SN Computer Science (2021) 2:346
https://doi.org/10.1007/s42979-021-00746-x

ORIGINAL RESEARCH

CoviChain: A Blockchain Based Framework for Nonrepudiable Contact Tracing in Healthcare Cyber-Physical Systems During Pandemic Outbreaks

Sukrutha L. T. Vangipuram[1] · **Saraju P. Mohanty**[1] · **Elias Kougianos**[2]

Received: 18 April 2021 / Accepted: 11 June 2021 / Published online: 20 June 2021

Abstract
With the world facing the new virus SARS-CoV-2, many countries have introduced instant Internet applications to identify people carrying the infection. Internet-of-Medical-Things (IoMT) have proven useful in collecting medical data as well in tracing an individual carrying the virus. The data collected or traced belongs to an individual and should be revealed to themselves and hospital providers, but not to any third-party unauthorized agencies. In this paper we use an off-chain distributed storage solution for loading large medical data sets and a blockchain implementation to securely transfer the data from the infected person to the hospital system using the edge infrastructure, and call it CoviChain. The Coronavirus Disease (COVID-19) statistics are loaded on to the edge, and moved to InterPlanetary File Systems (IPFS) storage to retrieve the hash of the data file. Once the hash is obtained, it is moved to the blockchain by means of smart contracts. As the information is being hashed twice, CoviChain addresses the security and privacy issues and avoid exposing individuals' data while achieving larger data storage on the blockchain with reduced cost and time.

Keywords Internet-of-Medical-Things (IoMT) · Healthcare cyber-physical system (H-CPS) · Blockchain · COVID-19 · Contract tracing · Data privacy · Data provenance

Introduction

People globally have been facing a new challenge called Coronavirus Disease (COVID-19), which has been pandemic in nature. A significant drawback of this disease is that it is not recognizable in those containing it, which is risky for other people. All the countries in the world have implemented lock-down strategies to break the chain, but this has economic implications, including increased poverty in many countries. Being connected to our daily lives and at the same time stay aware of the persons carrying the virus, technological solutions have been made available through contact awareness wearable devices via Internet-of-Medical-Things (IoMT), and mobile applications [1].

Many countries that have lifted lock-downs are suffering from the second or third waves of virus reinfecting people. Continued data filing from patients will help mitigate the issue by performing studies and observations on these facts. IoMT devices with appropriate bio-sensors recognize the virus; as the devices collect data 24/7, a lot of information accumulates near the device leading to limited storage considerations, along with continuous attempts for patient site breaching by hackers [1]. Time-bound storage lists one of the motivations for our writing since the patient's preliminary information is not available when needed the most, which is crucial for research, mainly when following outbreaks occur. Progress in various offline and online devices to determine COVID-19 spread is being done on a continuous base [2]. Gathering COVID-19 data with multiple platform medical things and central storing systems leads to interoperability problems, single-point failure, and latency issues. Furthermore, sending sensory readings from a wearable device to

✉ Saraju P. Mohanty
saraju.mohanty@unt.edu

Sukrutha L. T. Vangipuram
lakshmisukruthatirumalavangipuram@my.unt.edu

Elias Kougianos
elias.kougianos@unt.edu

[1] Department of Computer Science and Engineering, University of North Texas, Denton, TX, USA

[2] Department of Electrical Engineering, University of North Texas, Denton, TX, USA

medical authorities raises another test for data consistency in existing models. To overcome these critical issues using the blockchain (BC) and Distributed Storage Systems (DSS) would benefit current attempts. Figure 1 shows the problems and motivation for our paper.

Authentication is required from both the patient's side and the organizations to see if the data originates from a genuine patient. In this urgent situation, where time is priceless to save lives, we are not in a position to accept slow data access, interoperability limitations, and uneven data distribution. BC provides a decentralized, distributed architecture that uses cryptography as a security tool for creating immutable blocks consisting of transactions and data ordered in chains. These blocks, once appended to the chain, cannot be altered or modified and are secured with the help of hash functions and timestamps on transaction data. All blocks in the chain are of the same size. In addition, mining processes help validate the transaction block and help secure the blockchain network from malicious attacks. Smart contracts are programs to execute logic that act like small services and application program interfaces. As popular as blockchain is, the system is not perfect. Some of the challenges it faces include high fees and slow bulk data authentication, hence we use in this work a distributed storage system, namely the Interplanetary file system.

The paper makes use of a three-tier architecture, where the edge node rests in between the IoMT and the cloud. Transmitting the sensory data of COVID-19 patients from wearable devices towards edge and store in blockchain through contracts to prevent changes while transferred to web interfaces of Hospital Systems (HS) in H-CPS. Many applications are been developed that include blockchain mechanisms set in the IoMT-fog or IoMT-edge for increasing cybersecurity [3, 4]. In this paper, we explore the design and implementation of "Covichain" for sending COVID-19 readings from the patients' IoMT devices to the COVID-19 service provider (CSP) along with storing past readings of these patients cohesively for Hospital systems (HS) to keep track of these persons and assist in future research of the disease. Practicing Covichain combined with an edge setting is introduced as the main idea here.

The rest of the paper is organized in the following way. "Novel Contributions of the Current Paper" discusses the novel contributions through problem definition and challenges while finding novel solutions. Related works are discussed in "Related Prior Works". "A Novel Blockchain Based Framework for Nonrepudiable Contact Tracing" presents details of proposed CoviChain. "The Proposed Algorithms for Nonrepudiable Contract Tracing in CoviChain" presents the algorithms for contact tracing in CoviChain. "Implementation of CoviChain" has details of the implementation of CoviChain. "Experimental Validation and Results" presents experimental results. "Conclusion and Future Work" presents the conclusions and directions of future research.

Novel Contributions of the Current Paper

This section explains different problems that are present in carrying the COVID data and lists main drawbacks in the current IoMT and mobile applications. Solutions to the existing issues are analyzed and presented by proposing a novel architecture for handling and storing data through fog nodes. Decentralized handling of information is more advantageous, which makes the system resistant to attacks and single points of failure.

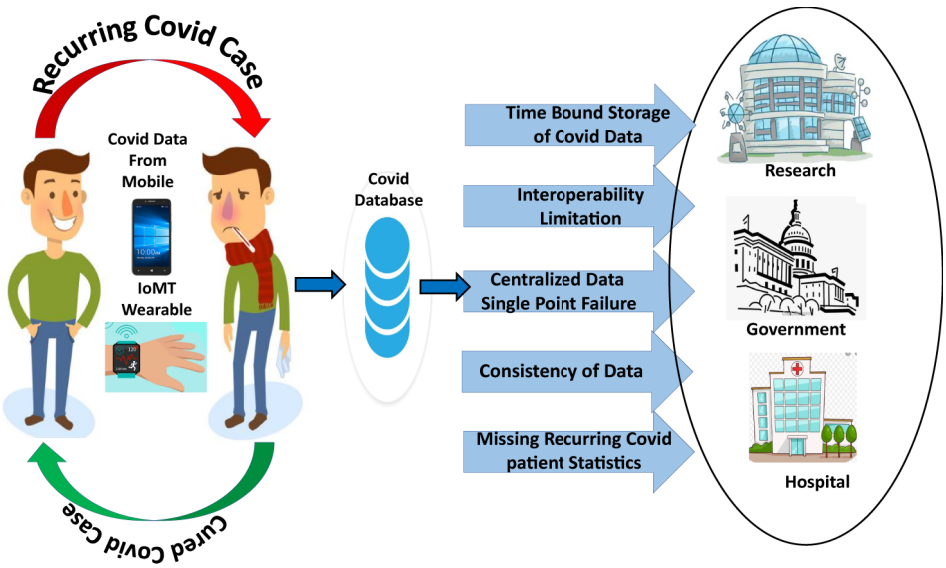


Fig. 1 Need of Covichain as nonrepudiable contact tracing

Problem Definition

In this COVID-19 situation worldwide countries have deployed mobile applications for tracing people who are infected with the virus. These applications are required to monitor and have access to the users' Bluetooth and/or Wi-Fi and cellular stacks continuously to identify the patient's location, which leads to high risk of invasion from both security and privacy viewpoints. Wearable IoMT devices are another approach to self-distance from people having the virus, and to alert authorities. The storage capacity of the patient's data in these devices is very low, is limited in time and is held in a centralized approach where distributing through the cloud and edge centers is risky. Hence, for storing intermittent wearable data, practicing the Edge layer with a Distributed storage system to avoid central servers and blockchain sharing to maintain immutable data and privacy.

The Challenges Faced in the Current COVID-19 IoMT

As more COVID-19 IoMT connections are established for contact awareness, more information is shared between these devices which consume more energy, and the possibility of hacking of the confidential readings of the patients and tampering is also increased. Moreover, storing this large data becomes a big challenge as every bit of information plays a vital role in further analysis. The data integrity throughout the flow from patients to hospitals is very hard to attain with all the compatibility issues and size of the data being transferred. When the data is distributed in a centralized model, if the wrong information is forwarded or is erroneous, there is a possibility that every other device connected can be corrupted. Figure 2 shows the problems that occur in the current COVID-19 IoMT devices in H-CPS framework. The burden to process data with security is increased on both the wearable device, as well as the cloud which leads to data latency issues.

Proposed Solution

Computing at the edge is done near the client devices for acquiring, storing, and examining facts instead of using centralized data processing. In our use case, the COVID-19 health records (CHR) traverse from wearable medical devices to the intermediate edge; hence the BC is used at the edge to maintain the similarity of the CHR while sharing the data. Some of the drawbacks of the BC comprise are the cost to upload data blocks and the time taken to validate the block; we have reduced the time and cost of BC by using distributed storage in the edge layer along with minimizing dependence on central systems.

The Novelty of the Proposed Solution

Time-bound storage leading to missing or incomplete COVID-19 statistics, and latency in centralized systems are the main challenges that occur due to the inefficiency of the current COVID-19 contact tracing devices and healthcare systems. Existing wearable devices cannot handle storage and dispersal strategies with their limited resources and low processing powers. By introducing an intermediate edge, statistics coming from the IoMT can be processed and manipulated near the device and in efficient ways. Distributed storage can handle larger uploaded data, and avoids central storage issues and maintains immutability and privacy of the COVID-19 health records through blockchain sharing. IPFS and blockchain functionality embedded into the edge layer allow COVID-19 data flow across heterogeneous systems. The BC implementation reads and stores IoMT data and handles calculations and verification of hashes to maintain

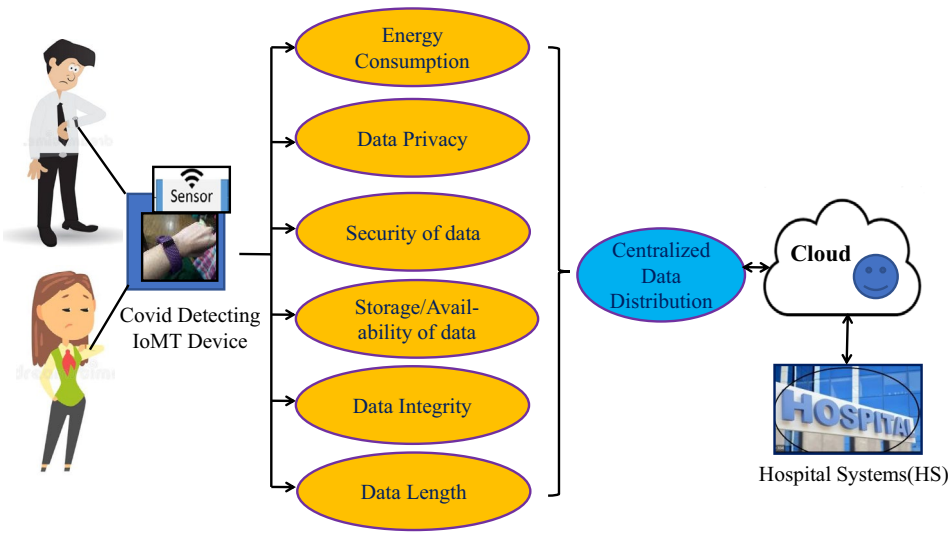


Fig. 2 Challenges in current COVID IoMT devices and central servers

the integrity of facts in chains. IPFS distributed storage with asymmetric encryption is used to load larger datasets.

Related Prior Works

Blockchain has been explored to be deployed in a variety of applications including smart healthcare, smart cities, and smart agriculture [5–10]. Whether it is contract tracing records, or medical data originating in the IoMT, mobile or any smart device, there have been different methods proposed to address the challenges and problems.

As described in "Novel Contributions of the Current Paper", many Contract Tracing Applications have been introduced by different governments to bring immediate resolutions for the current COVID-19 situations [1, 11–13]. But these devices along with instant advantages carry disadvantages with their limited capability to store and secure patients' data. During tracing, an infected individual consumes a lot of computation at the client end before transmitting the values to storage. While transmitting the data, there is high level of risk involved in both privacy and security terms along with latency and storage issues. A study of privacy preserving in these contact tracing kits provides a broader picture and the measure of how these devices use different protocols to keep an individual's data secure [14].

An Electronic Health Records are managed and updated in a cohesive manner using blockchain [15]. It also provides easy access to medical information for patients, makes the data resistant to variations and addresses interoperability issues with the help of Blockchain Smart Contracts. But it requires traditional database approaches for cache storage of medical data at the provider end, which is still contingent on centralized systems.

Blockchain security methods is deployed by installing bolsters (computing machines) near hospital systems to act as a server for IoMT devices as a private block, while securely interacting with other blocks in [16]. The bolster stores all readings in a blockchain ledger, prevents attackers from installing malware on devices and stops linking attacks. Cloud Layers are used for storing, analyzing and running algorithms on blocks of data for increasing security. This results in large amounts of data on the cloud .

A trust management scheme using blockchain in medical smartphone networks for identifying malicious nodes in an efficient way is presented [17]. The architecture presented gives an in-depth analysis of the system with well-organized elucidation supported with facts and outputs for secured health records from smartphones. However, the use of a central server in the design can be a major target by the attackers to compromise the medical records.

In [18], communication between different entities has been provided in a protected way by using key management. Validated users can access healthcare data from the cloud servers. The complete healthcare data is stored on a blockchain which is maintained through cloud servers and tested against all possible attacks. Here, the blockchain containing sensitive data of patients has to be sustained wholly with the help of cloud servers, which can lead to latency issues and time-consuming scenarios with huge data around for processing and analyzing.

In [19], a blockchain-based medical research support platform has been presented which can facilitate privacy-preserving data sharing against COVID-19 to speed up collaborative research among healthcare providers.

A novel blood sample-based Emergency Department (ED) return scheme that predicts the return probability of ED for COVID-19 has been presented in [10]. The scheme analyzes four kinds of blood samples (i.e., whole blood, serum, plasma, and arterial whole blood) for each patient. The patient information is then encrypted and stored in the blockchain database in IoMT.

While these methods of contract tracing devices and applications to store medical data on blockchain have increased security and privacy for Electronic Health Records (EHR) through different implementations, still limitations are prominent as some are solely designed with cloud layers and have dependence on centralized servers for storage while some have reliance on the client side for heavy computations. Storage of larger data on the IoMT contact tracing devices has always faced issues, but storing data of infected patients is necessary for further research as the infection is presented in variants in different countries. Relying on the cloud for data processing is a laborious task in terms of the huge data to be stored and authenticated through centralized servers, where the system can have a single point of failure which makes it limited in terms of reliability and security. We utilize edge nodes in our current CoviChain in IoMT/H-CPS system for computing and processing to overcome the limitations of IoT devices, with further addition of the Interplanetary File System for distributed storage combined with blockchain smart contracts. A comparative analysis for storage and security for the data originating from contract tracking devices, blockchain healthcare applications, and the proposed Covichain is given in detail in Tables 1 and 2 .

A Novel Blockchain Based Framework for Nonrepudiable Contact Tracing

H-CPS Components

IoMT/Healthcare-Cyber Physical System (H-CPS) It is the system which connects, manages and controls the physical organizations with virtual structures through networks [20].

Table 1 Comparative analysis of data storage in contact tracing devices and current Covichain

Application	Protocol	Edge	Data storage	Level of security	Computation
Aarogyasetu [1]	iBeacon	No	Centralized	High Privacy Breach	High at client side
TraceTogether [1]	Blue Trace	No	Partially Centralized	Possible Privacy Breach	High at client side
CovidSafe [12]	Blue Trace	No	Partially Centralized	User Privacy Concerns	High at client side
SwissCovid [11]	DP-3T	No	Decentralized	High Privacy Breach	Very High at client side
CoviChain [current paper]	IPFS	Yes	Off-chain Decentralized	File Hashing	Less at client side

Table 2 Comparative analysis of data security in blockchain healthcare applications and current Covichain

Application	Protocol	Edge	Blockchain type	Data storage	Level of security	Computation
MedRec[15]	Smart contract	No	Public	Off-chain centralized	Contract not encrypted	High at the User Side
BIoMT [16]	ECC, IBC	No	Light weight	Off-chain centralized	Good for smaller IoMT data	High at User Side
BACKMP-IoMT [18]	AKMP	No	Private	Centralized cloud server	Single hash security	–
CoviChain [current paper]	Smart contract	Yes	Public Ethereum	Off-chain decentralized	File hashing	Less computation at client

– Not in the paper, *EEC* elliptical curve cryptography, *IBC* identical based credential, *AKMP* authenticated key management protocol, *IPFS* interplanetary file system, *SC* smart contract

The hospital systems get the data from patients through different invasive and non-invasive devices and are able to process intelligently the dynamically changing health inputs.

Blockchain (BC) It provides a decentralized architecture which uses cryptography as a security tool for creating immutable blocks consisting of transactions and data that are ordered in the form of a chain. These blocks of chain cannot be altered or modified. The records in a block are secured with the help of hash functions that belong to the previous blocks, along with timestamp and transaction data. Transactions in the blockchain are non-recursive and are of parallel size. Each node can view all the blocks, but it is impossible to alter it. To attack, the node must have 51% share in controlling the whole network which is not possible without vast computer resources . Also, to change any data in the block, a rogue node requires other peers to validate this action [21]. Blockchain applications can be written through smart contracts with source code and deployed in the backend to communicate with the blockchain. They act like a digital contract or agreement [22].

Distributed Storage System It is for storing large files and accessing them with ease through websites and applications. It supports a robust Internet where, if servers get attacked, still web pages can be retrieved from other places. As the data comes from various locations, the content cannot be censored and speeds up the network when individuals reside in distant places or get disconnected [23]. The information can be retrieved from nearby devices instead of recovering it from distant cites.

Novel Blockchain (CoviChain) in H-CPS for Contact Tracing Data - Architecture in H-CPS

Contact tracing wearable devices are mainly used for identifying persons carrying an unknown virus for preventing further spread of the disease. During this process, while sending Electronic Health Records (EHR), there is a danger of personal data compromised, vital facts modified, and content could no longer be private and secure. Here we introduce a novel blockchain which is designed to transfer the Electronic health records (EHR) in the form of immutable ledgers through edges device along with a distributed storage system for service providers. The architecture of Health-Cyber Physical Systems (H-CPS) is shown in Fig. 3. The edge systems are responsible for storing, analyzing and transferring large amounts of data near the source by increasing data stream acceleration. Decentralized data storage and distribution is embedded in to the edge design that avoids repetition of medical records, dependencies on third parties, has immutable storage, offline data access, faster browsing, user/patient control of the information, data integrity, double spending evasion and valid consensus establishment to make different participating nodes to agree on the state of transactions (here Medical data). For blockchain content storage, several blockchains like Ethereum collect fees from miners. To reduce the charges associated for storing data we have chosen to store only the hash value, instead of the direct content on the blockchain.

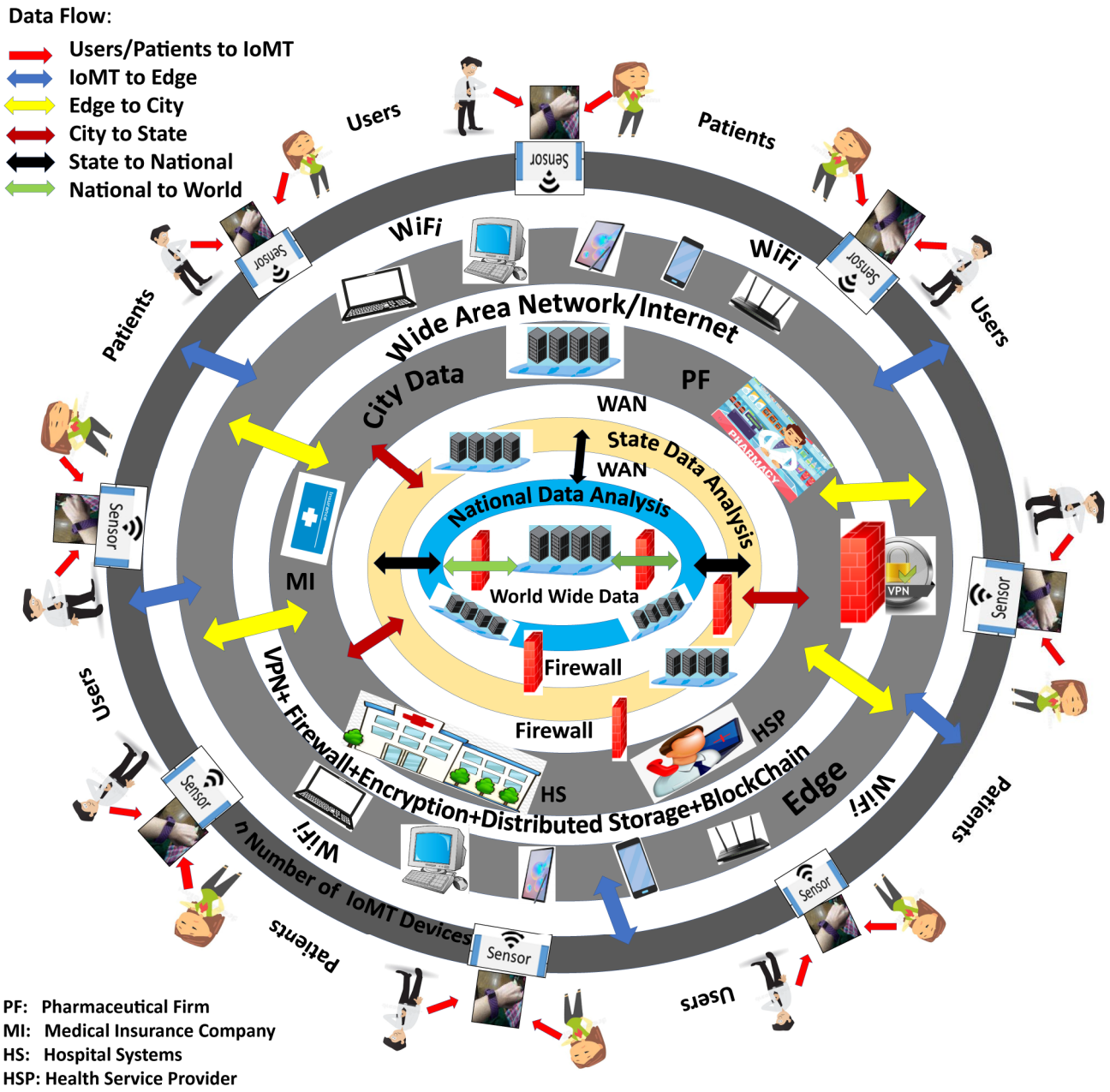


Fig. 3 Proposed H-CPS framework with blockchain

Novel Blockchain (CoviChain) in H-CPS for Contact Tracing Data - High-Level Operation

The components interact with each other to authenticate themselves. The simplified architecture of the proposed system is given in Fig. 4.

The following components play key roles in the proposed system and here we provide brief description of each.

IoMT devices Connecting different computing devices including mechanical, electrical, people, living and nonliving things which are assigned with a unique ID to transfer data through networks without the intervention of humans or machines. There are basically two functionalities of IoT, sensing and tasking. In sensing, sensors mainly collect environmental data such as temperature, pressure etc. while tasking abilities allows completing the implementation remotely. Here each IoMT device is mapped to one edge node.

Edge/Fog Nodes The main functionality of theses nodes is to have local storage facility and perform analysis of the

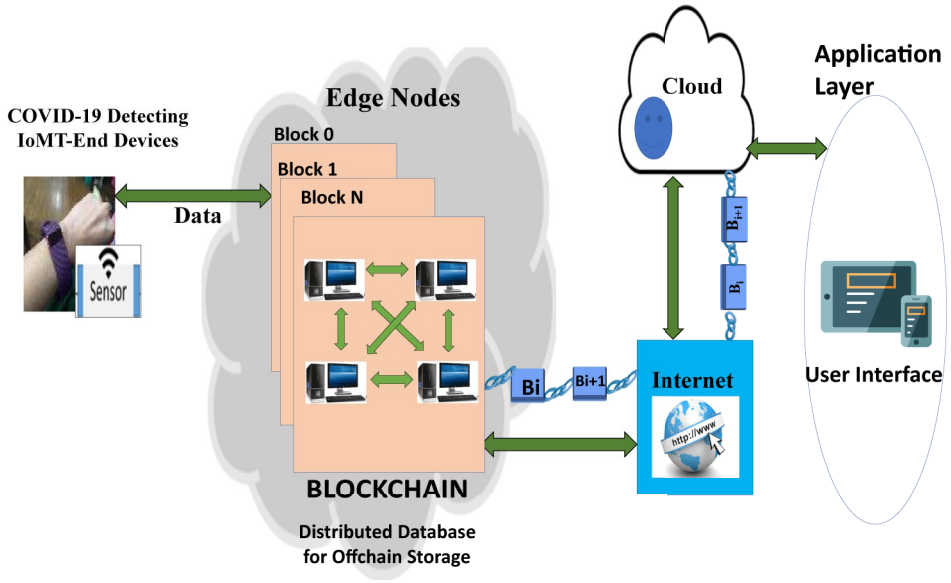


Fig. 4 A simplified view of the proposed CoviChain architecture

data near the IoMT device to reduce latency and response time. The edge increases capabilities and manages COVID-19 IoMT devices and relieves them from drain of storage, memory and computation loads involved in authentication and interfacing with the Ethereum Network.

Public/Private Blockchain This part provides the full functionalities that are required to connect and participate in the BC network. It connects to the peer to peer network, sends transactions and performs programming in detail. These are building blocks used to create BC applications. They are programs that can be written with source code and installed into the blockchain. The programming language used to write logic in smart contracts is Solidity which is immutable. Once they are deployed, the code cannot be updated like a normal application. The code of the contract contains mappings of edge nodes that are registered to their associated IoMT devices. All user and device registration, authentication, CSP staff calls and responses, OTP Generation and fee transfers are developed as function calls to interact with Blockchain Network.

Offchain Distributed File System It is used for storing large amounts of data and uses a decentralized network to make data available with or without an Internet connection. It performs hashes on the actual content to convert the data in the form of immutable links which can then be passed on to the BC avoiding direct data insertion in to the chains.

COVID-19 Service providers (CSP) They receive alert messages from the IoMT wearable devices and give a response with required services through notifying Hospital Systems (HS). They also make sure that the participants sending or receiving the data are legitimate from both ends.

Hospital Systems (HS) They are the end systems that receive COVID-19 data and authenticate themselves to the patients with the help of OTP which is time-bound to act accordingly as the situation demands based on the patient's readings.

Cloud Its primary purpose is to combine and store IoMT data for processing and analyzing. Authenticating users and cloud servers to IoMT devices using OTP validation methods and Username/Password creation is performed here.

The Proposed Algorithms for Nonrepudiable Contract Tracing in CoviChain

In this section we give detailed explanations of the protocols used in collecting COVID-19 data, sending them to the edge, storing data in distributed fashion and implementing the blockchain, and how the application can be designed to a larger scale using blockchain communication protocols between nodes.

Proposed Layered Architecture of CoviChain

With the help of Radio frequency Identification and Wireless Sensor Networks the COVID-19 data is traced and forwarded to the Edge nodes with the help of Bluetooth, Wi-Fi, Zigbee or Internet protocol 802.15.4. To avoid frequent storage failures and to maintain consistency of the data we use distributed storage, where data is replicated in different

worldwide storage devices. Once the edge layer receives data, the storage, processing, analyzing, data immutability through Blockchain, and forwarding are done. These steps are handled with a set of rules defined inside the design of the application along with the support of protocols. The distributed storage uses the KAD-DHT protocol which has a set of subprotocols to do different jobs such as node identification and verification, network, routing, block exchange protocols, objects, files, naming, along with public key cryptography embedded in to the storage system. Each object of the distributed storage has two fields called "Data" and "Links". The data field maintains binary data of certain size and links has three subfields for Name of the Link, Hash of the linked object and size of the linked object. Some of the distributed storage protocols examples include IPFS, Swarm and Storj which support interoperation with the consensus mechanism of the blockchain. Centralized systems use supercomputers for computation of engineering applications and to deal with large amounts of data, which increases cost and has too much dependency on these systems.

Through edge computing we can join the unused resources of computers, laptops and smart phones to form a decentralized supercomputer and users can make money by leasing their idle resources with low cost and more accessibility. Golem and Zennet are few distributed supercomputers where blockchain technology can be used to remove central data problems. From distributed storage, the hashed COVID-19 data moves towards the blockchain. Here, blocks of hashed data are taken as transactions and combines both Elliptical curve cryptography to show ownership of the blocks along with Proof-of-Work consensus algorithm which allows free entry to solve consensus and to collectively agree on the state of transaction. Each block has its own fields to verify and validate themselves. The proposed detailed architecture for the CoviChain along with block fields is given in Fig. 5. Some of the important fields in the block are the Previous Block Hash (PBH) and the Merkle Root. The PBH is a digital fingerprint of the block header of all the previous or last added block which is calculated by taking all the fields together and applying SHA256 algorithm twice on them. Once a block enters the blockchain all the transactions or records of COVID-19 are permanent. These transactions are listed as Merkle tree or binary hash tree which is also included in Fig. 5 and the root acts as a digital fingerprint.

The reason for including this Merkle tree protocol in the blockchain is that hashes travel upward and if any malicious user tries to inject fake transaction or fake COVID-19 data to the bottom, this changes all the nodes above it and finally changing root to register as a completely different block and invalid Proof-of-Work (PoW). The PoW algorithm is for confirming transactions or the data in the blocks and adding them to the chain. This algorithm mainly uses mathematical puzzles to be solved. Those who solve them are miners and the process is mining. The users of the network send digital tokens as rewards for validating data in blocks. From the edge the data is made immutable with blockchains and forwarded to other nodes in a decentralized way using interoperable communications with peer to peer connections without central services. One such way to fully embed blockchains in large scale networks for communications is the Whisper protocol. It is an encrypted messaging protocol which allows nodes to directly message with members by hiding sender, receiver and messages from third person. It gives confidentiality in two ways by protecting the content transferred between members and keeps the members participating secret thus making COVID-19 content private and sharing only to the members such as Hospital Systems, Government agencies and research organizations by creating confidential routes from user to city hospitals to state or national agencies to worldwide organizations. Using a shared INFURA infrastructure which acts a bridge to the blockchain helps to provide secure, stable, fault tolerant, robust, balanced and easily scalable blockchain with distributed storage nodes. It eliminates the requirement to install, configure and maintain blockchain Nodes and makes the process easier in global scenarios.

Proposed Algorithms at Different Stake Holders of CoviChain

Each user is registered in the blockchain by generating a public, private key pair to uniquely identify the user. Registering of the device is done by providing an identifier once authentication is completed. As soon as a device is registered, sensors start collecting data. Both source and edge have their respective public and private keys for requesting and uploading for signing and encryption of the data. Once authenticated and verified, uploaded data that need acknowledgment are configured as Confirmable message and that does not need acknowledgment are configured as a Non-Confirmable Message (NON). A Non-Confirmable message always carries a request or response and is not empty and is not acknowledged by the recipient. The request and response messages of sender and recipient are always appended with unique IDs for avoiding duplication of the messages and are secured through Datagram Transport Layer Security (DTLS) by using Raw Public Key and X.509 Certificate modes for node identifications and mutual authentications of nodes. The asymmetric public key pairs and certificates along with SHA-256 algorithm specifications are already inserted in the devices during manufacturing for recognizing nodes and message exchanges through verification. NON LIFETIME is the time from sending a Non-confirmable message to the time its message ID can be safely reused again for other messages.

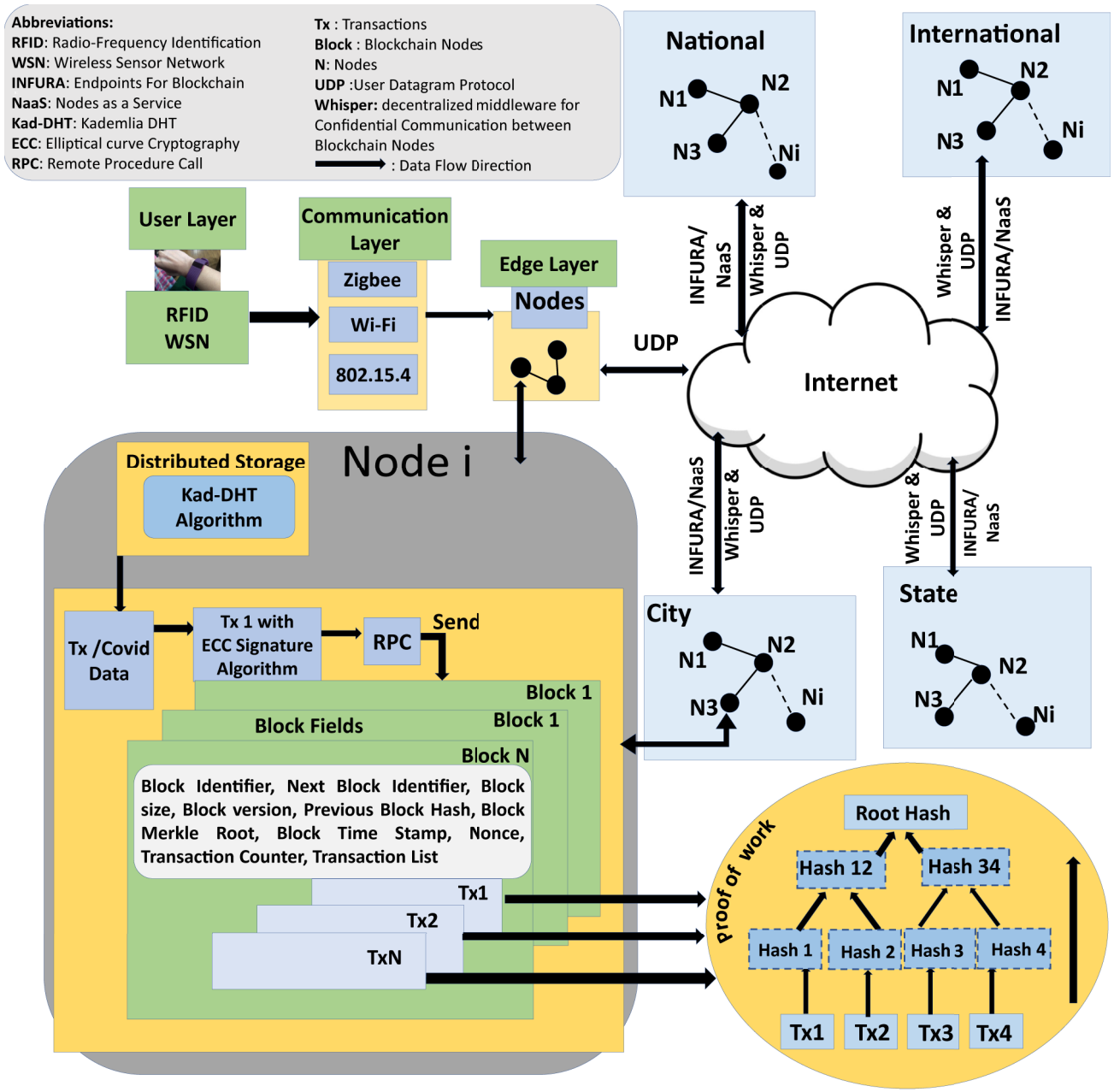


Fig. 5 Proposed detailed layered architecture for the CoviChain

The default time for waiting to successfully send the data is more than 145 s. If data is not sent, it is retransmitted multiple times but with limited retransmission permissions with same message ID's. If a partial message is sent from the source, then the edge rejects the message, prompts message error, sends reset message or ignores silently. The steps of sending data from source to edge is given in Algorithm 1.

Algorithm 1 To load the Covid data from Source to Edge Machine.

$S \leftarrow 0$
Each User U has a Public and Private key pair (PuU, PrU)to identify the User uniquely o Blockchain.
if User is Authenticated **then**
    Register Source Device by providing identifier for the device.
else
    End the process.
    Source Endpoint (IoT device) collects Data through Sensors.
    Both Source and Edge have their own respective Public and Private Keys (PuS, PrS) and (PuE, PrE).
    Edge Machine requests for the data upload in REQUEST CODE from Source in Confirmable format with DTLS Security for mutual authentication.
    The signature is appended to the REQUEST when request message is signed using the Private Key (PrE) of the Edge Device.
    The REQUEST along with signature is encrypted using public key of the Source (PuS).
    The Source uses the Edge's public key to verify that Edge node is genuine.
    Decryption of the request and signature verification is done by source.
    if REQUEST is authenticated and validated **then**
        Source sends response in RESPONSE CODE in NON- Confirmable format with DTLS and unique Message ID to the Edge Machine.
        if Data is Sent partially from the Source to analyze **then**
            Edge Machine must reject the Message.
            Message Format Error.
            Send Reset Message.
            Ignore Silently if Partial data uploaded.
            if NONLIFETIME is greater than 145 Seconds **then**
                Reuse Message ID for other Data Transmissions from source.
            end if
        end if
    else
        The Data sent was successfully received, understood and accepted by Edge Machine.
    end if
end if
Repeat the Steps from 1 through 28 every time there is a new data available from Source.

The data from the source gets stored in files and converted to a buffered file before publishing on to distributed storage. With the help of a web application, the functions of smart contract are invoked to upload the buffered file on to distributed storage. With the help of public and private keys of edge and admin nodes, the file is signed and encrypted to be published on distributed storage through a smart contract. By calling publish smart contract, the encrypted file is published along with the blockchain address to the distributed storage. An encrypted file is always taken as input and the hash of the encrypted data is given as output from storage. The private key of the user is used to interact with the smart contract to perform data access. A hash map is used by the smart contract that maps devices owned by the users to the owner's address on the blockchain. The device provides owners, addresses of owners and device ID as a key to the hash map. Along with hashed encrypted data, the hash map is written on to the blockchain using the smart contract function and checks the validity of the data through a Write Access Policy (WAP). Algorithm 2 gives all the steps to load data from the edge to distributed storage to blockchain. If the device owner and owner address corresponds to the device ID, then the smart contract executes the write operation and appends the encrypted data on to the blockchain. If they don't match the write operation is discarded.

Algorithm 2 Data from Edge Machine to Distributed Storage to Blockchain.

$S \leftarrow 0$
Edge Machine stores data from source in files.
The files are converted in to a buffered file before publishing on DS.
The web Application calls a function on a smart contract to upload the buffered file on to distributed storage.
The Edge device and Admin Node have their own public and private keys (PuE,PrE) , PuAN, PrAN) respectively to uniquely identify devices.
The file should be signed by private key of Edge Machine (PrE) and public key of Admin Node (PuAN) before publishing on Distributed Storage (DS).
if Data is signed and encrypted **then**
    Publish encrypted data on Distributed Storage (DS), using IPFS client or Smart Contract.
    The smart contract function is invoked by the User Web application.
    The Smart contract calls a function called Publish Smart Contract to publish on Distributed Storage with the Blockchain Address where Encrypted data is taken as input and obtaining Hash of the Encrypted Data as output.
    The private key of the User (PrU) is used to interact with smart contract to perform data access.
    Smart contract uses Hash map that maps devices owned by users to the owners address on the Blockchain
    Device provides Owners, Address and Device ID as key in Hash map along with encrypted Hashed data to be written on Blockchain.
    The Hash of the Encrypted Data is published on to Blockchain using Smart contract function and checks validity of the data with write access policy.
    if Device owner, Address corresponds to device ID **then**
        Execute the Write operation
    else
        Discard Write operation
    else
        End the Process
    end if
end if
Repeat the steps from 1 through 22 every time there is new file with new data upload from source to Edge.

Algorithm 3 shows all the steps that take place during the process of accessing data from the blockchain.The data access request is sent by the requester. Each admin and edge node has its respective public and private Keys. The data access request is signed by the private key of the requester

and encrypted by the public key of the admin node. The request is decrypted and message integrity is checked through a signature. If the signature matches, the requester requests permission to read the data. The smart contract hash map maintains all the device owners, addresses and IDs along with registered nodes. If the requester's device owner, device address, device IDs and registered nodes match the smart contract hash map then the requester can access the data to be read with Read Access Policy (RAP).

Algorithm 3 Process of accessing Data from Blockchain.

```
S ← 0
Edge device and Admin Node have their Public and Private Keys (PuE, PrE) and (PuAN, PrAN) respectively.
The data access request is sent by Requester.
Data access request is signed by Requester's private key (PrR) and signature is appended along with request data.
Data access request along with the signature is encrypted by public key of Admin Node (PuAN) and published through Smart contract Client Program.
The Request is decrypted by Admin Node and verifies the message integrity using the signature.
if there is a match in signature then
    The Requester will request for permission to read the data.
    The Requester will provide Device Owner, Device Address and Device ID.
    Smart contract maintains a Hash map that contains Device owner, Device address, Device ID as key along with City/County Nodes, State Nodes, National Nodes and global Nodes registered.
    if Requestor's Device Owner, Device Address and Device ID matches Smart Contract Hash map then
        Requester can access the data to read.
        if Requestor's City/County Nodes match Smart Contract Hash map then
            City/County Nodes can access the Data to read
            if Requestor's State Nodes match Smart Contract Hash map then
                State Nodes can Access the device Data to read.
                if Requestor's National Nodes Match Smart Contract Hash map then
                    National Nodes can Access the device Data to read.
                    if Requestor's Global Nodes Match Smart Contract Hash map then
                        Global Nodes can Access the device Data to read.
                    else
                        Data Access Denied.
                    else
                        End the process.
                    end if
                end if
            end if
        end if
    end if
end if
Repeat the steps from 4 through 38 every time there is a new data access request.
```

Implementation of CoviChain

Specific Architecture Implementation of CoviChain

CoviChain is designed with the help of IPFS and Ethereum smart contracts (SC) where we can devise the rules according to which the communication can take place in transferring the results to the cloud and store data by two-way authentication for increased security and privacy. The readings from a wearable device pass the edge node, where the Interplanetary File system (IPFS) protocol combined with Ethereum smart contracts reside. The components interact with each other to authenticate themselves. The architecture of the proposed system is given in Fig.6

Ethereum client (EC) This part implements the full functionality that is required to connect and participate in the BC network. It connects to the peer to peer network, sends transactions and performs programming in detail.

Smart Contract (SC) These are building blocks used to create BC applications. They are programs that can be written with source code and installed in to the chain. The programming language used to write logic in smart contracts is Solidity which is immutable: once they are deployed, the code cannot be updated like a normal application.

Inter Planetary File System (IPFS) It is used for storing large amounts of data and uses a decentralized network to make data available with or without an Internet connection. It performs hash on the actual content to make the data converted in the form of immutable links which can then be passed on to the BC avoiding direct data insertion in to the chains.

Experimental Validation and Results

The implementation uses technologies such as smart contracts, ReactJS, IPFS, and the development tools such as Truffle, Ganache, Metamask, Web3 Provider and Ropsten Test Network. Truffle acts as an end-to-end development tool that provides an environment for writing, compiling, deploying, and testing smart contracts. Ganache, a private blockchain environment, acts as a mirror to the Ethereum blockchain to interact with smart contracts. It has extensive built-in Block Explorer, advanced mining control, and displays blockchain log output. Metamask, an instance of web3, is a browser extension that functions similar to the Ethereum wallet. Web3 provider uses JavaScript library from the official Ethereum JavaScript API, mainly to interact with smart contracts and connect to the blockchain network and the Rospsten test network to test the transactions live (Fig. 7).

As it is expensive and time consuming process to store larger files, we implement the blockchain file storage using IPFS. Similar to Ethereum, IPFS uses nodes to store files in a distributed way which cannot be tampered or modified. The file is stored and identified by its hash.

In this application, we upload a processed COVID-19 data file coming from the edge to the browser application. From the front-end, we submit the COVID-19 file to the IPFS and store it. Once the file is stored, the hash of the file is returned to the browser console. The hash generated from IPFS is stored on the blockchain, instead of the actual file. With this way we have tried to reduce the cost and time taken to store the larger COVID-19 files on the blockchain. The user interface of the application is given in Fig.7a. The hash is updated for each and every file uploaded from the form.

The client side of the application is implemented using React js. The Form validates Username and Password to sign up the user and takes the processed COVID-19 data file from the edge to store it on the blockchain. Once processed, the COVID-19 file is sent from the form, it is converted to Buffer format, and gets submitted to the IPFS, returning the hash of the file to the user in the console, as shown in Fig. 7b and c. Metamask is connected once the file is submitted to IPFS and acts as an Ethereum wallet asking to confirm the transaction to store COVID-19 file on the blockchain, as illustrated in Fig. 7d. The blockchain acts as a back-end database to store the file sent from the front-end form.

The back end is designed by writing the smart contract in the Solidity language. Both "write" and "read" functions are used in order to interact with the blockchain for writing and reading the file respectively. Firstly, the smart contract is compiled, tested, and migrated within the local machine through truffle and ganache. Ganache acts as a mirror to the actual Ethereum blockchain and comes with ten permitted accounts to start developing distributed applications. Once a file is uploaded from the front-end, a hash-ID generated through IPFS is compared with hash stored on ganache to check that they are similar, as shown in Fig. 8a. This comparison is required because once the smart code moves to the back-end chain, it cannot be recalled. Ganache verifies the application in a local system. For real-life file transactions we use Ropsten Testnet. The COVID-19 data folder is deployed onto the test network, as shown in Fig. 8b. Next, we measure the deploying times to evaluate the performance of the current model. Instead of storing an actual COVID-19 data file, a hash-ID is taken as input for attaching to the blockchain. A transaction hash is generated after deploying on the test network which is used to get the transaction details along with mining times as given in 8c. Mining time is the length of the time it takes to validate a new transaction. In the current paper, the hash of the file is the new block, and the time taken to authenticate is calculated in the Rospten network, which is presented in Fig. 9a. The contents for the transaction history are explained in "Datasets".

Application and the Testnet are connected by modifying a truffle configuration file and inserting Web3 Infura API Key in it [24]. By inserting the API key, we get access to the Ropsten blockchain Explorer to migrate the files . The application executes successfully by taking in different file formats of the COVID-19 data from the edge and converting them to hash by using the IPFS decentralized storage system to store on the blockchain with the help of smart contracts.

Datasets

The Edge we use here is an Intel(R) Core (TM) i5-8250U CPU @1.60 Ghz that sends the processed COVID-19 file to the application to get stowed in the Covichain. The COVID-19 Data sets for testing and validating are taken

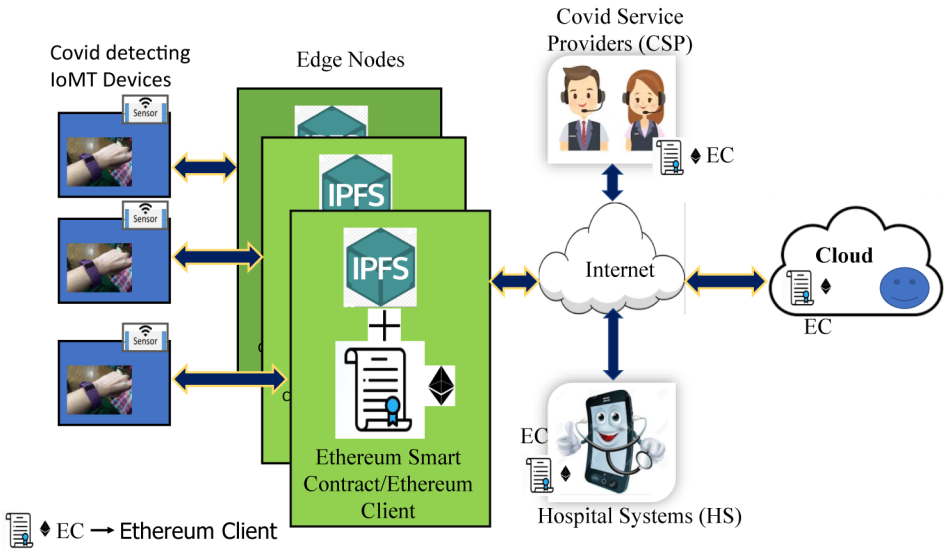
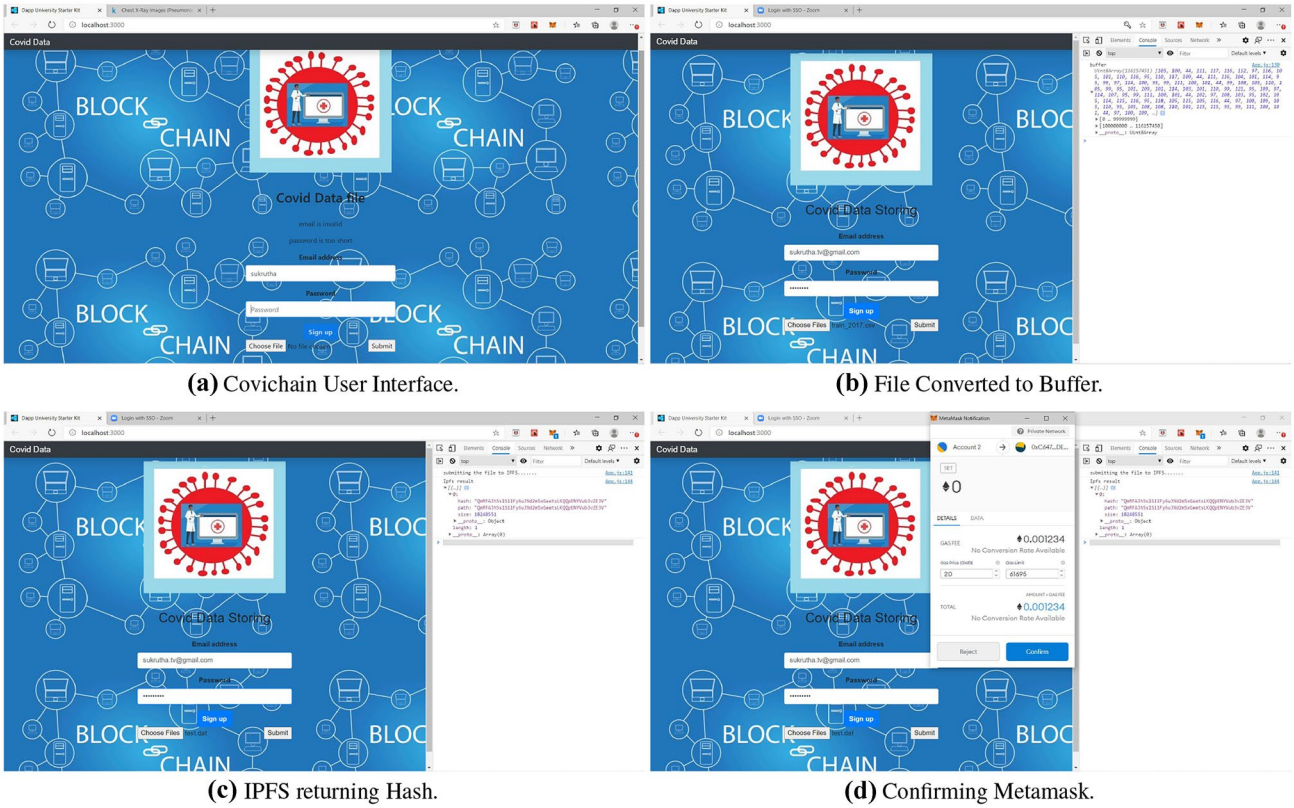


Fig. 6 Proposed implementation of CoviChain



(a) Covichain User Interface. **(b)** File Converted to Buffer.

(c) IPFS returning Hash. **(d)** Confirming Metamask.

Fig. 7 The user interface for the proposed CoviChain

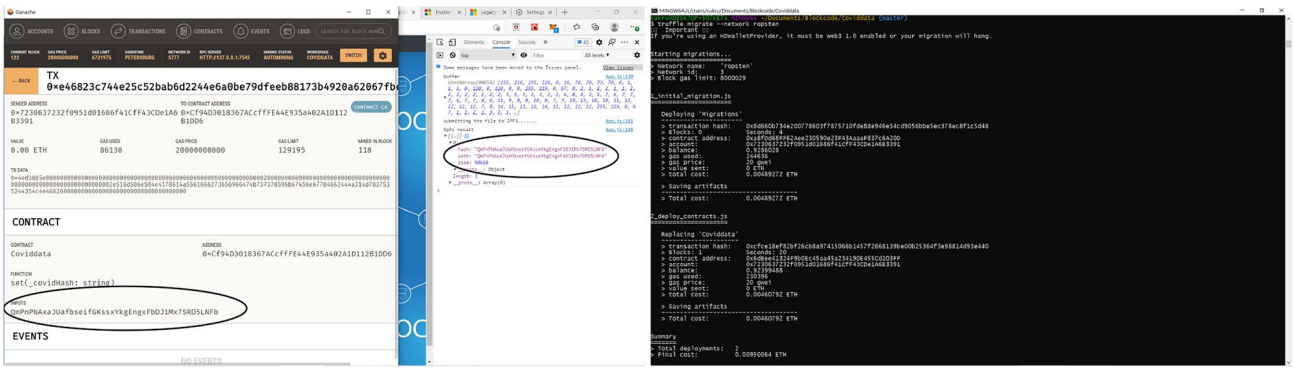
from the Kaggle site. Once the file IPFS hash is obtained, it is uploaded to the Block Explorer, Ethereum Ropsten Testnet which uses a similar protocol as Ethereum Mainnet. The Explorer is mainly used for testing the distributed applications. On the Testnet the specifics of Block Height, Timestamp, Transactions, Miner Address with Time taken to Mine, Block Reward, Uncles Reward, Total Difficulty, Difficulty, Gas Used, Gas Limit, Gas used by the Transaction, Nonce with Input Data are observed and noted for the validation of the results. One of the advantages of using blockchain for storage is it generates a timestamp for the inputted data. If only IPFS is used for distributed storage, we do not know when the data was added because it does not show timestamp while, if only the blockchain is used, the increased size of the data to be stored rises latency issues and cost [25]. In this paper we have successfully combined both technologies and reduced the cost and time along with inserting timestamp for the data stored. The MedRec [15] application uses traditional Ethereum blockchain to store the Electronic Health records by implementing smart contracts.

The times for data in MedRec are calculated assuming the mining time of the conventional Ethereum blockchain to be 13 s for 1MB Data [26, 27]. Details of performance information CoviChain which is undertaken for various datasets is presented in Table 3. As of 3/12/2021 the price of Ethereum Stands at $1811.41, taken from recent daily Ethereum YCharts [27] and to store 1 KiloByte of information it takes 0.032ETH [28].

We calculated the mining times and transaction fees for [15] based on the facts collected and compared between Med Rec [15], conventional blockchain and proposed CoviChain which are presented in Fig. 9b. A comparison of cost for different blockchains in presented in Table 4. The CoviChain Application residing on the edge can be enhanced and added to the IoMT devices to collect the data and upgraded to send up to 10GB of Data on to IPFS [29], and to store on blockchain, where the time and cost is substantially reduced if related to time and cost when sending the actual medical statistics.

Conclusion and Future Work

The application is successfully built to take the processed COVID-19 data file from the edge and store the file on blockchain to retrieve the information from the client side. The work presented here uses IPFS and moves the file hash to the public blockchain using smart contracts. As the storage of bigger statistics is expensive on blockchain, we have



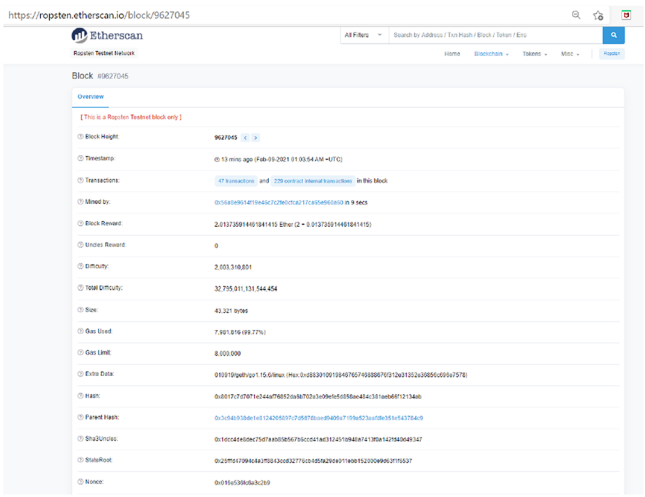
(a) Hash Sent to the Local Ganache Blockchain. **(b)** Deploying to Ropsten.

(c) Transaction Details.

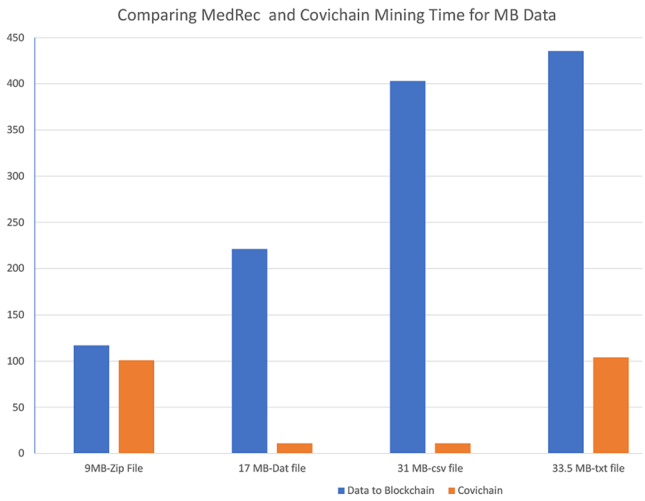
Fig. 8 Functional verification of the proposed CoviChain

Table 3 Performance evaluation of CoviChain for different datasets

File	File size	Deploy time to Ropsten Testnet (s)	Mining time (s)	IPFS hash or distributed storage	Tx hash or blockchain hash	Tx fees
.jpeg	88.5 KB	12	10	QmPnPNAxaJUa fbseifGKssx-YkgE ngxFbDJ1Mx7SRD5L-NFb	0xb3586a12af23183ae9 cee11579aa291b06271f311 c05807872ac78843b74d52d	0.00542016 ETH
.png	224 KB	28	7	QmYbwUNvASes Rq87ADy-amqUwgMJ 9pkU9n58VhAYm-VMpKBk	0xcd2f9678436ba6a0a8 4c770a8df8f5b4d7c5dc7b9 989f4a7e03add14af5cc27e	0.00460792 ETH
.dat	17 MB	28	11	QmRFAJh5s1S1 1Fy6uJNd2mSx-Gee tsLKQQpENYVub3vZE3V	0x9c24d52e0def680eff 5368a652be6006b68366c07 cd36384acde635ac02e0264	0.00481952 ETH
.csv	31 MB	40	11	QmVppTpmR9ed h2GoxfwD-f9kxf7S iouohfTLYNtpRtbvhPz	0x7148323de8055c4517 a23e08a9a2ee42f25824d0e 57ba7bfa60d2ab2577fc991	0.00474598 ETH
.txt	33.5 MB	28	104	QmXG46eZ7K9V UqDeUo3P-WFfJ4By 3p96F5iYqVSb-fUaEkGW	0x88ec9224cc19497ab70 2e2bbd-3cd5dd1fd4a13eb1 85b8bc0bf-98b802f889fe2a	0.00254896 ETH
.zip	9 MB	8	101	QmTejx1kq8X8 hHWctpK8Hb-C7GMq fTjmea2VbEzN4TEk-WFG	0xf4767a03aeb58ac9955 dd729ae592f309e2365552 28d1b486ca6099e416758a0	0.00356897 ETH
.pdf	644 KB	27	8	QmTsfkbnNbvb U8ZmiPcPs-6bRdfN dtErMKAzUYwLh-PEm8Xy	0xd48b000e923f7bb742d0 3e145c31b0ac297275257 eff-51636ca1e049654d3454	0.00478956 ETH



(a) Mining Time of the Transaction.



(b) Time Comparative Graph for MB Data between Conventional Blockchain and Currently Proposed CoviChain in Seconds.

Fig. 9 Experimental results for CoviChain for various datasets

Table 4 Cost comparison for MB data between conventional blockchain and current Covichain

File	File size (MB)	Tx fees in CoviChain	Cost in Dollars	Tx fee in TB-MedRec [15]	Cost in Dollars
.dat	17	0.00481952 Eth	8.8	272 Eth	492,592
.csv	31	0.00474598 Eth	8.6	490.5 Eth	888,496.6
.txt	33.5	0.00254896 Eth	4.7	490.5 Eth	681,180.7
.zip	9	0.00356897 Eth	6.5	167.4 Eth	303,228.3

1 Eth 1811.41 Dollars, *1 KB* 0.032 Eth, *1 MB* 32.768 Eth, *TB* traditional blockchain

used data on off-chain in a distributed way since central storage has latency, loss of content by single point failure and compromised privacy of users. The data upload and mining time on to blockchain is significantly decreased as we are storing hash output coming from IPFS instead of actual content. The proposed application is a precise and cost-effective solution and useful for contact tracing and medical data storing and retrieval by only authorized personnel in a secure way. For future research the application can be enhanced to upload larger medical files with images with more data to be stored on a blockchain with lower cost and decreased energy consumption.

Acknowledgements This work was supported in part by the National Science Foundation under Grant OAC-1924112. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Declarations

Conflict of interest The authors declare that they have no conflict of interest and there was no human or animal testing or participation involved in this research. All data were obtained from public domain sources.

References

1. Tripathy AK, Mohapatra AG, Mohanty SP, Kougianos E, Joshi AM, Das G. EasyBand: a wearable for safety-aware mobility during pandemic outbreak. IEEE Consum Electron Mag. 2020;9(5):57–61. https://doi.org/10.1109/MCE.2020.2992034.
2. Tidy J. Coronavirus: Israel enables emergency spy powers. 2020. https://www.bbc.com/news/technology-51930681. Accessed 12 Apr 2021
3. Adanur B, Bakir-Güngör B, Soran A. Blockchain-based fog computing applications in healthcare. In: 2020 28th Signal processing and communications applications conference (SIU), pp. 1–4;2020. https://doi.org/10.1109/SIU49456.2020.9302168
4. Cech HL, Großmann M, Krieger UR. A fog computing architecture to share sensor data by means of blockchain functionality. In: 2019 IEEE international conference on fog computing (ICFC), pp. 31–40; 2019. https://doi.org/10.1109/ICFC.2019.00013
5. Egala BS, Pradhan AK, Badarla VR, Mohanty SP. Fortified-chain: a blockchain based framework for security and privacy assured internet of medical things with effective access control. IEEE

Internet Things J. 2021. https://doi.org/10.1109/JIOT.2021.3058946.
6. Mohanty SP, Yanambaka VP, Kougianos E, Puthal D. PUFchain: a hardware-assisted blockchain for sustainable simultaneous device and data security in the internet of everything (IoE). IEEE Consum Electron Mag. 2020;9(2):8–16. https://doi.org/10.1109/MCE.2019.2953758.
7. Puthal D, Mohanty SP, Kougianos E, Das G. When do we need the blockchain? IEEE Consum Electron Mag. 2021;10(2):53–6. https://doi.org/10.1109/MCE.2020.3015606.
8. Rachakonda L, Bapatla AK, Mohanty SP, Kougianos E. SaYoPillow: blockchain-integrated privacy-assured IoMT framework for stress management considering sleeping habits. IEEE Trans Consum Electron. 2021;67(1):20–9. https://doi.org/10.1109/TCE.2020.3043683.
9. Runzel MA, Hassler EE, Rogers R, Formato G, Cazier JA. Designing a smart honey supply chain for sustainable development. IEEE Consum Electron Mag. 2021. https://doi.org/10.1109/MCE.2021.3059955.
10. Shin Y, Kim S, Chung JM, Chung HS, Han SG, Cho J. Emergency department return prediction system using blood samples with LightGBM for smart health care services. IEEE Consum Electron Mag. 2021;10(3):42–8. https://doi.org/10.1109/MCE.2020.3015439.
11. Leprince-Ringuet D. Contact tracing: data shows apps can help in fight against COVID, say researchers. 2020. https://www.zdnet.com/article/contact-tracing-data-shows-apps-can-help-in-fight-against-covid-say-researchers/. Accessed 12 Apr 2021
12. Lodders A, Paterson JM. Scrutinising COVIDSafe: frameworks for evaluating digital contact tracing technologies. 2020. https://journals.sagepub.com/doi/full/10.1177/1037969X20948262. Accessed 12 Apr 2021
13. Nisar S, Zuhaib MA, Ulasyar A, Tariq M. A privacy-preserved and cost-efficient control scheme for coronavirus outbreak using call data record and contact tracing. IEEE Consum Electron Mag. 2021;10(2):104–10. https://doi.org/10.1109/MCE.2020.3038023.
14. Patel FB, Jain N, Menon R, Kodeboyina S. Comparative study of privacy preserving-contact tracing on digital platforms. In: Proceedings of international conference on computational intelligence (ICCI), pp. 137–141; 2020. https://doi.org/10.1109/ICCI51257.2020.9247782
15. Azaria A, Ekblaw A, Vieira T, Lippman A. MedRec: using blockchain for medical data access and permission management. In: 2016 2nd International conference on open and Big Data (OBD), pp. 25–30; 2016.
16. Seliem M, Elgazzar K. BIoMT: blockchain for the internet of medical things. In: Proceedings of IEEE International Black Sea conference on communications and networking (BlackSeaCom), pp. 1–4; 2019
17. Meng W, Li W, Zhu L. Enhancing medical smartphone networks via blockchain-based trust management against insider attacks. IEEE Trans Eng Manag. 2020;67(4):1377–86. https://doi.org/10.1109/TEM.2019.2921736.
18. Garg N, Wazid M, Das AK, Singh DP, Rodrigues JJPC, Park Y. Bakmp-iomt: design of blockchain enabled authenticated key management protocol for internet of medical things deployment. IEEE Access. 2020;8:95956–77. https://doi.org/10.1109/ACCESS.2020.2995917.
19. Yu K, Tan L, Shang X, Huang J, Srivastava G, Chatterjee P. Efficient and privacy-preserving medical research support platform against COVID-19: a blockchain-based approach. IEEE Consum Electron Mag. 2021;10(2):111–20. https://doi.org/10.1109/MCE.2020.3035520.
20. Haque S, Aziz S, Rahman M. Review of cyber–physical system in healthcare. Int J Distrib Sens Netw. 2014;2014:20. https://doi.org/10.1155/2014/217415.
21. Puthal D, Malik N, Mohanty SP, Kougianos E, Das G. Everything you wanted to know about the blockchain: its promise, components, processes, and problems. IEEE Consum Electron Mag. 2018;7(4):6–14.
22. Hoffman C. What is Ethereum, and what are smart contracts? 2018. https://www.howtogeek.com/350322/what-is-ethereum-and-what-are-smart-contracts/. Accessed 12 Apr 2021
23. Saini V. What is IPFS? 2019. https://hackernoon.com/understanding-ipfs-in-depth-1-5-a-beginner-to-advanced-guide-e937675a8c8a. Accessed 18 Apr 2021
24. Jelly. How to get Infura API Key. 2020. https://ethereumico.io/knowledge-base/infura-api-key-guide/. Accessed 12 Apr 2021
25. Ober M. The decentralized power of Ethereum and IPFS how to create immutable files. 2018. https://medium.com/pinata/ethereum-and-ipfs-e816e12a3c59. Accessed 12 Apr 2021
26. Kenton W. Block time. 2020. https://www.investopedia.com/terms/b/block-time-cryptocurrency.asp. Accessed 12 Apr 2021
27. Patton M. Under the Hood: What You Should Know About the YCharts Tool. 2014. https://www.thinkadvisor.com/2014/12/15/under-the-hood-what-you-should-know-about-the-ycharts-tool/. Accessed 12 Apr 2021
28. Wood G. Ethereum: a secure decentralised generalised transaction ledgereip 150 revision. 2017. http://gavwood.com/Paper.pdf. Accessed 12 Apr 2021
29. Tak. IPFS Production Configuration. 2019. https://medium.com/coinmonks/ipfs-production-configuration-57121f0daab2. Accessed 12 Apr 2021