Journal of Ambient Intelligence and Humanized Computing (2022) 13:3903–3921
https://doi.org/10.1007/s12652-021-03660-5

ORIGINAL RESEARCH

Simorgh, a fully decentralized blockchain-based secure communication system

Ebad Mirzaei[1] · Massoud Hadian Dehkordi[1]

Received: 29 April 2020 / Accepted: 15 December 2021 / Published online: 3 February 2022

Abstract

Every industry, government and even culture has been affected by the recent progress of modern technology. Communication and the communication industry are no exception to this. Most communication systems operate in a centralize manner; this is in conflict with information security and privacy. These centralize systems cannot offer completely transparent, trustworthy, fast and uninterrupted communication. Decentralized communication systems can be secure and transparent, designed with the aid of blockchain technology. Blockchain technology is currently used extensively in communication systems, but none of them have provided a completely decentralized system for communication. We are looking to design a secure and decentralized data exchange network. In this paper We present Simorgh. Simorgh is a completely secure, decentralized system based on blockchain technology with the ability to exchange data and efficiently in the field of smart contracts in today's world. To overcome some of the limitations of file saving, The InterPlanetary File System (IPFS) technology has been used in this new system. We also discuss and analyze features and potential advantages of this system. Assuming that the Simorgh blockchain is identical to Ethereum blockchain, a smart contract is simulated.

Keywords Blockcahin · Distributed system · Information security · IPFS · Smart contract

1 Introduction

Nowadays, the growing trend of communication, as an inevitable and important aspect of human life, has led to the emergence of digital communication. The world of communication in modern life allows people to store and collect large amounts of data, resulting in attracting a great deal of attention from many governments, departments, central systems, and the like. However, world powers have controlled the tremendous power of storage in today's world. On the other hand, the security of personal and organizational information is of great importance in this era. Nevertheless, the solutions suggested to establish security lack sufficient transparent and are mostly concentrated, indicating the lack of sufficient security. The use of decentralized systems can be a safe, transparent, and suitable solution, among which blockchain technology, as a promising strategy, is regarded as one of the most important techniques in decentralized systems.

Blockchain technology provides the possibility for users to properly and completely control their digital identities and carry out their transactions in a secure, transparent, and safe way. Blockchain technology-based communication systems benefit from asymmetric cryptography, consensus-based algorithms, and peer-to-peer network structure.

There are currently many communication structures (such as Khacef and Pujolle 2019; Sarıtekin et al. 2018; Nizamuddin et al. 2018) and messenger softwares (such as E-chat[1] and CrypViser[2]) that use blockchain technology; But if we examine them, we will see that they either do not have the complete comprehensiveness for data exchange between sender and receiver as a data exchange system, or if they are comprehensive, they have definitely used centralized structures in one or more sections. This will cause some users to distrust them. This research seeks to introduce a completely decentralized communication system (Simorgh[3]), which is

✉ Massoud Hadian Dehkordi
mhadian@iust.ac.ir

[1] School of Mathematical Sciences, Iran University of Science and Technology, Tehran, Iran

[1] https://echat.io.

[2] https://ico.crypviser.net.

[3] Simorgh (/sɪˈmɜːrg/; Persian: سيمرغ, also spelled Simurgh, simorg, simurg, simoorg, simorq or simourv) is a benevolent, mythical bird in Iranian mythology and literature that helps good people and heroes. It is sometimes equated with other mythological birds such as a phoenix.

based on blockchain technology and utilizes encryption and IPFS technology.Simorgh is a comprehensive data exchange system that has all the features of a communication structure used for data exchange.

1.1 Motivation

Blockchain has been of great interest to engineers and investors because of its immense commercial potential and its use in applications as diverse as a cryptocurrency. Blockchain based Messenger softwares has grown very well, with aimed to employ security and transparency. These softwares are based on three principles including asymmetric encryption, consensus mechanisms, and blockchain. Decentralized messengers use peer-to-peer networks to communicate between nodes. In addition, there are softwares which uses blockchain and IPFS technology but cannot communicate and used as a messenger.

As far as we know, bitcoin is the first emergence of blockchain technology, and most efforts in the field of blockchain-based systems were initially has been made using Bitcoin's Blockchain.

BitDNS was developed as a domain name following suggestions on creating a decentralized service based on Bitcoin. BitDNS later took the form of the first altcoin: Namecoin (Loibi 2014). Although the security of blockchain data are computationally guaranteed on Bitcoin, it needs all of the users to make breaking changes so it can adopt another function. There was no specific method in Bitcoin transactions for a data capacity payload or other transmission data. Moreover, the blockchain technology is thriving so fast as to require increasingly more network bandwidth and storage space, outdating the Bitcoin blockchain for use in messenger software.

As the problem-solving expanse of peer to peer systems reaches more modern problems (more than just client-to-server issues), newer problems begin to appear. Issues regarding building the trust relationship within P2P networks and using blockchain for developing a messenger software are examples of these problems.

Decentralized applications can be performed on platforms of Ethereum, Cardano and a few similar blockchains. These are called smart contracts. Each of these platforms has only one identifier as their address. The ownership of these identifiers is fixed; therefore, it is possible to track and analyze their processes.

It is possible to define a smart contract as an executable code running on top of the blockchain. Using smart contracts, an automatic, non-obstructive executive agreement is possible between two parties.

In Simorgh, unlike some communication structures and messenger softwares, there is no centralized part in processes in sending data such as authentication, receiver routing process in the network, data storage process, etc. This is why Simorgh is a comprehensive and decentralized system for exchanging data.

Our work describes the system of using blockchain, smart contract and IPFS for data transfer, ensuring data tracking, personal message certification and security, and creating the storage space needed to save data based on blockchain and IPFS technology. Simorgh has applications that are not visible in many communication structures and messenger softwares. Simorgh, for example, uses smart contracts, which increases Simorgh's comprehensiveness and efficiency.

1.2 Contribution and organisation of paper

In this article, we explain how to design secure messaging using blockchain. We suggest a plan whereby the sender can send an encrypted message to the recipient end-to-end so that he or she is confident and knows that no individual or organization can withhold confidentiality, integrity and accessibility of his or her message. IPFS technology has been used to eliminate the centralization of memory storage.

The main contribution of this paper is:

- A new secure messaging architecture based on a blockchain.
- Introducing a framework for the formation of decentralized communication structures.
- Providing a way to eliminate centrality using authentication nodes and blockchain technology.
- Using smart contracts to exchange messages and express the capabilities of these contracts in today's world.
- Using decentralized storage to store data based on IPFS technology.

The next section presents the fundamental and primary definitions, including blockchain definition, IPFS technology, and smart contracts, which are required for designing the Simorgh system. The third section discusses other messaging systems and related topics. The fourth section explains the principles of the Simorgh system. After eliminating possible ambiguities, the sixth analyze the features and outline the advantages of this system. Finally, Sect. 6 provides a brief conclusion to the topic.

2 Fundamental

2.1 Blockchain technology

The word blockchain indicates a chain or string of blocks. A block of the blockchain is a list or string of continuous and fixed records. The blockchain is a string of connected blocks which is secured by cryptography. In addition, blockchain consists of a hash value, which refers to the previous block. As an open and distributed ledger, blockchain can permanently record transactions between parties in an effective, reliable, and secure (Benet 2014).

The blockchain design cannot be manipulated and changed. Moreover, a peer-to-peer network manages the blockchain, in which nodes are collectively connected to a protocol to validate a new block. In general, all previous blocks will change if a valid block is changed, and the next block cannot be changed without changing the previous block. The consensus protocol controls this process, which requires the communication and agreement of the majority of the network on a shared decision (Benet 2014).

Blockchain was first proposed as a solution for the Byzantine Generals problem (Karaarslan and Akbaş 2017). This technology is designed fully secured and is considered as a paradigm of a distributed, quick, and transparent computing system. Some of the various applications of blockchain include identification and accounts management, transaction records investigation, document resource management, food traceability, voting systems, and insurance industry (Vukolić 2015). Nowadays, blockchain technology is widely used for design cryptocurrencies. In this technology, the authentication process is conducted by using asymmetric cryptographic principles and techniques (ZKEDg 2018). In cryptocurrencies, a public wallet has an address, which is a public key or a derivative of it and obtained by a private key of encryption functions, as well as is only available for wallet owner. The public key can be used in privacy features to execute key exchange protocols in order to establish a secure and encrypted connection. Further, the private and public keys can be used in integrity features to transactions signing and signature verification (Nakamoto 2008).

However, blockchain technology fails to provide all the features of a security system into one system, and it is only suitable for the environment of networks, which require complete reliability and authentication (Karame 2016).

2.1.1 The blockchain structure as system-level integritry

In Biktimirov et al. (2017) the blockchain is expressed structurally and with a more scientific perspective. According to Biktimirov et al. (2017) maintaining the chain features that contribute to paraphrase of integrity is why blockchain technology is developed (and has its current form). One of these features is continuity, which refers to the sequence of the blocks (i.e. links) in the blockchain that has been specified while the blockchain was forming. The other feature is reliability. Reliability exists when it is not possible to make any changes in the placement of a link in the chain.

A blockchain can be defined as an integrity at system level when it is comprised of separate parts (links), which can be divided into components called atoms. These atoms have different types; they all have names to identify their characteristics, have specific features, and when concerned with the computer system, are either passive or active.

Atoms are not used in this paper and are merely used to explain the structure of the blockchain according to Biktimirov et al. (2017).

The following are types of atoms:

1. The first and the last atoms of the blockchain (at its borders); the first atom being the end of the previous link and the last atom referring to the beginning of the next.
2. Structural atoms, which are concerned with the vector of identifiers of atoms and the logical continuation of the first and last atoms.
3. Data atoms; they are comprised of passive link components interpreting the data.

There are several categories of data atoms, including:

- Open data atoms, which consist of unprocessed data and are characterized by their length.
- Integral data atoms. The length of fixed-integrity data and the identifier of or reference to the data integrity control procedure characterize these atoms.
- Signed data atoms. The length of signed data, the identifier of or reference to the data signature check procedure, and the reference to the signature check key serve to describe this type of atoms. A signed data atom can become a subject atom; it is also possible to define the signature check procedure outside of and relative to blockchain.
- Encrypted data atoms. The length of the encrypted data, the identifier of or reference to the data encryption or decryption procedure and to the encryption or decryption key are their descriptive characteristics.
- Signature atoms. They include the signature of one or several signed data atoms with (a) preset identifier (-s); and
- Hash atoms. They include the integrity control standard for one or several integral data atoms with (a) pre-set identifier (-s).

4. Subject atoms. They are comprised of active link components interpreting the data. There are several of subject atoms, including:

   - Scenario atoms. They include an interpretable code to process data atom.
   - Executor atoms. They have a compiled code for processing a blockchain in a real processor, a computer system hypervisor, or an atom machine.

- Machine atoms. They provide the environment suitable for the executing scenario or executor atoms.

2.1.2 Consensus protocol

The fundamental technology in the blockchain system that makes decentralization possible is distributed consensus protocol. This protocol eliminates the need for a central authority to allow a unified transaction ledger, on which a general consensus exists. Message passing and locale scale decision making at each node are determined by this protocol. Changing the design of the consensus protocol has a significant effect on the scalability, transaction capacity, fault tolerance, and other performance parameters of a blockchain system.

The Bitcoin network is based on implementation of Nakamoto consensus protocol (Nakamoto 2008), which protected Bitcoin against double-spending attacks in an unreliable decentralized peer-to-peer (Croman et al. 2016) network.

Nakamoto protocol, following the continuous expansion of the Bitcoin network, is faced with constrictions in maintenance and performance. According to blockchain specialists, the following are problems, in particular regarding to the proof-of-work (PoW) mining mechanism, that need to be considered in Nakamoto consensus.

1. Unsustainable energy consumption.
2. Low trans-action capacity and poor scalability.
3. Long-term security concerns as mining rewards diminish.

There are thousands of nodes in the present Bitcoin network and the highest number of transactions in the network is 7 transactions per second (TPS). The most possible transactions without harming the security of the protocol are 25 per second, which can be achieved by tuning protocol parameters (Croman et al. 2016). As of the end of 2019, one Bitcoin transaction requires as much electricity as the one-day average of 21 households in North America (29 November 2019).

Compared to the Bitcoin, there are millions nodes and the possibility of about 65,000 transactions per second (VISA Fact Sheet 2019) in the VISA network.

These disadvantages in PoW mining have led blockchain specialists to consider alternative block proposing mechanisms with lower computation requirements, e.g., proof of stake (PoS), proof of authority (PoA), and proof of elapsed time (PoET). These mechanisms can help lower energy costs and use.

It is sometimes possible to build trust among different nodes by employing cryptography techniques to allow for more organized block proposing schemes (e.g., round-robin and committee-based block generation). Other essential factors in consensus protocol are the right incentives for making participants confident in the blockchain network. As a result, block proposing schemes usually come with new incentives that enhance fair participation and, subsequently, further the sustainability in general.

Peercoin (King and Nadal 2012), Bitcoin-NG (Eyal et al. 2016), Ourosboros (Cardano) (Kiayias et al. 2017), Snow White (Daian et al. 2019), and EOSIO (IO 2017), and POA Network (Network 2018) are examples of widely used blockchain consensus protocols that include such ideas.

2.2 IPFS technology

InterPlanetary file system (IPFS) is a decentralized and peer-to-peer file distribution system which allows users to have a better experience in using the Internet. Nowadays, Information, videos, photos, and etc. are stored centralized on the internet. Google and Amazon companies give their servers to users for store and share their information. Therefore, these companies can increase or decrease the quality and speed of stored files whenever they want (Güncelleme 2016).

IPFS performed content-based storage, despite HTTP protocol which is storage location-based and has more advantages than the previous method. In order to download files using the HTTP protocol, users must have the URL of the site or server in which the image is stored. In another word, the user cannot access the files, if the server is unavailable. IPFS method prevents storing duplicate files because each file has a hash value. Therefore, users can download intended files through the unique hash of that file. Validation can be performed through this hash to make sure the downloaded file is the same, which was not possible in the HTTP protocol (Güncelleme 2016).

IPFS is very similar to the Bit Torrent network because there is no mutual reliability between nodes in both of them. In addition, IPFS uses blockchain protocols and infrastructures. Bitcoin uses these protocols and infrastructures to store unchanged data on blocks, delete duplicate files from the network, and search to find a file or data address. The following is a summary of the benefits of IPFS: (Bernaille and Teixeira 2007; Güncelleme 2016).

- Security: Content-based addressing as well as digital signature data prevents DDOS attacks, while HTTP is highly vulnerable to the attack.
- Bandwidth optimization: The data is stored on decentralized servers, resulting in high bandwidth for users to receive data at high speeds.
- Availability: Data is always available because it is stored in multiple locations.
- Reliability: Reliability of data content regardless of the performance of the nodes that store the data in their memory (Stark 2016).

2.3 Smart contract

Based on stark (Stark 2016), clack stated (Clack et al. 2016) that "smart contract" usually utilized in two concepts as follows:

- The first concept described an operational concept which is related to software factors, but unnecessarily available in the ledger. "contract" term means that these software factors performed definite obligation and certain rights, as well as controlled on certain assets in the ledger. There is no specific consensus in this concept and smart contract definition. There are different definitions (Szabo 1997; Swanson 2014) in this sense. Stark renames these factors as the "smart contract code".
- The second concept focused on how legal contracts were explained and implemented in software structure. Hence, this concept consists of operational aspects, such as how writing and interpreting legal contracts. Currently, ideas and projects are existed which focused on these aspects such as CommonAccord (2016), Legalese (2016), and the Ricardian Contract (Grigg 2004). Stark renames these as "smart legal contracts".

A comprehensive definition of smart contracts is required because the terms of smart contracts lack any specific consensus. Hence, based on the clack, a definition of the smart contract is provided which sufficient generality to both of the above concepts.

A smart contract is an automatable and enforceable agreement. Automatable by computer, although some parts may require human input and control. Enforceable either by legal enforcement of rights and obligations or via tamper-proof execution of computer code.

The above definition is brief and comprehensive, and consist of the legal smart contract and smart contract code. In legal smart contracts, some of the contracts can be implemented by software. In the smart contract code, a contract is fully automated and can be unrelated to a formal contract.

This definition indicated that it must be executable without reviewing any side aspects. Legal smart contracts can be complicated due to legal obligations. Nevertheless, the smart contract code simply implemented this contract because implementing a contract means implementing a computer code.

3 Related work

In this section, we present the works made use blockchain for secure communication between users. Blockchain technology was used industrially and operationally before sufficient academic studies were performed. The first appearance of this technology was proven outside the university, by inventing bitcoin.

After the invention of Bitcoin, researchers tried to design systems to solve industrial problems using the technology provided in the Bitcoin cryptocurrency. After a while, as mentioned in the previous section, the researchers found that the Bitcoin blockchain and its consensus protocol do not indicate much capability for designing these systems. To this end, the researchers decided to design the systems needed by the industry using blockchains and different consensus protocols. Nowadays one of the most important needs in the world is the design of decentralized communication systems.

Peer-to-peer network is defined as a technology for creating connection and an exchange of resources between devices (Li et al. 2016) that does not use a centralized server. A blockchain is defined as a decentralized universal ledger on which records of data are collected. The technology of blockchain is based on peer-to-peer network; the interaction and exchanges between users across the network are direct, and trusted authorities are not required. Instances of applications that benefit from blockchain technology are voting systems, messengers, social networks, applications for predicting markets, etc. The blockchain network records the data on different network locations and devices and does not use a centralized server for this purpose (Chronopoulos et al. 2008).

3.1 Systems and cryptocurrencies

BitDNS paved the way for further approaches. The first system to use blockchain for creating a decentralized system for naming is Namecoin (the first altcoin from bitcoin having its blockchain). Satoshi supported the idea of using an independent blockchain in BitDNS. Satoshi proposed merged mining for the first time for securing blockchain. Additionally, Namecoin created a naming system with these three features all at the same time: decentralized; meaningful to humans; and secure, thus solving the Zooko's triangle. However, Namecoin is weak in computation and is unable to prevent 51% of attacks.

Blockstack serves to eliminate centralization points at the application layer. It utilizes the existing internet transport layer (TCP or UDP) and basic communication protocols, and is limited by the underlying blockchain of Bitcoin.

Certcoin utilizes the blockchain Namecoin as a distributed ledger of domains and their associated public keys to eliminate central authorities. Since Certcoin users all store the entire blockchain, issues of latency for the controller and in the security of merged mining used by Namecoin arise.

Emercoin is a PKI based on blockchain. It doesn't eliminate the central authorities; instead, it utilizes blockchain to store hashes of issued and revoked certificates. It also helps optimize network access through verifying keys and signatures on local copies of the blockchain.

These systems, nevertheless, were all obstructed by the same thing. Solving the PoW requires large amounts of resources, leading this consensus to carry heavy energy consumption, excessive computation time, and delays.

3.2 Papers

Khacef and Pujolle (2019) provides a plan to establish secure, peer-to-peer communications based on the blockchain and IPFS. In Khacef and Pujolle (2019), the authors turned their attention to the security of messaging (e.g., an email, a web site or any type of a message) in blockchain and suggested a protocol for encrypted messaging across the public blockchain network. They aimed to utilize blockchain to store public keys, digital signature, and peer information. This way, it becomes possible for every component to validate information on all other components throughout the network, thus activating the functionalities of PKI.

Their suggested model for reaching their main goal, which is securing the communications throughout the network, involves a blockchain validating the identity of users and maximizing security for the exchange of messages, thus securing trust between the users.

The users' communication should only be with those identities which the smart contract has validated. Any other attempt at communication must be treated as a possible attack.

The first step to communicate on the system is to register the user's identity and public key. The system stores these informations on the blockchain. The blockchain that was employed to produce this system was Ethereum public blockchain. This blockchain uses smart contracts.

A comprehensive communication system with all data storages is absent from this plan, and it also fails to address the issue of the necessity of an intermediary between the sender and the recipient. The presented study solely attempts to use blockchain technology to provide security for other systems.

Vimal and Srivatsa (2019) provides a file sharing scheme based on blockchain technology and IPFS technology. In Vimal and Srivatsa (2019), authors offered a solution based on cryptographic hashes for the expenses of storing on the blockchain. Instead of storing the files on the blockchain, the file hash value is stored in the blockchain. Recognized users can receive a public key of the recipient to encrypt the files they upload on IPFS.

This article only provides a way to store data and secure this blockchain-based storage, and does not provide any explanation of other aspects of a comprehensive communication system.

The proposed HO-Auth scheme in Salim et al. (2021) introduces a novel method for device authorization for existing decentralized telecom networks. According to the authors, each base station is managed by a private blockchain network service provider using a unique ID. The authors try to protect the network against user data theft and the transmission of corrupted data to the blockchain network. HO-Auth implements a deep learning LSTM algorithm to monitor all devices on different network base stations. In this scheme the user profile is created based on the user movement pattern.

Połap et al. (2021) introduces a multi-agent architectural system that is capable of creating Internet of Medical Things (IoMT) solutions that guarantee the security of private data and, at the same time, enable the addition and/or modification of the methods implemented for classification. The suggested system benefits from its incorporated/integrated blockchain elements and threaded federated learning. The location of each individual element is on the agents of exchange information. The study claims that their proposed system is able to provide better solutions for the Internet of Medical Thing as it offers a novel multi-agent system for differentiating of/classifying tasks (like security) and, therefore, minimizing the operation time and improving precision.

The Internet of Medical Things (IoMT) has been shown to possess promising capabilities in various applications. Its practicality is of particular importance in certain medical center and, more importantly, in the whole healthcare network that is a widespread phenomenon in decentralized industries of today (Al-Turjman et al. 2020; Yaacoub et al. 2020). IoMT has numerous different features to study, including its architecture, and data security, and its method of processing categorizing medical research results. One of the architectural systems proposed for IoMT are agent-based solutions (Mostafa et al. 2018; Allioui et al. 2020; Chakour et al. 2020) that have attracted the attention of researchers. The objects constructed in an agent-based system impact information and events in their environment. In fact, the several agents present in these systems generate solutions that can help achieve the current objectives of modeling information exchange and communication mechanisms.

In Zhou et al. (2018), a blockchain solution is proposed that is highly credible for applications in medical insurance storage system. Shen et al. (2019) proposes a layered blockchain-based system that offers an unusual structure for transactions and includes blocks of stored vectors obtained from images. The construction of blockchain determines its security. Meng et al. (2019) represents a trust management based on blockchain and examines its security in the face of insider attacks. The investigated models are only some of the selected examples from the patient data security blockchain models. Other solutions

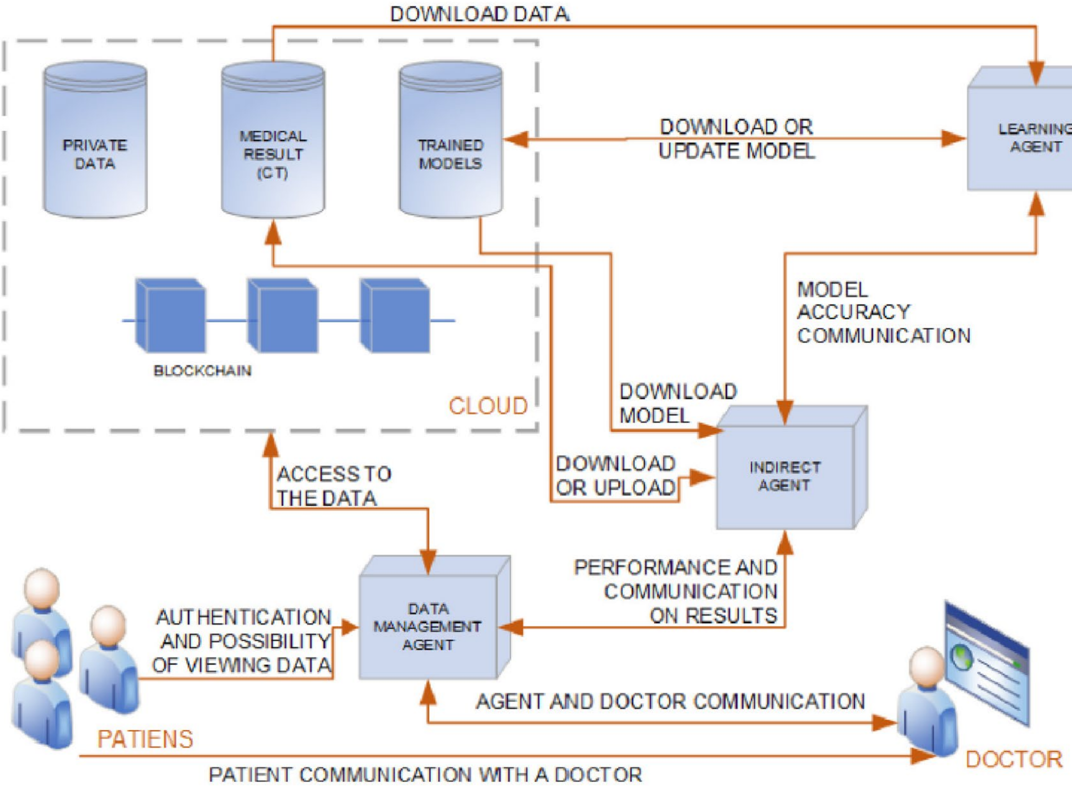


Fig. 1 Visualization of the operation of the proposed system in Połap et al. (2021)

can be found in Jamil et al. (2019), Du et al. (2020) and Lu et al. (2020) describes another blockchain-based decentralized routing registration system.

Połap et al. (2021) follows the two main studies of the authors, namely (Połap et al. 2020; Yazdinejad et al. 2020). Yazdinejad et al. (2020) introduces a network based on blockchain that can perform distributed analyses for medical applications. The groundwork for this study is presented in Połap et al. (2020). The agent architecture introduced in Połap et al. (2021) incorporates machine learning solutions with selected data sharing policies or models of trained classifiers and the security of user data in the blockchain.

The following are the primary suggestions of this study: (1) a new IoMT architecture that incorporates a multi-agent system (MAS) with separated security and data processing; (2) new agent models for medical applications in which it is possible to assign specific tasks to agents units; (3) agent-based design in combination with federated learning according to the number of threads, allowing for parallel training of classifiers and to categorize their weights for achieving a classifier architecture; (4) a consortium mechanism for the classified data from multitude machine learning solutions according to the theory of soft set, and; (5) using the capabilities of blockchain in MAS to share and protect the private data.

3.2.1 Architecture of Połap et al. (2021)

To design an agent-based architecture for IoMT, agents are created and connected to cloud infrastructure. Cloud infrastructure makes the database available to various agents in several geographic areas. In the present study, it is suggested to apply this method by setting a learning agent on a number of devices connected to medical databases, all the while observing data confidentiality—all databases are connected through unique IDs.

There is a deep and correct connection between Simorgh and the plan presented in Połap et al. (2021), and in practice Połap et al. (2021) was considered a special mode in the implementation of Simorgh, and this is a true sign of Simorgh's efficiency. As mentioned above, this system is designed in the field of IoMT and is not designed to exchange messages independently. In fact, this architecture can be considered as the customization of Simorgh system. This system uses cloud storage instead of decentralized storage. Also, in this system, the data management agents, which is responsible for verifying the authenticity of agents, operates in a centralized manner, and this is a weakness for it. This system does not have many capabilities of Simorgh, but it is possible to create all the capabilities of this architecture in Simorgh by customizing Simorgh (Fig. 1).

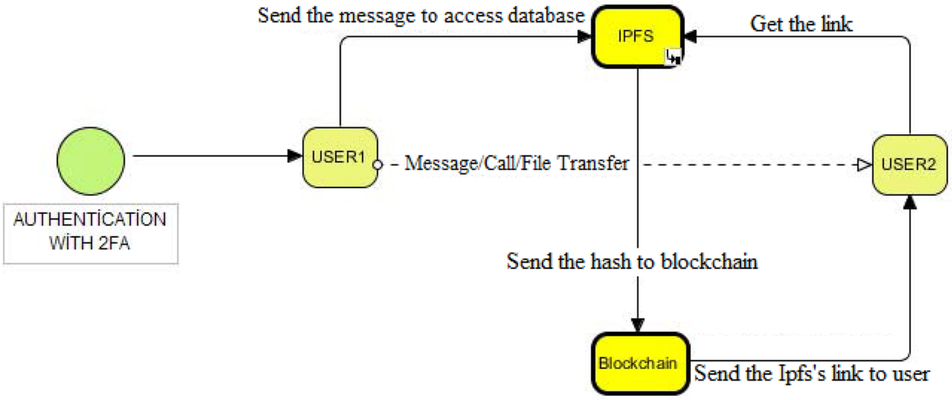

Send the message to access database
IPFS
Get the link
USER1
Message/Call/File Transfer
USER2
AUTHENTİCATİON WİTH 2FA
Send the hash to blockchain
Blockchain
Send the Ipfs's link to user

Fig. 2 Cryptouch communication system architecture

3.3 Messenger softwares

Messenger software has grown very well, with aimed to employ security and transparency. These softwares are based on three principles including asymmetric encryption, consensus mechanisms, and blockchain (a chain of blocks). Decentralized messengers use peer-to-peer networks to communicate between nodes. In addition, there are softwares which uses blockchain and IPFS technology but cannot communicate and used as a messenger. (For Example: Filecoin, Uport, Ujomusic).

The messenger E-chat software utilizes blockchain and IPFS technologies and peer-to-peer network simultaneously. When a user sends a message in this messenger, all data is stored in a decentralized and blockchain-based data storage network, if connecting to a peer-to-peer network and online access is not necessary. In another word, the data URL (message or file) is directly stored in IPFS to send data to the blockchain-based system.

CrypViser is another software which uses end-to-end encryption and authentication algorithms. Furthermore, the decentralized distribution of public keys is created on the blockchain. This software can prevent any manipulating, tracking, and executing the MITM attack at all levels of communication by presenting a cryptographic key ID.

The main application of E-chat and CrypViser softwares is sending and receiving cryptocurrencies; meanwhile they have messaging capabilities. Cryptouch messenger was designed by a team of researchers (Sarıtekin et al. 2018), which communicates by IPFS and blockchain technology. In this messenger, people can only communicate with each other on the same network. Therefore, it just works for the private network and cannot be used as a messenger on a public network. In the present study, the presented idea in Sarıtekin et al. (2018) and Nizamuddin et al. (2018) were combined to design Simorgh. In the following, both of the presented ideas in Sarıtekin et al. (2018) and Nizamuddin et al. (2018) are described briefly.

3.3.1 Cryptouch

Simorgh has taken its idea from Sarıtekin et al. (2018) in many ways. Recep Ahmet Saritekin et al. introduced a messenger named Cryptouch in Sarıtekin et al. (2018). This blockchain-technology-based messenger was designed using IPFS technology. Understanding the structure, communication process, and message sending process of Cryptouch is beneficial to understand the communication process of Simorgh. In the following, a summary of the presented study was explained.

Cryptouch is a distributed-design messenger software in which the blockchain and IPFS technologies have been used. This software aims to meet the communication, social, personal, and intra-company needs. The private or authorized blockchains can be used to implement this messenger.

3.3.1.1 The main goals of Cryptouch

- Create a distributed, open source, secure, transparent, uninterrupted, free messaging software for users.
- Ability to exchange messages, file transfers, voice and video call and user-to-user notification systems using the new IPFS technology.
- Create a stable, secure environment with other extra facilities on the network where Cryptouch is used.

The communication process in Cryptouch is illustrated in Fig. 2.

3.3.1.2 Cryptouch architecture and design

1. First, users log in to their account in the software and it is authenticated under a two-step authentication or 2FA.
2. The user then sends the data to the IPFS memory and notifies the recipient to receive the data from the IPFS.
3. In the next step, the data link and hash value are placed in the blockchain.

4. Then data link in IPFS memory is sent to the recipient.
5. The recipient receives its associated data from IPFS.

This software is designed so that only the users within a network can use it. The most obvious application of this messenger is used in private networks. In Cryptouch, creating, sending, and receiving messages are done based on the private and decentralized backbone. Companies usually employ private networks in which filtering and E2E security processes are much easier.

3.3.2 Process of IPFS-blockchain-based authenticity of online publications

There are some defects in Cryptouch, including its centralized identification center and its exclusive design for private networks. These defects have been eliminated in Simorgh. The idea used in Simorgh has been taken from the solution presented by Nizamuddin et al., in their article (Nizamuddin et al. 2018). They introduced a solution to authenticate the blockchain and IPFS-based online books and publications. In the following, a summary of the present study was explained. The present study was provided in three sections including problem description, goals, and proposed solution to provide a comprehensive summary.

- Problem statement: Nowadays, there is not any way to authentication and integration of online digital content. In addition, the sent and received digital contents may have changed or their authors' identities may have been forged. Furthermore, there is no valid, reliable, open-source, and decentralized way to verify and authenticate the publication history of these digital contents—publication history includes all the information about the different versions of the digital contents from the first to the last and about the changes in the versions and who applied them. Today, a published book by a publication may be re-published again. So, different versions of the original books are produced, and there is not any solution to stop that. In addition, the validity of a digital document can't be verified at any time because publishers are not accountable for the published literature.
- Goals: The main goals of this article are as follows:
- Providing an IPFS-blockchain-based solution and framework for authenticating digital content and published online books.
- Providing decentralized data storage and management over storage, tracking a highly integrated content over the original, and other published versions of the book.
- Presenting and examining the architecture and design of the system
- A summary of the proposed solution: The proposed solution is based on the Ethereum blockchain and smart contracts, as well as uses IPFS technology for the centralized and distributed storage of digital contents (electronic books and multimedia files). The data existing in this storage can be accessed and tracked by everyone using their Hash value. The Hash value is used in the smart contracts of the Ethereum blockchain framework to ensure integration, originality, and validity. The hash value remains unchanged until a change (even a little one) occurs in the contents of the document or electronic book. The Hash value in the IPFS of the document or electronic book changes and doesn't match the one saved in related smart contracts when any change occurs in the publication process. Therefore, verifying and authenticating the publication history of a document or book based on its Hash value can be a reliable way to ensure the legality (in terms of the author) of the document or book.

3.3.2.1 System architecture and design Figure 3 depicts the overall system architecture and design for automating the online books authenticity, originality, and integrity using Ethereum smart contracts and IPFS. The proposed solution uses smart contracts to trigger events that are logged to notify the participating parties to keep track of events and transaction details.

1. The author first creates a smart contract.
2. The main publishing house then asks the author to prove the validity of the writings during the execution of the contract.
3. The author then requests a grant from the publishing house.
4. By executing the contract until this stage, the main publishing house obtains the permission to publish the desired literature.
5. The main publishing house then uploads the desired electronic file to IPFS.
6. At this stage, the main publishing house requests a grant from the reader and the reader must pay the grant.
7. After paying the grant, the reader accesses the IPFS memory and downloads the desired file.
8. By continuing to implement the smart contract, other publications will request the publication of the desired article from the main publications.
9. By receiving a grant during the execution of the contract the main publishing house will allow other publications to publish the file.
10. Other publications upload the file to IPFS only with their own access.

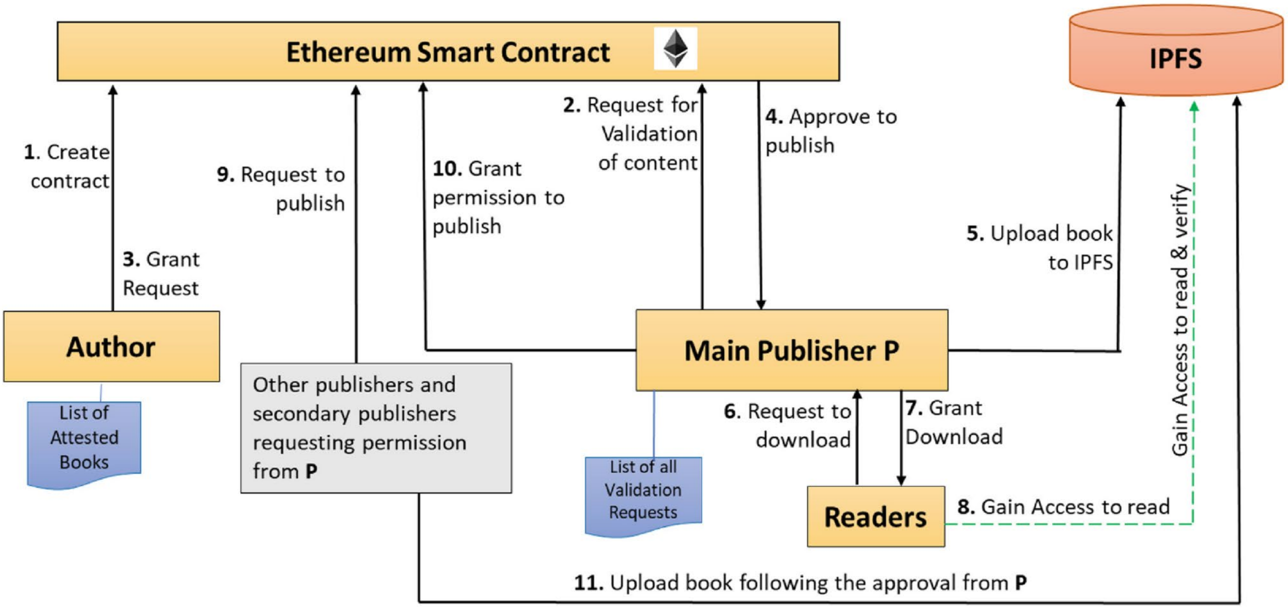

Ethereum Smart Contract
IPFS
1. Create contract
3. Grant Request
9. Request to publish
10. Grant permission to publish
2. Request for Validation of content
4. Approve to publish
5. Upload book to IPFS
Gain Access to read & verify
Author
List of Attested Books
Other publishers and secondary publishers requesting permission from P
Main Publisher P
List of all Validation Requests
6. Request to download
7. Grant Download
Readers
8. Gain Access to read
11. Upload book following the approval from P

Fig. 3 Process of IPFS-blockchain-based authenticity of online publications

4 Simorgh system design principles

Simorgh can be converted into a public messenger which can be used in public networks, although it is a comprehensively decentralized software. This system used smart contracts in which these contracts not only use two-stage authentication but also can implement the required terms and conditions by the sender and receiver. In this system, messages are encrypted by end-to-end and safe encryption methods. All the blockchains equipped with an acceptable consensus mechanism and capability to provide smart contracts can be employed in Simorgh.

The main goals of designing this system are:

- Providing a secure, transparent open source communication system that can provide uninterrupted communication to the users who utilize it.
- User-to-user message transfer—transfer video, audio, and so on without a centralized data server.
- Providing an environment for continuous, secure, convenient and simple communication for users.
- Ability to authenticating of message and authentication of sender and receiver by authentication nodes
- Taking advantage of using smart contracts to determine the terms and conditions for a submitted data

The messaging architecture of this communication system is shown in Fig. 4.

4.1 Outline of the process of sending the message

1. Before sending, the sender must find the blockchain addresses and IP addresses of trusted authentication nodes and the blockchain address of the recipient.
2. The sender on the blockchain platform (capable of executing smart contracts) creates a smart contract with authentication node. During this contract, they prove their authenticity by implementing an authentication cryptographic protocol (based on secure authentication protocols). The sender sends the cryptocurrencies required for the contract.
3. The sender sends data to authentication node using their IP address.
4. Then, during the execution of the contract, after authentication of the parties to the contract, the public key assigned to the sender ought to be sent to the authentication node and the authentication node's public key should be sent to the sender.
5. The parties then sign the hash of the message and send it to the contract, and if this value is the same, the authenticity of the message is confirmed. Then a block of data (including hash signature messages—sender address—recipient address—authentication node address) is placed in the blockchain. Further provisions may be made in this contract as stated in the following section.
6. After authentication of the message, sender and authentication node and sending certain data to the blockchain,

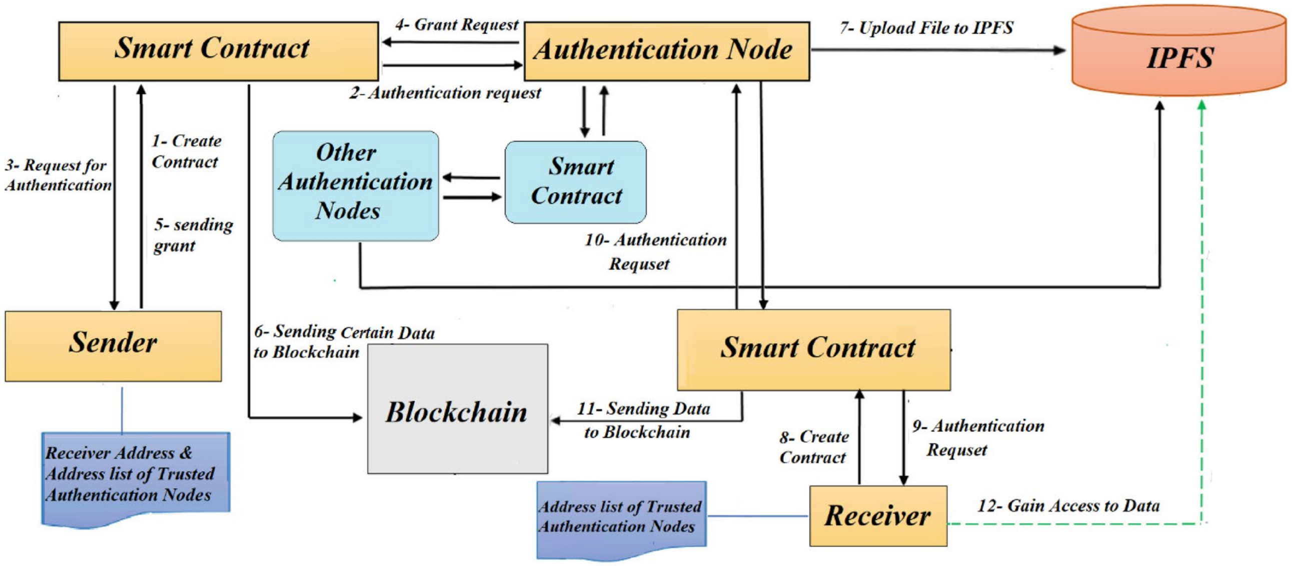

Smart Contract
4- Grant Request
Authentication Node
7- Upload File to IPFS
IPFS
2- Authentication request
3- Request for Authentication
1- Create Contract
5- sending grant
Other Authentication Nodes
Smart Contract
10- Authentication Requset
Sender
6- Sending Certain Data to Blockchain
Smart Contract
Blockchain
11- Sending Data to Blockchain
8- Create Contract
9- Authentication Requset
Receiver Address & Address list of Trusted Authentication Nodes
Address list of Trusted Authentication Nodes
Receiver
12- Gain Access to Data

Fig. 4 Simorgh communication system architecture

this message is placed in the IPFS memory by authentication node.

7. In order to receive the file by the recipient, a smart contract is executed again between the recipient and the authentication node, during which authenticity of the recipient and the authentication node are verified. The recipient then checks the data for the message in question in blockchain. If the data in the blockchain is exactly consistent and there is no tampering with the data, the recipient receives this message from authentication node or receives permission and links to access the IPFS memory from authentication node.

4.2 Explanation of authentication protocol and end-to-end encryption

- Authentication protocols: One of the points to pay close attention to is the use of secure protocols to authenticate users (nodes) and authentication nodes. This authentication is done by smart contracts. These authentication protocols are based on public key cryptography and digital signatures. To prevent MITM attack, the hash value of each user's public key is placed in the blockchain with that user's signature. We could describe these authentication protocols here, but in Simorgh it is intended that users can use their own authentication protocols. But their data sent to the blockchain must follow the general format, for example, the maximum length of data allowed to be sent to the blockchain, and so on. However, a general authentication protocol can be designed for other users and authentication nodes. In our opinion, algorithms based on elliptic curve play an important role in the implementation of these protocols. (We have also designed a protocol which we have not mentioned here.)
- End-to-end encryption: The sender encrypts his data using the recipient's public key, and since the hash of this public key with the recipient's signature on it is in the blockchain, there is sufficient assurance of its validity, and without a doubt the only person who can read the data is the recipient and no one else can read the data.

5 Disambiguation

The explanations in the previous section have created some ambiguities undoubtedly. For example, what is the node of authentication? What is its precise task and role? Does the existence of this node lead to security breaches? What are the other terms that can be performed in smart contracts? Why is used the IPFS technology? In the following, some explanations were presented to eliminate such ambiguities.

5.1 The reason for using IPFS technology

The most crucial advantage of Simorgh is that it uses IPFS technology in association with blockchain technology. The blockchain technology can't store a large volume of data, and IPFS technology is a great option to overcome this limitation. An essential point about IPFS is that everyone can access IPFS as a public database and no organization has complete control over it.

Any uploaded file in the IPFS network has a unique Hash value which allows the network to stop sending a copy of a file into the network and, the repetitive files are deleted if a copy is sent. In addition, it provides the ability to check the history and different versions of a file, as well as prevents the easy deletion of an older version which connected to the current one. Also, the speed of upload and download is much higher because IPFS is a distributed database. The advantages as mentioned earlier, have caused a great, powerful, and quick database for saving files in the IPFS technology of Simorgh.

5.2 Authentication node

These nodes [as described in Nizamuddin et al. (2018)] have access to IPFS storage. In Nizamuddin et al. (2018), a centralized authentication center is existing which indicates the partial centrality of the system. In Simorgh, there is no sign of centrality, and this system is thoroughly designed for decentralized public networks.

This system has two significant advantages compared to Nizamuddin et al. (2018) due to the existence of authentication nodes. First, the authentication process in Nizamuddin et al. (2018) is done by a centralized authentication center but that in Simorgh is done by decentralized authentication nodes. Second, the system presented in Nizamuddin et al. (2018) is a message transmission system and it needs a real connection between a node and IPFS memory. Therefore, a server is required to meet the needs such as the need to a DNS server or inform the receiver about starting a connection. In addition to the mentioned needs of a public system, authentication nodes can eliminate the centrality in the authentication of nodes.

Authentication nodes are not exclusive for specific individuals or groups, and everyone can be an authentication node. Even a person can be a sender with an address and play the role of an authentication node with another one (providing access to IPFS). Everything in Simorgh is based on the reliability proving; it means that nodes start a connection with an authentication node only providing which they trust. This truth can be created through smart contracts. According to Nizamuddin et al. (2018), the author trusts a publication through an imperative contract, many groups, organizations, and companies can introduce a trusted node to their users to connect with as the authentication node and use this authentication node to exchange data.

Sometimes, a publication signs contract with another like Nizamuddin et al. (2018). Then, an authentication node may send data to another through a smart contract and the receiver gets data from the second authentication node. It can happen if the sender has not taken the permission and possibility of this data exchange from authentication node. Based on this feature, people can trace the hash value of a message in the blockchain to know which authentication nodes have intended data. Then, they receive the data through one of the trusted nodes if they want and realize the security requirements.

5.3 Applications of smart contracts

The smart contracts can be used to transparent authentication because the smart contract cannot be modified and canceled. In these contracts, the sender/receiver and node must confirm their identity to each other. Then, the massage authenticity is investigated by checking the hash of the sent message and verifying its signature. Hence, a smart contract can be used by both of them, and they can set conditions for each other.

The execution guarantee of this contract is possible in two ways. First, the blockchain platform can be equipped with cryptocurrencies, and the authentication node assigns some cryptocurrency to the contract for guarantee. Second, there must be public addresses, which a message containing a violation document of an authentication node is sent to a public address by another node if an authentication node violated. Furthermore, sending a message to this address means sending a message to the public. This information is stored in blockchain so that the address of the violated authentication node is recorded with the message hash containing the violation document. Finally, this makes the address of the violated authentication node unreliable for the public.

The terms and conditions of these contracts have not any limitation. For example, the sender can request a reliable authentication node to assign this data to a node with a specific address whereas another one cannot access this data. In addition, the authentication node can request the cryptocurrency from the sender in return for its performance if the cryptocurrency is designed on that blockchain. Similarly, various terms and conditions can be defined.

Simorgh system must be implemented on a platform which is equipped with cryptocurrency to indicate all of its unique features.

5.4 A brief overview of attack analysis

It should be noted that, as stated in most protocols and structures, the public key can be used to exchange private keys, and the private key can be used to encrypt the transmitted data. The point to mention here is that Simorgh is a structure and due to its decentralization there is no requirement to use the same password algorithm for all users and any group or even two users can use their own public key and private key algorithms.

The important point here is that it is not difficult to study and analyze the attacks on Simorgh. To analyze attacks on Simorgh, keep in mind that as mentioned above, all groups can have their authentication nodes and as a result they can use their own public key and private key encryption algorithms. They can even use their own key exchange protocol for key exchanges. Therefore, it is not possible to execute a specific attack on these algorithms and protocols.

To execute an attack on Simorgh, the Simorgh blockchain or its overall design structure must be attacked. Considering that our proposal for designing Simorgh is to use the proof of stake protocol, attacking the Simorgh blockchain is equivalent to attacking the security of this structure (proof of stake) for the consensus of the distributed ledger.

The attack on Simorgh design structure can be analyzed in several ways. If an authentication node is malicious, it is easy to identify it by placing the data information in the blockchain and examining the data by the sender and receiver. If a node intends to steal data, it must be stated that the data will be encrypted end-to-end, and also the authentication protocol in the smart contract will prevent the data from being stolen and received by a malicious node. In addition, if an attacker is intended to replace the fake data with the original data, the recipient can be notified of the fake data by examining the information received in the blockchain, and consequently MITM attack is not possible in Simorgh.

In the same way, the implementation of any destructive process in Simorgh can be detected by using the Simorgh blockchain. In fact, the security of Simorgh depends entirely on the security of the blockchain and its algorithms and protocols, and as long as the blockchain or its algorithms and protocols have sufficient security, Simorgh will be safe and there is no way to attack it, and in order to run a malicious process the attacker must break the security of the algorithms or cryptographic protocols and also target and destroy the security of the blockchain or consensus algorithm.

6 Analysis and discussion

In this section we talk about the features and advantages of Simorgh system. There are two aspects to this analysis. We will first describe the features of this system and then present its benefits.

6.1 Simorgh's features

All governments, non-governmental organizations, companies and organizations, students, the commons or any other group of people can use Simorgh. Some of the features of this communication system that can be of interest to those interested are:

1. Fully private message exchange: In this system any group of people or organizations can form a private network on this system and so they will be able messaging totally private.
2. File sharing: Easily any company, group, etc. can share their files. It can also act as a cloud and allows you to access your data anywhere.
3. Using cryptocurrencies: As mentioned earlier, smart contracts are used in this system and these contracts can also be used to raise economic benefits using cryptocurrencies. Smart contract can be applied to asking money from sender node for sending messages as well.
4. Data broadcasting and group communications: In Simorgh, each group can identify and introduce addresses as public addresses of the whole group or specific subgroups, so that each node in that group can see the messages sent to that particular address as a message sent to them.

6.2 Simorgh's advantages

Utilizing the features and capabilities of the Simorgh system will bring many benefits to users. These benefits include:

1. Security: In this system, for the sake of complete transparency, any misuse of data and impersonation cannot be regarded as cyber fraud but as a breach of cryptographic principles which is practically impossible. For example, an attacker intends to breach data privacy. Meanwhile, the sender and receiver use symmetric cryptography with AES-256 function for privacy and also exchange keys through Diffie–Hellman key exchange and smart contract. In this case, the attacker has to either solve the discrete logarithm problem or find a way to infiltrate the AES-256 function algorithm. In both scenarios, it can be concluded that data privacy security (in terms of confidentiality) in Simorgh is higher than either of AES-256 encryption and Diffie–Hellman key exchange protocols.
2. Resistance to the data manipulation: Data manipulation is not possible or very difficult due to message authentication of the sender and receiver, authentication node, and record all security information in Blockchain. Data manipulation and forgery indicate the loss of integrity in encryption terms. Suppose a subgroup of nodes is being implemented in Simorgh and in order to provide integrity, RSA-2048 and SHA-256 functions are used to create a digital signature. If data manipulation occurs, it means one of the functions, either RSA-2048 or SHA-256 has been forged. Therefore, data security in Simorgh is at least equal to the security of these functions.
3. Increased trust: In Simorgh due to the use of distributed ledger and consensus mechanism, the reliability of the verified data increases. Unreliable mediums (such as large servers of large messaging services that users are forced to use) always generate a sense

of insecurity. In Simorgh, on the other hand, the user is not forced to trust a particular medium and they can use the reliable medium (authentication node) of their choice (although it is possible to develop large reliable servers in Simorgh for some users; for example, WhatsApp can create an identity verification node on Simorgh).

4. Transparency: Transactions and processes are obvious to everyone due to using smart contracts and blockchain technology, which leads to high transparency in the Simorgh system.
5. Earning money: Since authentication nodes are decentralized and anyone with access to IPFS can act as an authentication node, so any authentication nodes can ask for money (cryptocurrency) for offering services.
6. Ability to add encryption algorithms and protocols: Since sending and receiving data in this system (due to the use of smart contracts) is completely separate from the data encryption process, so any organization, group, etc. can use their own encryption in this system.
7. Reliability: Since data in Simorgh are stored in multiple locations due to IPFS technology, the consensus protocol only allows data to be changed when all parties agree on it, and thus the reliability for the consensus becomes very high.
8. More resistance to DDOS attacks: There are various techniques to prevent DDOS attacks, but all lack adequate security. Security against DDOS attacks can be enhanced by increasing the distribution and decentralization of databases. Simorgh will be highly resistant to DDOS due to the use of blockchain technology and decentralization of all databases (IPFS).
9. Entrance ability of organization for profit: Organizations or government can deliver service and processed data to a group of people. To this end, it can send raw data and material of a particular activity or operation to an authentication node. Then, this node processed raw data and material using its facility and, finally, send the results to receiver people and make a profit.
10. Establishment of private networks: Any organization or group can set up a completely private network for messaging. To this end, the organization manager acts as an authentication node and does not serve anyone other than his/her employees. His staff are trusted only to him/her and are only related to him.
11. High availability and speed: Sometimes, a server slows down due to high user traffic. In such cases, the distribution of a server can prevent slowdown in all parts and some parts of the server can continue working properly. It should be noted that the accumulation of all servers in one place leads to several problems (e.g., in heating/temperature and configuration of the servers) and the only solution is distributing these servers across different locations (and using different IP for transfering data traffic). In Simorgh, due to the use of IPFS, data is stored in multiple locations instead of being stored in one place, which facilitates and speeds up information access.
12. Reducing costs: Since Simorgh is a decentralized, blockchain-based system, this is likely to reduce the cost of performing or confirming a particular process.
13. A platform to all social networks: All social networks can use this system because it is a public system and not limited to private networks. For example, Whatsapp can perform as a reliable authentication node for its users by providing memory and accessing to IPFS memory. Furthermore, the users of this social network only used this authentication node to send and receive data.

6.3 Comparison with other systems

Table 1 shows a comparison of the capabilities of the Simorgh system in comparison with other systems and softwares. Vimal and Srivatsa (2019) is merely a file-sharing system, and Khacef and Pujolle (2019) is not a stand-alone system, but is used to secure other systems based on blockchain, and Vimal and Srivatsa (2019) has no application in peer-to-peer communication; However, we have assumed these three systems as independent communication systems and examined the desired capabilities for them.

7 Implementation

Simorgh is a comprehensive system that can be implemented in various dimensions, from a small company to a global scale. Due to its comprehensiveness, it can encompass several other exchange systems which can, in turn, be customized according to their functions.

Implementation of Simorgh requires careful determination of algorithms, protocols, and the incorporated blockchain platform. A new design of all these components, however, is too expensive; therefore, it is advisable to use the existing protocols and algorithms and only create a new, exclusive platform. Here, the available platform Ethereum is used to demonstrate the practicality of Simorgh.

Here we suppose that Simorgh and Ethereum have the same blockchain and their blockchain's structures are implemented in the same way. We implemented and checked the smart contract by Remix IDE.[4] The implementation process will be discussed in detail focusing especially on the participants' proper interaction and functionality in the system.

[4] http://remix.ethereum.org.

Table 1 Comparison with other systems

Capabilities	Simorgh	Cryptouch	(Nizamuddin, Hasan et al. 2018)	(Khacef and Pujolle 2019)	(Vimal and Srivatsa 2019)	echat	Crypviser
Message Exchange	✓	✓	✗	✓	✗	✓	✓
File sharing	✓	✓	✓	✗	✓	✓	✓
E-to-E encryption	✓	✓	✗	✓	✗	✓	✓
MITM resistance	✓	✓	✓	✓	✓	✓	✓
Run as a public network	✓	✗	✓	✓	✓	✓	✓
Using Smart Contract	✓	✗	✓	✓	✓	✗	✓
Decentralize Datacenter	✓	✓	✓	✗	✓	✓	✗
Decentralize Authentication center	✓	✓	✗	✗	✗	✗	✗
Using blockchain	✓	✓	✓	✓	✓	✓	✓
DDOS resistance	✓	✓	✓	✗	✓	✓	✗
Formation private sub-network	✓	✓	✗	✗	✗	✗	✗
Platform for cryptocurrency	✓	✗	✗	✗	✗	✓	✓
Data broadcasting	✓	✓	✓	✗	✓	✓	✓
elimination of central management in all parts	✓	✗	✓	✗	✗	✗	✗

The numerous powerful features in Remix IDE may help in checking and, if needed, debugging the contracts before their implementation.

7.1 Implementation details

The programming language used for writing the code was Solidity on the web browser-based IDE, Remix. The contract includes two entities: sender and authentication node. There is a separate Simorgh address for the two, and they can call the smart contract functions at different timestamps to participate in the contract.

Figure 5 illustrates the message sequence diagram for sending a message from sender to authentication node. It shows the interaction between sender, authentication node and smart contract.

Figure 6 represents the message sequence diagram for receiving a message from authentication node by receiver. Next, we show the important parts of the code of our smart contract. The smart contract code for sending a message from sender to authentication node and smart contract code for receiving a message from authentication node is available at: https://www.github.com/ebad1993/Simorgh-example-contract.

First, after creating a contract between the sender and authentication node, using the address of each party and sender's private key, the identity of both parties is verified by the smart contract.

In order to express this example more precisely, using the concept presented in 6 April 2016 (which is the implementation of RSA digital signature verification), the authentication node verify the authenticity of sender by verification of sender digital signature. Below is the constructor of

Fig. 5 Smart contract between sender and authentication node

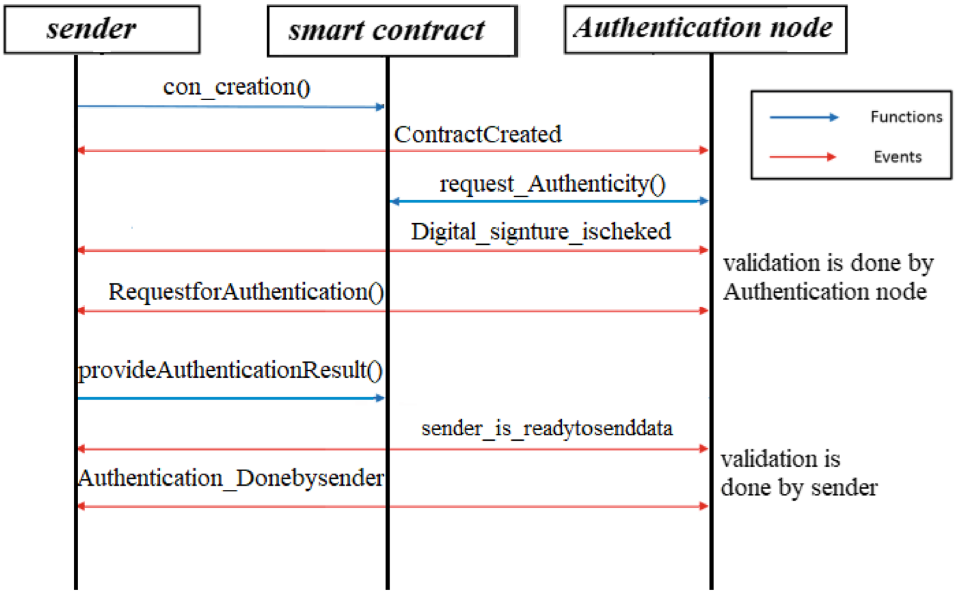
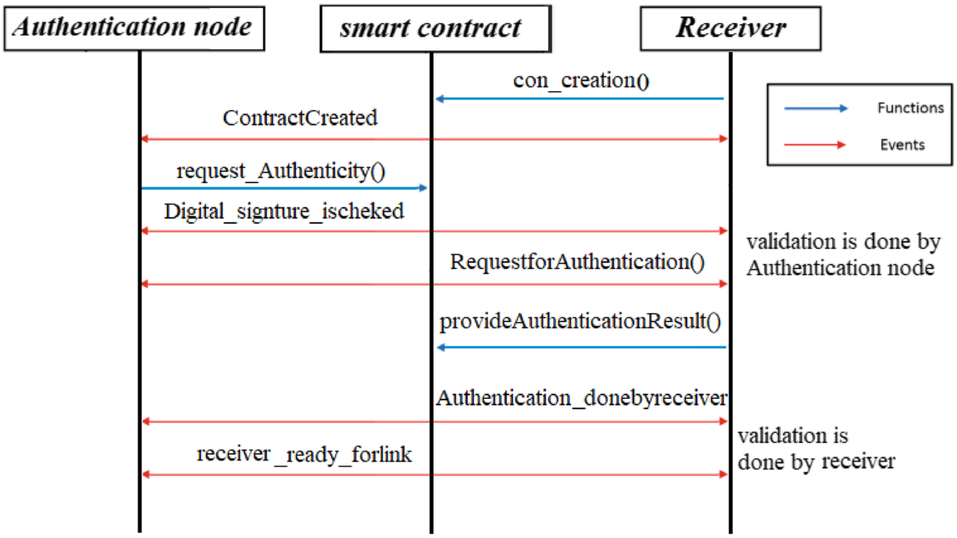
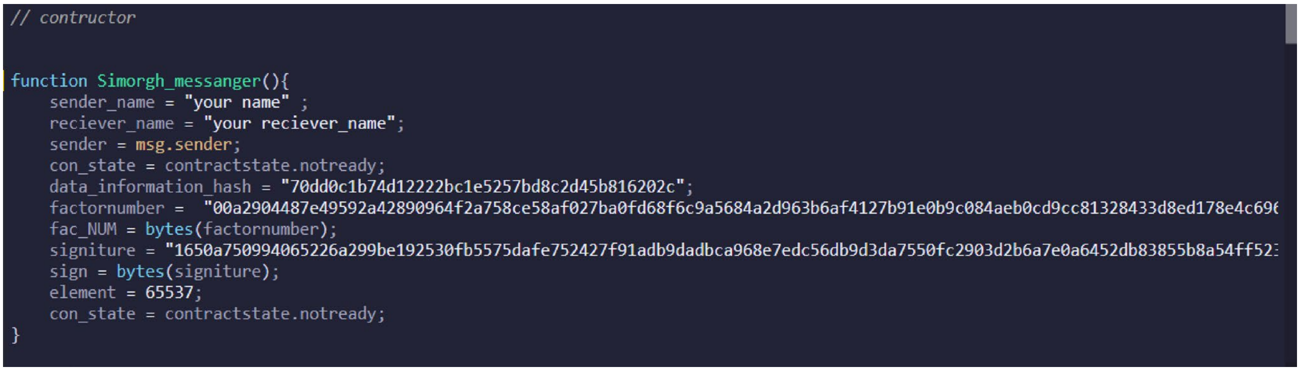

sender
smart contract
Authentication node
con_creation()
ContractCreated
Functions
Events
request_Authenticity()
Digital_signture_ischeked
validation is done by Authentication node
RequestforAuthentication()
provideAuthenticationResult()
sender_is_readytosenddata
validation is done by sender
Authentication_Donebysender

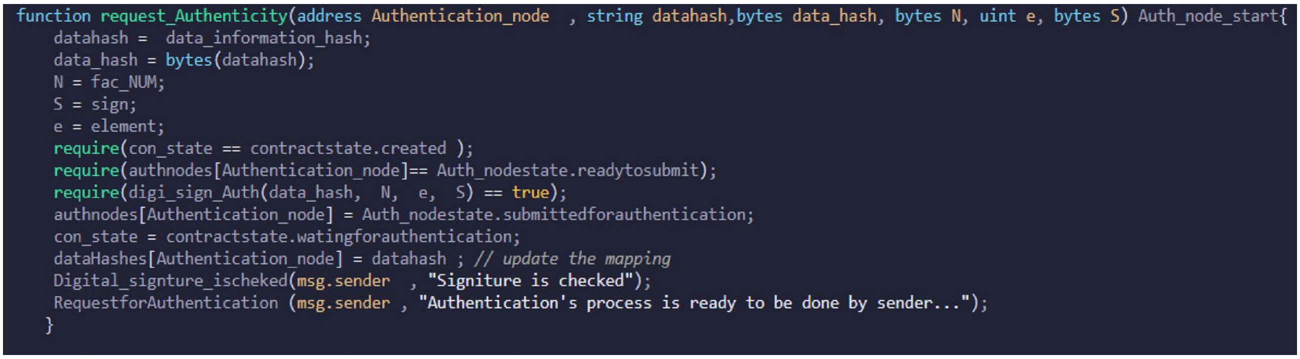
Fig. 6 Smart contract between receiver and authentication node


Authentication node
smart contract
Receiver
con_creation()
Functions
Events
ContractCreated
request_Authenticity()
Digital_signture_ischeked
validation is done by Authentication node
RequestforAuthentication()
provideAuthenticationResult()
Authentication_donebyreceiver
validation is done by receiver
receiver_ready_forlink



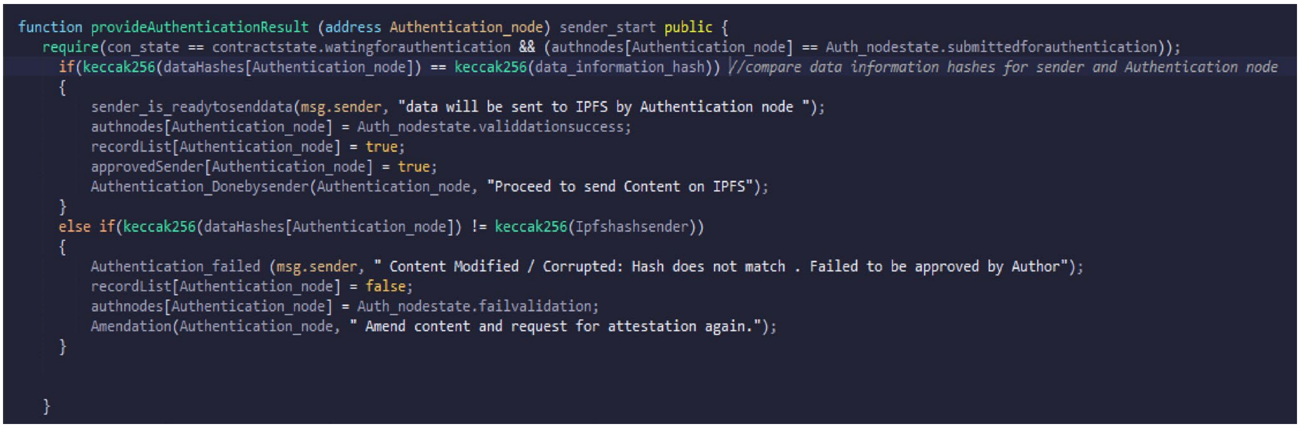
```
// contructor


function Simorgh_messanger(){
    sender_name = "your name" ;
    reciever_name = "your reciever_name";
    sender = msg.sender;
    con_state = contractstate.notready;
    data_information_hash = "70dd0c1b74d12222bc1e5257bd8c2d45b816202c";
    factornumber =  "00a2904487e49592a42890964f2a758ce58af027ba0fd68f6c9a5684a2d963b6af4127b91e0b9c084aeb0cd9cc81328433d8ed178e4c69
    fac_NUM = bytes(factornumber);
    signiture = "1650a750994065226a299be192530fb5575dafe752427f91adb9dadbca968e7edc56db9d3da7550fc2903d2b6a7e0a6452db83855b8a54ff52
    sign = bytes(signiture);
    element = 65537;
    con_state = contractstate.notready;
}
```

Fig. 7 Constructor of the smart contract between sender and authentication node



```
function request_Authenticity(address Authentication_node  , string datahash,bytes data_hash, bytes N, uint e, bytes S) Auth_node_start{
    datahash =  data_information_hash;
    data_hash = bytes(datahash);
    N = fac_NUM;
    S = sign;
    e = element;
    require(con_state == contractstate.created );
    require(authnodes[Authentication_node]== Auth_nodestate.readytosubmit);
    require(digi_sign_Auth(data_hash,  N,  e,  S) == true);
    authnodes[Authentication_node] = Auth_nodestate.submittedforauthentication;
    con_state = contractstate.watingforauthentication;
    dataHashes[Authentication_node] = datahash ; // update the mapping
    Digital_signture_ischeked(msg.sender  , "Signiture is checked");
    RequestforAuthentication (msg.sender , "Authentication's process is ready to be done by sender...");
   }
```

Fig. 8 Verification of sender by `request_Authenticity` function



```
function provideAuthenticationResult (address Authentication_node) sender_start public {
   require(con_state == contractstate.watingforauthentication && (authnodes[Authentication_node] == Auth_nodestate.submittedforauthentication));
     if(keccak256(dataHashes[Authentication_node]) == keccak256(data_information_hash)) //compare data information hashes for sender and Authentication node
     {
         sender_is_readytosenddata(msg.sender, "data will be sent to IPFS by Authentication node ");
         authnodes[Authentication_node] = Auth_nodestate.validdationsuccess;
         recordList[Authentication_node] = true;
         approvedSender[Authentication_node] = true;
         Authentication_Donebysender(Authentication_node, "Proceed to send Content on IPFS");
     }
     else if(keccak256(dataHashes[Authentication_node]) != keccak256(Ipfshashsender))
     {
         Authentication_failed (msg.sender, " Content Modified / Corrupted: Hash does not match . Failed to be approved by Author");
         recordList[Authentication_node] = false;
         authnodes[Authentication_node] = Auth_nodestate.failvalidation;
         Amendation(Authentication_node, " Amend content and request for attestation again.");
     }


   }
```

Fig. 9 Confirm sent data by `provideAuthenticationResult` function

the smart contract between sender and authentication node (Fig. 7).

After the sender has made the contract, the sender's identity must be verified for authentication node. For this purpose, the sender's signature on the sent hash data will be checked and if the signature is confirmed, the contract will continue to be executed (Fig. 8).

Next, the sender must ensure the accuracy of the data received by the authentication node. Therefore, by checking the hash of the message received by the authentication node and comparing it with the original message, the authentication of the message is done (Fig. 9).

8 Conclusions

Simorgh is an information exchange system which is fully decentralized and blockchain-based. IPFS technology was used in Simorgh due to the limitation of blockchain storage and database decentralization. Simorgh is the first fully decentralized information exchange system in which the authentication of nodes, messages, and data storage is fully decentralized.

In Simorgh, three security features are available including privacy, integrity, and accessibility. The encrypted communication can be provided as E2E and fully private, leading to increased privacy because Simorgh allows to separate data encryption and data transfer. In addition, integrity is achievable by blockchain technology and authentication nodes. Also, Simorgh's structure and IPFS technology cause appropriate accessibility.

The formation of all communities and structures is basically based on the relationship between each of the small components of these communities and structures, and therefore Simorgh can be considered a safe and completely decentralized system for managing all structures, and most of the problems in these structures can be solved with the customizing of Simorgh in the fields of communication, Internet of Things, data transparency, financial communications, Securities and Exchange Organization, audit organizations, etc.

One of the most important benefits that Simorgh provides is that after its operational implementation, to joining

Simorgh, many companies and organizations, even large companies such as WhatsApp, Telegram and Facebook, can create an authentication node and define their own algorithms and protocols for their user and causing Simorgh globalization. Simorgh can acquire many assets using cryptocurrencies. The structure of Simorgh can be customized in such a way that it can be used in many sciences and industries and it can also play an effective role in many problems and the design of structures in today's world. However, Simorgh can be a start for valuable research. In our opinion, Simorgh is the first structure designed to overthrow centralization of management in the world.

References

Beregszaszi A (2016) RSA signature verification in Ethereum. MIT License Copyright (C) 2016. https://www.github.com/axic/ethereum-rsa. Accessed 6 Apr 2016

Bitcoin Energy Consumption Index, Digiconomist. https://www.digiconomist.net/bitcoin-energy-consumption/. Accessed 29 Nov 2019

Al-Turjman F, Nawaz MH, Ulusar UD (2020) Intelligence in the Internet of Medical Things era: a systematic review of current and future trends. Comput Commun 150:644–660

Allioui H, Sadgal M, El Fazziki A (2020) An improved image segmentation system: a cooperative multi-agent strategy for 2D/3D medical images. J Commun Softw Syst 16(2):143–155

Benet J (2014) Ipfs-content addressed, versioned, p2p file system. arXiv preprint arXiv: 1407.3561

Bernaille L, Teixeira R (2007) Early recognition of encrypted applications. In: International conference on passive and active network measurement. Springer

Biktimirov M, Domashev A, Cherkashin P, Shcherbakov AY (2017) Blockchain technology: universal structure and requirements. Autom Doc Math Linguist 51(6):235–238

Chakour I, El Mourabit Y, Daoui Y, Baslam M (2020) Multi-agent system based on machine learning for early diagnosis of diabetes. In: 2020 IEEE 6th international conference on optimization and applications (ICOA). IEEE

Chronopoulos AT, Musku MR, Penmatsa S, Popescu DC (2008) Spectrum load balancing for medium access in cognitive radio systems. IEEE Commun Lett 12(5):353–355

Clack CD, Bakshi VA, Braine L (2016) Smart contract templates: foundations, design landscape and research directions. arXiv preprint arXiv: 1608.00771

CommonAccord (2016) http://www.commonaccord.org/

Croman K, Decker C, Eyal I, Gencer AE, Juels A, Kosba A, Miller A, Saxena P, Shi E, Sirer EG (2016) On scaling decentralized blockchains. In: International conference on financial cryptography and data security. Springer

Daian P, Pass R, Shi E (2019) Snow white: robustly reconfigurable consensus and applications to provably secure proof of stake. In: International conference on financial cryptography and data security. Springer

Du M, Chen Q, Chen J, Ma X (2020) An optimized consortium blockchain for medical information sharing. IEEE Trans Eng Manag 68(6):1677–1689. https://doi.org/10.1109/TEM.2020.2966832

Eyal I, Gencer AE, Sirer EG, Van Renesse R (2016). Bitcoin-ng: a scalable blockchain protocol. In: 13th {USENIX} symposium on networked systems design and implementation ({NSDI} 16)

Grigg I (2004) The Ricardian contract. In:: Proceedings of the first IEEE international workshop on electronic contracting, 2004. IEEE

Güncelleme (2016) IPFS is starting to replace HTTP!". http://www.chip.com.tr/haber/ipfs-httpnin-yerine-gecmeye-hazirlaniyor.html

IO E (2017) EOS. IO technical white paper. EOS. IO. https://github.com/EOSIO/Documentation. Accessed 18 Dec 2017

Stark J (2016) Making sense of blockchain smart contracts. http://www.coindesk.com/making-sense-smart-contracts/

Jamil F, Hang L, Kim K, Kim D (2019) A novel medical blockchain model for drug supply chain integrity management in a smart hospital. Electronics 8(5):505

Karaarslan E, Akbaş MF (2017) blokzinciri tabanli siber güvenlik sistemleri. Uluslararası Bilgi Güvenliği Mühendisliği Dergisi 3(2):16–21

Karame G (2016) On the security and scalability of bitcoin's blockchain. In: Proceedings of the 2016 ACM SIGSAC conference on computer and communications security

Khacef K, Pujolle G (2019) Secure peer-to-peer communication based on blockchain. In: Workshops of the international conference on advanced information networking and applications. Springer

Kiayias A, Russell A, David B, Oliynykov R (2017). Ouroboros: a provably secure proof-of-stake blockchain protocol. In: Annual international cryptology conference. Springer

King S, Nadal S (2012) Ppcoin: Peer-to-peer crypto-currency with proof-of-stake. Self-published paper, August 19: 1

Legalese (2016) http://www.legalese.com/

Li W, Cheng X, Bie R, Zhao F (2016) An extensible and flexible truthful auction framework for heterogeneous spectrum markets. IEEE Trans Cognit Commun Netw 2(4):427–441

Loibi A (2014) Namecoin. In: Seminar innovative Internettechnologien und mobilkommunikation SS2014. IEEE

Lu H, Tang Y, Sun Y (2020) DRRS-BC: decentralized routing registration system based on blockchain. IEEE/CAA J Autom Sin 7(1):1–9

Meng W, Li W, Zhu L (2019) Enhancing medical smartphone networks via blockchain-based trust management against insider attacks. IEEE Trans Eng Manag 67(4):1377–1386

Mostafa SA, Mustapha A, Mohammed MA, Ahmad MS, Mahmoud MA (2018) A fuzzy logic control in adjustable autonomy of a multi-agent system for an automated elderly movement monitoring application. Int J Med Inform 112:173–184

Nakamoto S (2008) Bitcoin: a peer-to-peer electronic cash system. Decentralized Bus Rev 31:21260

Network P (2018) Poa-network-whitepaper. Accessed 10(3): 19

Nizamuddin N, Hasan HR, Salah K (2018) IPFS-blockchain-based authenticity of online publications. In: International conference on blockchain. Springer

Połap D, Srivastava G, Jolfaei A, Parizi RM (2020) Blockchain technology and neural networks for the internet of medical things. In: IEEE INFOCOM 2020-IEEE conference on computer communications workshops (INFOCOM WKSHPS). IEEE

Połap D, Srivastava G, Yu K (2021) Agent architecture of an intelligent medical system based on federated learning and blockchain technology. J Inf Secur Appl 58:102748

Salim MM, Shanmuganathan V, Loia V, Park JH (2021) Deep learning enabled secure IoT handover authentication for blockchain networks. Hum-Cent Comput Inf Sci 11:12

Sarıtekin RA, Karabacak E, Durgay Z, Karaarslan E (2018) Blockchain based secure communication application proposal: Cryptouch. In: 2018 6th international symposium on digital forensic and security (ISDFS). IEEE

Shen M, Deng Y, Zhu L, Du X, Guizani N (2019) Privacy-preserving image retrieval for medical IoT systems: a blockchain-based approach. IEEE Netw 33(5):27–33

Swanson T (2014) Making sense of blockchain smart contracts. https://www.s3-us-west-2.amazonaws.com/chainbook/Great+Chain+of+Numbers+A+Guide+to+Smart+Contr

Szabo N (1997) Formalizing and securing relationships on public networks. First Monday

Vimal S, Srivatsa S (2019) A new cluster p2p file sharing system based on ipfs and blockchain technology. J Ambient Intell Humaniz Comput 1–7

VISA Fact Sheet (2019) https://www.usa.visa.com/dam/VCOM/global/about-visa/documents/visa-fact-sheet-july-2019.pdf

Vukolić M (2015) The quest for scalable blockchain fabric: proof-of-work vs. BFT replication. In: International workshop on open problems in network security. Springer

Yaacoub J-PA, Noura M, Noura HN, Salman O, Yaacoub E, Couturier R, Chehab A (2020) Securing internet of medical things systems: limitations, issues and recommendations. Future Gener Comput Syst 105:581–606

Yazdinejad A, Srivastava G, Parizi RM, Dehghantanha A, Choo K-KR, Aledhari M (2020) Decentralized authentication of distributed patients in hospital networks using blockchain. IEEE J Biomed Health Inform 24(8):2146–2156

Z, K. E. D. g (2018) Usage of blockchain technology in the E-government applications: preliminary study (Turkish). Akademik Bili¸sim

Zhou L, Wang L, Sun Y (2018) MIStore: a blockchain-based medical insurance storage system. J Med Syst 42(8):1–17

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.