bitnation [60], Civic [61], Jolocom [62], Sovrin [63], Evernym

Identity Standard ERC725/735 [66], and W3C Decentralized Identifiers (DIDs) [67] just to mention a few. Generally speaking, these Blockchain identity solutions fall into one of two categories: identity solutions relying on permissionless public Blockchain platforms (e.g., Uport

And ERC725/735) and identity solutions having authenticated block producers of the permission identity Blockchain (e.g., Sovrin). For example, Uport, which is a core component of the Consensus Ethereal ecosystem [68], aims at building decentralized applications to solve the digital identity problem. It mainly uses the ethereal smart contract to design the digital identity model and ensures reliability and usability of identities through a set of operations (i.e., keys revocation and identities recovery). Sovrin takes a different approach and provides a complete full-stack to manage identities from the distributed ledger to devices. It adds the identity layer for every entity on the Internet and operates as a global public utility designed to provide permanent, private and trustworthy identities.

Sovrin establishes a public permission Blockchain in a peer-to-peer network in which nodes are divided into authenticated validate nodes and observer nodes to ensure high performance and scalability. More importantly, the Sovrin token is introduced to their framework as their incentives to power their transactions. In general, the Blockchain-based identity is also called the self-sovereign identity, which denotes an approach that transfers access control rights and management of identities from traditional identity providers to the edge under the control of identity owners. In other words, only owners have the right to dispose of their identities, which blocks attacks from malicious third-party identity providers. Although there are two different methodologies, permission (e.g., Sovrin) and permissionless (e.g., Uport), to implement the Blockchain self-sovereign identity solutions, the basic concepts could be summarized as Identities of individuals (i.e., human beings) and collectives (e.g., companies, banks, governments, etc.) can be selectively stored in the Blockchain without compromising privacy.

* Individuals and collectives issue claims to each other using these Blockchain identities. Claims are the endorsement by other individuals or collectives, which could be governments, banks, universities or even friends.

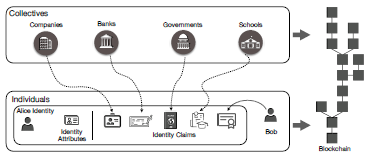
As shown in Figure 2, there are two individuals (Alice and Bob) and several collective examples

(i.e., companies, banks, governments, and schools). The individual or collective identity, composed of the identity attributes and identity claims, could be gradually completed through the following steps:

* The individual or collective, for instance, Alice, could generate and add as many identity attributes (identifier, public and private key pairs, biometrics, etc.) as she wants.
* The individual or collective will create the Blockchain identity by submitting identity-related information such as public keys and the corresponding signatures.
* The individual or collective could use the public and private keypairs, which are correlated to the mined Blockchain identity, to issue claims.

In Figure 2, Alice could self-generate identity attributes and also receive the claims from her

employer, bank, government, school and even her friend Bob. In different scenarios, Alice could give the necessary identity claims to identify herself or to demonstrate that she has some qualifications. For example, when applying for a job, Alice can give the ID Card from the government and diploma from her university. After entering the company, she has to give her a bank account to receive the salary.

****

**Fig4.3** Overview of Blockchain based identity management solutions.

**CONSENSUS ALGORITHMS**

In the blockchain, how to reach consensus among the untrustworthy nodes is a transformation of the Byzantine Generals (BG) Problem, which was raised in [20]. In BG problem, a group of generals who command a portion of Byzantine army circle the city. Some generals prefer to attack while other generals prefer to retreat. However, the attack would fail if only part of the generals attacks the city. Thus, they have to reach an agreement to attack or retreat. How to reach a consensus in a distributed environment is a challenge. It is also a challenge for blockchain as the blockchain network is distributed. In the blockchain, there is no central node that ensures ledgers on distributed nodes are all the same. Some protocols are needed to ensure ledgers in different nodes are consistent. Next to present several common approaches to reach a consensus in the blockchain.

In POW, each node of the network is calculating a hash value of the block header. The block header contains a nonce and miners would change the nonce frequently to get different hash values. The consensus requires that the calculated value must be equal to or smaller than a certain given value. When one node reaches the target value, it would broadcast the block to other nodes and all other nodes must mutually confirm the correctness of the hash value. If the block is validated, other miners would append this new block to their blockchains. Nodes that calculate the hash values are called miners and the PoW procedure is called mining in Bitcoin. In the decentralized network, valid blocks might be generated simultaneously when multiple nodes find suitable nonce nearly at the same time. As a result, branches may be generated as shown in Figure 3. However, it is unlikely that two competing forks will generate the next block simultaneously. In PoW protocol, a chain that becomes longer thereafter is judged as the authentic ones. Consider two forks created by simultaneously validated blocks U4 and B4. Miners keep mining their blocks until a longer branch is found. B4, B5 forms a longer chain, so the miners on U4 would switch to the longer branch.

**Advances on consensus algorithms**

A good consensus algorithm means efficiency, safety, and convenience. Recently, several endeavors have been made to improve consensus algorithms in the blockchain. New consensus algorithms are devised aiming to solve some specific problems of blockchain. The main idea of PeerCensus [33] is to decouple block creation and transaction confirmation so that the consensus speed can be significantly increased. Besides, Kraft [34] proposed a new consensus method to ensure that a block is generated in a relatively stable speed. It is known that high blocks generation rate compromise Bit coin’s security. So the Greedy Heaviest-Observed Sub-Tree (GHOST) chain selection rule [35] is proposed to solve this problem. Instead of the longest branch scheme, GHOST weighs the branches and miners could choose the better one to follow. Chepurnoy et al. [36] presented a new consensus algorithm for peer-to-peer blockchain systems where anyone who provides non-interactive proofs of irretrievability for the past state snapshots is agreed to generate the block. In such a protocol, miners only have to store old block headers instead of full blocks.

**Selfish Mining**

Blockchain is susceptible to attacks of colluding selfish miners. In particular, Eyal and Sirer [10] showed that the network is vulnerable even if only a small portion of the hashing power is used to cheat. In selfish mining strategy, selfish miners keep their mined blocks without broadcasting and the private branch would be revealed to the public only if some requirements are satisfied. As the private branch is longer than the current public chain, it would be admitted by all miners. Before the private blockchain publishment, honest miners are wasting their resources on a useless branch while selfish miners are mining their private chain without competitors. So selfish miners tend to get more revenue. Based on selfish mining, many other attacks have been proposed to show that blockchain is not so secure. In stubborn mining [48], miners could amplify their gain by non-trivially composing mining attacks with network-level eclipse attacks. The trail-stubbornness is one of the stubborn strategies that miners still mine the blocks even if the private chain is left behind. Yet in some cases, it can result in 13% gains in comparison with a non-trail-stubborn counterpart. [49] Shows that there are selfish mining strategies that earn more money and are profitable for smaller miners compared to simple selfish mining. But the gains are relatively small. Furthermore, it shows that attackers with less than 25% of the computational resources can still gain from selfish mining. To help fix the selfish mining problem, Heilman [50] presented a novel approach for honest miners to choose which branch to follow. With random beacons and timestamps, honest miners would select more fresh blocks. However, [50] is vulnerable to 562 forgeable timestamps. Zero Block [51] builds on the simple scheme: Each block must be generated and accepted by the network within a maximum time interval. Within Zero Block, selfish miners cannot achieve more than their expected reward.

**CHAPTER-5**

**RESULTS AND DISCUSSION**

This section is divided into three parts. In the first part, we review the encryption algorithm used. Secondly, an analysis of 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 of.

**B. Resistance to Denial of Service attacks**

Since every Whisper message is routed to every node that it can reach, the network might be susceptible to two types 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 rating impacts 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 adversary might be able to ascertain the identity of the sender if they control a significant portion of the network. This type of analysis is curtailed by having the sender node generate an appropriate amount of encrypted junk messages. In this way, the adversary would not be able to determine the identity of the sender of a certain message.

**THE MODEL OF DECENTRALIZED ENERGY TRADING**

The prominent example of a trading energy system that uses blockchain is Brooklyn microgrid [12] 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 the blockchain concept. The goal of the system is to measure the ability of blockchain technology adapted to buy and sell the energy among the neighbors and how effective blockchain technology is adopted.

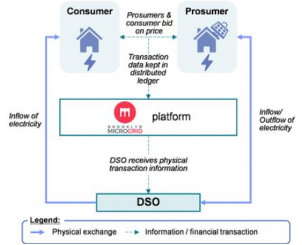


Fig 5. Brooklyn microgrid network

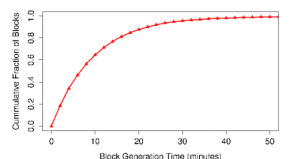


Fig 6. Block generation time in Bitcoin

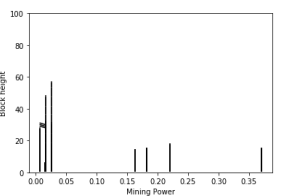


Fig 7. Performance of dishonest miner

Decentralized storage of the transaction on the blockchain system allows keeping a distribution, transparency and secure record of energy transaction between the party which is tamper-proof from the attacker. The decentralized energy trading system would no longer require third-party involvement e.g. banks, energy companies to do the energy trading activities among the traders. Instead, the process will run automatically and the energy will be sent after the miner solving the proof-of-work and propagates to the entire network which in the end the presume will get his/her reward. The router transmits the packet data of record transactions from a source to another computer in the peer-to-peer network. The wireless routers use WPA-PSK and PSK Pass Phrases to connect with other devices and every message is encrypted by using AES (Advanced Encryption Standard). The presume who has surplus energy and wants to sell the energy announces to the network by giving details of the information related to the amount of energy and the price.

The consumers will be able to see that announcement and if the consumers want to do the transaction, it will execute by the miners via smart contract. After a transaction is broadcast to the Bitcoin network. When that happens, it is said that the transaction has been mined at a depth of 1 block. With each subsequent block that is found, the number of blocks deep is increased by one. To be secure against double-spending, a transaction should not be considered as confirmed until it is a certain number of blocks deep. Cumulative distribution function (CDF) of the block based on figure 6, approximately 30% of Bitcoin blocks take between 10 and 40 minutes to be generated [13]. In the selfish mining attack, an attacker tries to find a new block by solving the proof-of-work puzzle and keep the block secret and doing mining continuously till they reach the longest chain on the blockchain network. The selfish chain publishes its secret block if only the honest network comes close to their secret network or when the selfish chain wants to claim the unfair rewards. It will affect the rational miner to join in the selfish mining pool. The rational miners are preferred to join the pool with the highest revenue.

The selfish miner is relying on power mining (resources) as shown in Fig. 7 and always competes with the honest miner to find a new block. Once their network becomes the longest, the selfish miner will easily to invalidate the valid block from the honest miner. The current assumption of the Bitcoin system is safe as long as 51% of the mining power is under the honest miner, but Eyal and Sirer [14] show the attacker can gain the unfair revenue with 25% hashing power. The components model in the network architecture performs a key role in supporting the continuity of a system and it has the input and output gateway of the data blocks. In charge of distributing and replicating the data record of a transaction in the peer-to-peer network, the component requires knowledge of data blocks location, as well as the ability to fetch data blocks in the appropriate node that contains the requested data and return to the requester.

## CONCLUSIONS

The concept of blockchain technology is used for trading the renewable energy system in an environment among the neighbors in the peer-to-peer network. We discussed a model for trading energy in a small environment by using blockchain 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 each company and across market participants. Blockchain turns both currencies and commodities into a digital form without relying on middleman which allows one person to trade with another. For the future, the strategy is needed 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 sufficient POW. However, since the mail server would not know whether the recipient had received the message it would have to continue doing so indefinitely while new messages would continue being added. It is unlikely that any mail server could be capable of managing the appropriate POW and 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 anonymous legitimate request by the recipient, there is an issue with storage and incentivizing storage by the participants in the network. One possible implementation could be a ‘social storage’ scheme that would utilize the user’s friends to store messages that correspond to a certain Whisper topic. However, this could potentially result in the recipient’s identity being discovered by an adversary if no mechanism to randomize the topic regularly is put in place.

# 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?" PwC Global Power and Utilities, 2016.

[3] G. Karame, E. Androulaki, "Bitcoin and Blockchain Security", Information Security and Privacy Series, United States of America, Artech House, 2016.

[4] P. Danzi, A. Marko, C. Stefanovic, and P. Popovski, “Distributed Proportional-Fairness Control 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 Managing Decentralized Energy Production and Consumption", IEEE International Conference on Environment and Electrical Engineering (EEEIC/ I&CPS Europe), 2017.

[6] S. Nakamoto, “Bitcoin: A Peer-to-peer Electronic Cash System”, 2008.

[7] P. Narayan, “Building Blockchain Projects: Develop realtime DApps using Ethereum and JavaScript”, Birmingham-Mumbai, Packt Publishing Ltd. 2017.

[8] C. Troncoso, M. Isaakidis, G. Danezis and H. Halpin, Systematizing Decentralization and Privacy: Lesson from 15 Years of Research and Deployments, Proceeding on Privacy Enhancing Technologies: 307-329, 2017.

[9] K.S. Ramesh, M. Kasbekar, W. Lichtenstein, J. Manish, “Overlay Network: An Akamai Perspective”, University of Massachusetts, Amherst, Akamai Technologies Inc\*

[10] Chord (peer-to-peer): <https://en.wikipedia.org/wiki/Chord>

[11] S. Ion, R. Morris, D. Karger, M. Frans Kaashoek, H. Balakrishnan, “Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications”, SIGCOMM’01, San Diego, California, USA, 2001.

[12] Brooklyn Microgrid, Available online at <https://www.brooklyn.energy/>

[13] Average Confirmation Time in Bitcoin: <https://blockchain.info/charts/avg-confirmation-time>

[14] I. Eyal, and E.G. Sirer, “Majority is not Enough: Bitcoin Mining is Vulnerable”, Financial Cryptography and Data Security, Berlin, 2014.