Fortified-Chain: A Blockchain Based Framework for Security and Privacy Assured Internet of Medical Things with Effective Access Control

Bhaskara S. Egala, Ashok K. Pradhan, Venkata R. Badarla, *Senior Member, IEEE*, and Saraju P. Mohanty, *Senior Member, IEEE*

***Abstract*—The rapid developments in the Internet of medical things (IoMT) help the smart healthcare systems to deliver more sophisticated real-time services. At the same time, IoMT also raises many privacy and security issues. Also, the heterogeneous nature of these devices makes it challenging to develop a common security standard solution. Furthermore, existing cloud-centric IoMT healthcare systems depend on cloud computing for electrical health records (EHR) and medical services, which is not suggestible for a decentralized IoMT healthcare systems. In this paper, we have proposed a Blockchain-based novel architecture that provides a decentralized EHR and smart-contract based service automation without compromising with the system security and privacy. In this architecture, we have introduced the hybrid computing paradigm with blockchain-based Distributed Data Storage System (DDSS) to overcome blockchain-based cloud-centric IoMT healthcare system drawbacks like high latency, high storage cost and single point of failure. A decentralized Selective Ring based Access Control (SRAC) mechanism is introduced along with device authentication and patient records anonymity algorithms to improve the proposed systems security capabilities. We have evaluated the latency and cost effectiveness of data sharing on proposed system using Blockchain. Also, we conducted a logical system analysis, which reveals that our architecture based security and privacy mechanisms are capable of fulfilling the requirements of decentralized IoMT smart healthcare systems. Experimental analysis proves that our Fortified-Chain based H-CPS needs insignificant storage and has a response time in the order of milliseconds as compared to traditional centralized H-CPS while providing decentralized automated access control, security, and privacy.**

***Index Terms*—Internet of medical things (IoMT), Healthcare Cyber-Physical System (H-CPS), Blockchain, Mutual Authentication, Access Control, Privacy, Distributed Data Storage System (DDSS), Hybrid Computing**

I. Introduction

THE Internet of medical things (IoMT) is an integrated embedded system of software, hardware, network access, and sensor/actuators [1, 2]. As these systems are more sophisticated and interfere with critical healthcare operations, due to which it raises many security and privacy issues. However, IoMT technology revolves at a greater speed, yet majority of devices are resource-constrained and limits us from considering the high-end mechanism for security and privacy perspectives. Regardless of having multiple protocols and standards for IoMT ecosystems, it lacks in security and privacy issues [3]. In addition, the classical cloud-centric healthcare systems have inherent problems like a single point failure, lack of transparency, low level of control over personal data, and high latency. Because of less availability of medical service professionals, the healthcare industry is unable to provide critical healthcare services to large number of patients [4] during pandemic time. These particular technical challenges are achievable with the help of a perfect combination of protocols, mechanisms, and enhanced system architecture. The remote patient monitoring system (RPMS) helps the healthcare service providers to eliminate unnecessary engagement of professionals in regular consultancy and provides more time for understanding the patient's health issues to improve the patient health status. It is suggestible to have a parallel supporting system to automate primary healthcare services to handle pandemics with limited healthcare professionals.

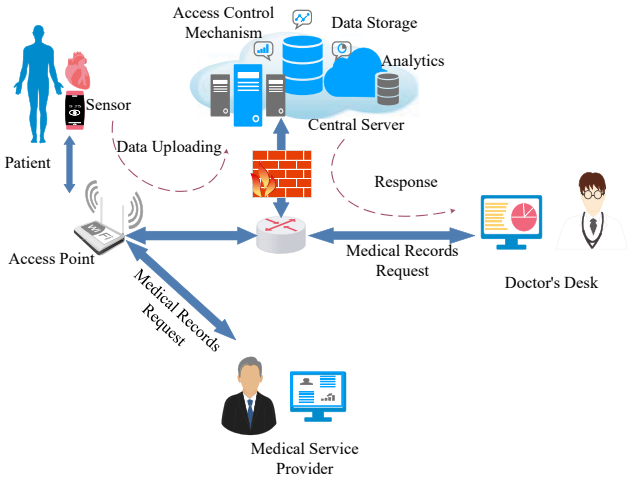


Fig. 1: Overview of a Smart Healthcare System or Healthcare Cyber-Physical Systems (H-CPS).

A cloud-centric H-CPS architecture is illustrated in Fig. 1, where the patient's data is transferred to a cloud for data processing and analysis [5]. However, this model is not a suitable option for patient-centric time-critical healthcare systems [6] which requires low latency. Moreover, they are vulnerable to single point of failure and Denial of Service

B. S. Egala is with the Department of Computer Science Engineering, SRM University, AP, India. e-mail: bhaskara_santhosh@srmap.edu.in.

A. K. Pradhan is with the Department of Computer Science Engineering, SRM University, AP, India. e-mail: ashokkumar.p@srmap.edu.in.

V. R. Badarla is with the Department of Computer Science Engineering, Indian Institute of Technology, Tirupati, India, e-mail: ramana@iittp.ac.in.

S. P. Mohanty is with Department of Computer Science & Engineering, University of North Texas, TX, USA, e-mail: smohanty@ieee.org.

Manuscript received XX, XXXX; revised YY, YYYY.

attacks (DoS) attack that leads to service unavailability. As a result, healthcare service providers migrate to a decentralized community managed frameworks. These are more patient-centric and provides transparent healthcare services on decentralized architectures. Present cloud-centric IoMT healthcare architecture models guarantees event traceability, security and low cost maintenance. However, it still lacks in privacy, transparency, data ownership, and community control mechanism. To overcome these important issues, the new generation systems utilize robust advanced technologies like blockchain, distributed data storage systems (DDSS), and hybrid computing.

The blockchain technology is a decentralized distributed ledger system which provides smart-contracts, and exhibits traceability, transparency in digital asset management [7]. The transactions in blockchain are represented as blocks linked together to form a chain of blocks. If one block or transaction is forced to alter, then we need to change the entire chain header information of that blockchain. The transaction integrity is maintained by using Merkel tree mechanism. However, use of blockchain technology for IoMT or H-CPS is not straightforward due to several deficiencies in the original blockchain, such as lack of scalability and high computational demand [8]. The smart-contracts automates event-driven actions without intervention of a third party to provide a cost-effective automation solutions [9]. In order to mitigate large data storage problem of blockchain, we have adopted a distributed data storage system (DDSS) named as InterPlanetary File System (IPFS) [10]. The chosen DDSS provides a content-centric peer-to-peer faster data sharing. It uses data caching and file versioning to maintain multiple documents with same name. However, when a large size file is uploaded to DDSS, it breaks the file into multiple objects of 256kb and connects all these objects to an empty object to retrieve the complete file using Distributed Hash Tables (DHTs) [11].

Another major issue is high latency in real-time data utilization. A proposed novel hybrid computing model focuses on low latency and real time data utilization by combining the edge and cloud computing topologies [12]. However, sending critical data to cloud for data processing and analysis requires high bandwidth and 24/7 connectivity which offers high latency and cost [13]. On the other hand, edge computing provides computational capability at the edge of the network to reduce the latency and eliminates the high bandwidth cost. Therefore, in our work we have designed hybrid model to gain benefits from both the technologies.

The remainder of the article is organized as follows. Section II discusses the contribution of the paper and Section III illustrates the related works in the field of smart healthcare systems. Section IV details the system three layer architecture and Section V represents system architecture overview. The Section VI describes the cryptographic methods of our proposed Fortified-Chain model. The logical analysis of Fortified-Chain is interpreted in Section VII. Section VIII represents the experimental analysis like latency and storage cost. Finally, conclusions of our work is illustrated in Section IX.

II. Contributions of the Current Paper

A. The Research Problem Addressed in the Current Paper

- High latency in real-time data utilization and analysis in critical healthcare systems.
- The classical system drawbacks like centralize data processing, data immutability, and third party trust issues.
- The vulnerabilities related to data privacy and security in decentralized systems.
- Scalability issues of blockchain in larger data storage and management.
- Lack of stand-alone transparent service automation system for critical healthcare services.

B. The Challenges in Solving the Problem

- The leveraging of blockchain technology to H-CPS is a complex process because it demands for high-end system resources.
- Storing large data in blockchain is a costly operation and produces high latency.
- Though the smart-contracts are useful in event based automation still it lacks in intelligence to support real-time system service automation.
- Leveraging of distributed data storage system and hybrid computing with blockchain is a complex process that involves more mechanisms, protocols, and standards.

C. Novel Contributions of the Current Paper

- A novel architecture model is proposed for decentralized H-CPS based healthcare systems to support low latency services along with real-time patient monitoring using blockchain, DDSS, and hybrid computing framework.
- The proposed system introduces Selective Ring based Access Control (SRAC), Patient Anonymity and Device Authentication algorithms to support system level privacy and security.
- A draft model of user friendly smart digital agreements is introduced to create dynamic digital service agreements (smart contracts) between different parties.

III. Related Prior Works

The rapid development in IoMT based smart healthcare systems demand more stability and robustness in-terms of privacy, security, high availability, and low latency. In this section, we have introduced different works related to smart healthcare systems and trends in patient electronic health records (EHRs) access management. Further, we have divided these works into two categories, the table I represents the blockchain based non-healthcare applications and Table II represents works directly focused on smart healthcare EHR management. We have also highlighted few centralised works in this section to showcase the latest works in access control management, privacy and security in IoT.

Ying et al. [14] proposed a fine-grained access control mechanism for EHR sharing on the cloud by using attribute-based encryption mechanism. A smart contract based access control mechanism is proposed in [15] to share EHR among

TABLE I: A Brief Overview of Related Works on Blockchain for IoT Applications.

Prior Work	System Type	Computing Platform	Proposed Solutions	Description
Ying et al. [14]	centralized Data Sharing	Central Cloud	Attribute-based Access Control	General IoT applications
Xia et al. [15], Zhang et al. [16]	decentralized-Cloud based System	Cloud Computing	Blockchain-based Access control	General IoT applications
Nguyen et al. [17]	Cloud based distributed System	Mobile Cloud	Blockchain-based Access control	IoT-Data centric Applications
Wang et al. [18], Zhang et al. [19], Xu et al. [20], Novo et al. [21]	decentralized Data sharing	Cloud-Blockchain	Access Control System	IoT-Data centric Application
Huang et al. [22], Hassan et al. [23]	Distributed decentralized	Mobile edge computing	Edge technology leveraging	Time-Critical Applications

TABLE II: A Brief Overview of Related Works on Blockchain for Healthcare Applications.

Prior Work	System Type	Computing Platform	Proposed Solutions	Description
Gong et al. [24], Zhang et al. [25]	centralized	Cloud	Smart healthcare suggestions	General Healthcare Applications
Pace et al. [26]	decentralized	Hybrid computing	An Edge based architecture	Time-Critical Healthcare Applications
Ekblaw et al. [27]	decentralized	Server centric	EHR management	General Healthcare Applications
Dagher et al. [28]	decentralized	Cloud	Privacy preserving and security techniques	General Healthcare Applications
Gonalves et al. [29]	decentralized	Semi Blockchain network	Security techniques	General Healthcare Applications
Alkhushayni et al. [30]	centralized	Server-centric	Blockchain based data management	General Healthcare Applications
Angeles et al. [31]	centralized	Cloud	Blockchain integration	General Healthcare Applications
Ito et al. [32]	decentralized	Individual-centric	Privacy-Preserved Use of Personal Health Data	General Healthcare Applications
Niu et al. [33]	Semi centralized	Cloud	Searchable Attribute-Based Encryption	General Healthcare Applications
Yang et al. [34]	decentralized	Cloud	Attribute Cryptosystem for data sharing	General Healthcare Applications

different cloud-based parties. Zhang et al. [16] suggested an attribute based decentralized file access control system using blockchain technology. Nguyen et al. [17] proposed a secure blockchain for EHRs sharing on a mobile cloud, and also suggested trust worthy access control mechanism using smart-contracts. A decentralized access control model using an attribute-based encryption techniques on blockchain-based IPFS is suggested in [18]. The work in [19] illustrated smart contract based access control mechanism for distributed IoT systems. The author in [20] suggested a decentralized blockchain enabled capability-based access control mechanism for IoT system. Novo et al. [21] proposed blockchain based access control mechanism for high scalability in IoT critical system. The above mentioned works are primarily focused on access control mechanisms. However, the storage cost and latency related issues are not yet discussed.

A window-based rate control algorithm (w-RCA) is proposed by Ali Hassan [23] to optimize the quality of service in the mobile edge computing healthcare system. Pace et al. [26] proposed a human-centric architecture called BodyEdge for the emerging healthcare industry applications. Ramani et al. [35] proposed a theoretical framework for reliable sharing of EHR using blockchain technology. Liang et al. [36] proposed a patient-centric data sharing mechanism to enhance identity management and data privacy. The work in [37] suggested an improved Peer-To-Peer (P2P) file sharing scheme for IPFS using blockchain technology. As the cloud-based centralized systems are not suitable options for real-time critical systems, hence there is an intense urgency for edge computing technology solutions. Zhang et al. [25] proposed a programmable blockchain-based healthcare system to fulfil the gaps in report delivery and communication between different parties using Distributed applications (DApps). A decentralized record management system MedRec [27] is proposed for patient records management. Dagher et al. [28] introduced a privacy preserving framework called “Ancile” using blockchain technology. A secure storage and sharing environment for health data is suggested in [29]. Alkhushayni et al. [30] integrated blockchain technology into medical health records management. Angeles et al. [31] showcased a successful blockchain Proof-of-Concept (PoC) leveraging in healthcare. A blockchain empowered individual-centric privacy preserving framework is proposed in [32]. Niu et al. [33] and Yang et al. [34] proposed different schemes for secure health records sharing system using attributed encryption techniques on blockchain. Gia et al. [38] proposed a model to enhance the health monitoring systems using advanced techniques like embedded data mining, distributed storage, and notification services on fog computing. Biswas et al. [39] proposed an interoperability and synchronization management of blockchain based decentralized e-Health system. Rachakonda et al. [40] present a blockchain integrated H-CPS framework with stress monitoring as a specific example.

IV. Layered-Organization of Fortified-Chain

A. Organization of Fortified-Chain

In this paper, we have proposed a novel architecture and cryptography methods for IoMT-based decentralized distributed smart healthcare systems to provide data privacy, security, traceability, low latency, low storage cost and availability.

Our proposed architectural model is divided into three layers for easy system implementation and management. The first layer has data generators or IoT sensors like heart beat sensor, temperature sensor, etc. The second layer represents novel hybrid computing paradigm which combines the edge and cloud paradigms to gain benefits from both technologies. The third layer is data consumers layer which consists of actuators like smart insulin, caretaker robot, smart beds, etc. The system layer view is represented in the Fig. 2. In our proposed work, we suggested Datagram Transport Layer Security (DTLS) [41] protocol which ensures the secure communication between these three layers.

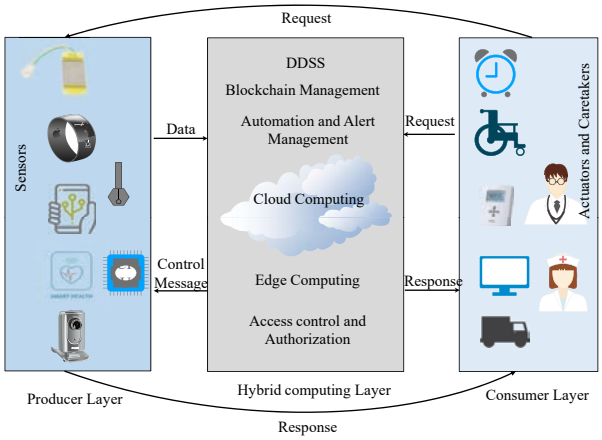


Fig. 2: Fortified-Chain in a Layered view.

In addition, we have also introduced three cryptography mechanisms in the form of algorithms to deliver system level privacy and security. Likewise, we combined blockchain technology and DDSS to achieve a decentralized EHRs transparent system. Our proposed model maintains a public ledger for each medical record and critical event to provide traceability. Moreover, system specific smart-contracts help medical professionals to perform event-based automation activities without human interference.

B. Different Layers of Fortified-Chain

The overall system of cryptographic operation flow starts with public and private key generation at every node of hybrid computing using Elliptic Curve Cryptography (ECC) [42] cryptosystem. All public keys from three layers are pooled and maintained in a blockchain based DDSS secure public key records. Subsequently, the device registration process at hospital level is initiated with the help of respective edge computers. Eventually, every patient, service providers and nodes get a specific set of identity code from the edge computer. Our proposed model uses device and actors grouping mechanism in which a set of sensors, actuators and actors are allocated to each patient. Every group gets a unique identity to combine sensors data for analysis and decision making. The sensor generated patient medical raw data is appended with additional information and encrypted with respective edge computer public key before it transfer to consumer layer. At the middle layer, hybrid computing performs the data processing and analysis followed by decision making. Moreover, it performs tasks related to data storage, access controlling, and anonymity along with blockchain based DDSS management. Also, the hybrid computing layer takes care of smart-contract creation and deployment. In the consumer layer, the actuators and terminals perform events defined by a specific smart-contract. In this paper, we have illustrated all Fortified-Chain functions and algorithms in an abstract format. The following sub sections give a detailed view of the three layers.

1) Data Producer Layer: The layer performs device registration and secure patient data transfer to the respective edge computer in a fixed time intervals. As, majority of IoT sensors are primarily resource constraint devices, hence, our architecture eliminates blockchain and DDSS operations overhead using edge devices as proxy for constrain devices. Due to this, the essential resources of the sensors is only used for patient data operations to provide long uninterrupted services. In our proposed system, the sensors generate patient's raw medical data (Dat_{raw}) and appends with supplementary information like service identity, device identity, digital signature, raw data hash value and time-stamp. Thereafter, the created data is encrypted with a receiving edge computer's public key and handovered to the edge computer for the data analysis.

2) Hybrid Computing Layer: A hybrid computing is distributed computing topology that brings computational and storage capability near to the data source along with robust features of cloud computing. In our proposed work majority of the critical operations of the system is performed by the edge computing topology where as cloud computing provides necessary additional and backup services. In this layer, the data received from sensors is decrypted and validated before data processing and analysis. Based on the data analysis, edge or cloud computer initiates the automation decisions to control the consumer layer devices. It synchronizes all patient non-critical global data and access rights records on main DDSS to project as a single layer. All the edge devices in the hospital select a leader edge device in a random fashion to maintain local cache of access control records, public key pool etc. This mechanism speed ups the data retrieval and automation process. The cloud computing on the other hand provides a selective services to remote patients. The incoming requests are further divided based on the request origin. The internal edge computing utilizes collective resource capability to handle massive incoming requests.

3) Data Consumer Layer: This layer consists of actuators, service provider terminals, emergency alerting systems etc. The nodes act as per the decisions received from the hybrid computing layer. Simultaneously, hospital level edge computer monitors and alerts node operational status like online or offline in a frequent time intervals to hybrid computing for smoother system operations.

V. Architecture of Fortified-Chain

A. Overview of the Proposed Architecture

In this section, we have introduced our proposed system "Fortified-Chain" architecture and its internal elements. The

organization level view of the proposed Fortified-Chain is represented in Fig. 3. We interconnected the organizations with blockchain based DDSS networks to achieve secure data sharing and remote patient monitoring. The system level DDSS network is logically divided into self contained sub-networks (local DDSS) for managing the local patient medical data at the hospital level. The local DDSS are created with the help of a sub-swarm key generation process [43]. Hence, the hospitals maintain the complete patient medical data on local DDSS network. A subset of patient's non-critical information is generated and published to main DDSS to support third party healthcare services. The global secure backup at cloud level ensures the system level non-critical data availability.

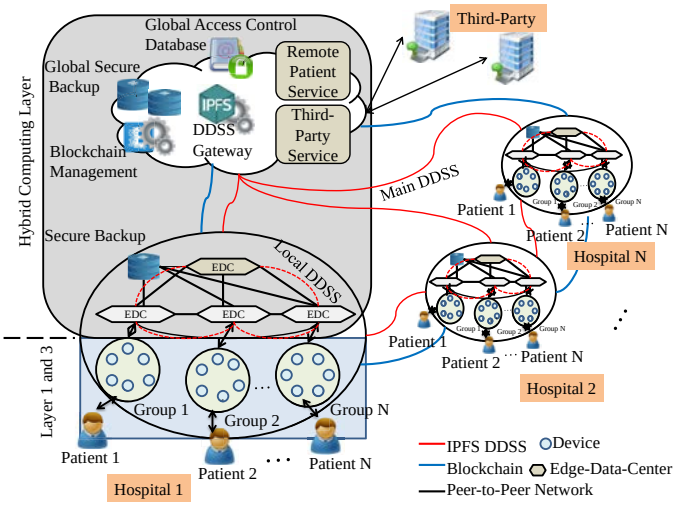


Fig. 3: Overview of Fortified-Chain Architecture.

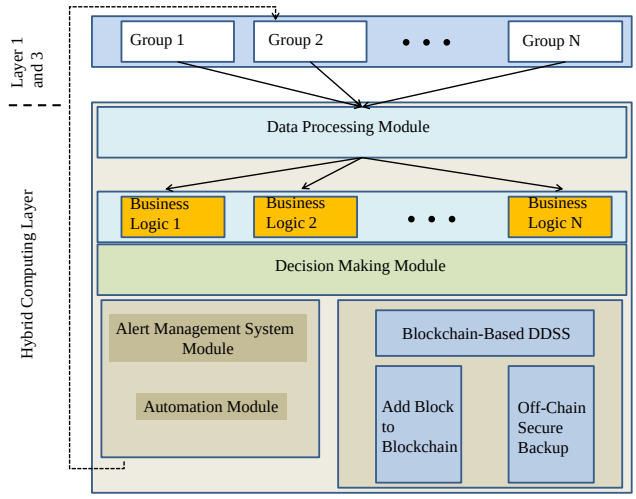


Fig. 4: Modules of hybrid computing in Fortified-Chain.

Fig. 4 represents the overview of the system internal working modules framework. It represents the way how internal modules are connected with each other and also shows the data flow from producer's layer to consumer's layer. The data processing module process and restructures the patient sensor data for data analysis purpose. The blockchain based business logic is an organization's propriety data analysis and decision making module, which computes patient's health status and take decisions to stabilise patient health. The alert management system module alerts the caretaker or doctors about emergency situations. The automation module triggers the event signals for layer 3 devices.

B. Elements of the Proposed Architecture

1) Sensors and Actuators: In our proposed Fortified-Chain architecture, every device have their own unique embedded identity (ID_{dev}). Moreover, it is very difficult to manage the system services using device embedded identity in big organizations. Due to this, every node register themselves with the organization's edge computers and receives a unique set of identities like device identity (ID_{div}) and service identity (SI_{div}).

2) Hybrid computing: In our proposed work, the edge and cloud computing paradigms are logically combined to perform system level critical operations. It generates decisions based on predefined business logic and then triggers actions by sending decisions to the consumer layer. Further, it create non-critical data from patient data for third party services. It Secures the organization level patient data by securely backuping at two locations, one at the cloud and another at hospital's data-centre, which provides additional security against data unavailability. The overall logical process of a hybrid computing layer is depicted in Fig. 4.

3) Smart-Contracts: In our proposed system, we used smart-contacts for writing business logic for decision making at hybrid computing layer. The smart-contracts are classified into two categories, one for creating service agreement between different actors and the other for service automation (business logic). The service agreements are created and published by the concerned service providers (Doctors, insurance agents, etc.) and service consumers (patients). The dynamic smart contract process is illustrated in Fig. 5. The two parties write their terms (T), conditions (C), and agreements (A) into the smart contract creator module to convert the given inputs into the smart-contract. Further smart-contract is signed by two parties and published using service provider wallet account. In addition, the hybrid computing links the smart-contract address with the patient service identity (SI_{pat}) and public identity (ID_{pat}) for service automation.

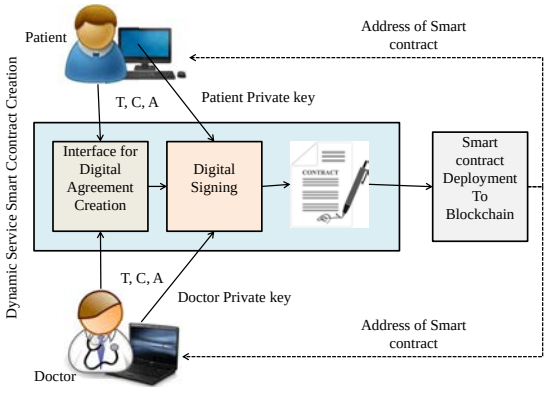


Fig. 5: The Dynamic Smart Contract Creation Process.

TABLE III: Notions and detailed description

$D(key, encrypted-data)$: decryption function	DT_{raw} : raw sensor data	DT_{reg} : registered data
DI_{sig} : device digital signature	$E(key, plain-data)$: encryption function	EI_{div} : device embedded identity
FN_{req} : requested file name	FL_{acc} : file access code	FL_{req} : requested file
$f()$: private identity generation function	SPOF: single point of failure	FL_{add} : file address on DDSS
GI_{id} : group identity	ID_{hos} : hospital identity	$H()$: hash function
ID_{dev} : Device developer given identity	ID_{div} : device identity	ID_{rin} : ring identity
ID_{pat} : patient identity	ID_{doc} : doctor identity	ID_{req} : requester public identity
ID_{HC} : hybrid computer identity	$\oplus$: XOR operation	PA_{inf} : patient basic information
PU_{div} device public key	PU_{usr} : user public identity	PR_{usr} : user private identity
RC_{ID} : record identity	PU_{edg} : edge computer public key	PR_{edg} : private key of edge computer
RI_{div} : device registration application identity	RN_{HC} : hybrid computing random number	SI_{div} : device service identity
SI_{pat} : patient service identity	TM_{stm} : file creation time stamp	X_{HC} :random number from Elliptical curve
RC_{tot} : total successful rings created	RG_{tot} : total requests generated	RR_{hos} : total rings representing
TR_{hos} : total rings in the hospital	TN_{suc} : total successful file access	TN_{tot} : total access attempts
EV_{doc} : doctor experience factor value	IV_{doc} : doctor index value	PR_{div} : device private key

VI. Proposed Security and Privacy Methods of the Fortified-Chain

This section contains proposed architecture-specific cryptographic methods to fulfil the system requirements. Table III represents the symbols and their descriptions used in this paper. The device authentication mechanism helps the system to verify the device identity during system boot-up to prevent unauthorised device participations. The SRAC mechanism performs two operations i.e actor's authenticity validation and resource access control mechanism. In our proposed work we have suggested the patient anonymity mechanism which helps the system to hide the relationship between patients and their data on DDSS network. The upcoming subsection explains the steps initiated by the hybrid computing to make the system ready for real time healthcare operations.

A. IoMT Device Enrollment/Registration

1) Initialization: The hybrid computing chooses an Elliptic Curve Cryptography (ECC) approach for its cryptographic methods and selects an elliptical curve $y^2 = x^3 + 2*x + 3$ over finite field of size 263. At first, hybrid computing generates a secret number (X_{HC}) using true random number generator method that produces a private key with size 256-bits. The hybrid computing public key is generated using the point doubling method with a fixed generator point [44]. In addition, a permutation operation generates different 256-bit key string for each hospital from hybrid computing 256-bit public identity. Subsequently, from hospital level 256-bit string, the edge computer generates local identities like patient identity (ID_{pat}), group identity (GI_{dev}, GI_{id}), doctor identity (ID_{doc}), service (SI_{div}), device identity (ID_{div}) and ring identity (ID_{rin}). The global identity of each patient is composed of using HXPY format, where HX represents the hospital identity (ID_{hos}) and PY represents the patient public identity (PU_{usr}). Moreover, the organization level edge computer creates its local DDSS sub-networks by using a sub-swarm key generation process. Simultaneously, every organization publishes its device and actor's public keys on blockchain based main DDSS in an encrypted form. This public key pool helps the devices and patients to establish a secure data exchange with each other.

2) Device Registration: In our proposed work, device registration and authenticity validation is done by hybrid computing layer. Hospital level edge computer initiates the device registration process and generates different identities for various devices using their unique embedded identity (ID_{dev}). For every applicant device, hybrid computing issues a device registration application identity (RI_{div}) by considering hash value of embedded identity H(ID_{dev}) as input. A hospital level 32-bit random number (RN_{HC}) is generated by the edge computer for device identity generation. Hybrid computing computes a hospital-level unique identity for each device using the following expressions:

$$RI_{div} = H(RN_{HC} \oplus H(ID_{dev}) \oplus ID_{HC}) \quad (1)$$

$$ID_{div} = H(ID_{dev}) \oplus H(RI_{div}) \quad (2)$$

In the first equation, an XOR operation is performed between device embedded identity hash value, chosen random number and hybrid computing identity to generate device registration application identity (RI_{div}). The second equation generates device identity (ID_{div}) by performing an XOR operation between hash value of application identity and hash value of device embedded identity. Subsequently, every device gets a system level unique identity by prefixing the hospital identity before the device identity. Moreover, it also assigns a service identity (SI_{div}) and group identity (GI_{id}) to the device for patient data aggregation and system automation. After successful device registration the edge computer creates a register record in device registration database (DT_{reg}).

Then, every device generates its public key and private key pair PU_{div}, PR_{div} using the predefine ECC parameters, also generates its own digital signature (DI_{sig}) using Alg.1.

The digital signature (DI_{sig}) is generated by using devices registration application identity (RI_{div}). In the beginning, the device generates hash value of RI_{div} and it is encrypted with its private key. Then, it appends the RI_{div} to encrypted data to generate its DI_{sig} as shown in line (1-2). The signature validation process decrypts the $E(PR_{div}, Temp)$ part using device public key (PU_{div}). Simultaneously, it calculates the hash value of RI_{div} and it is compared with the received hash value. If the both hash values are equal then the digital signature is considered as valid otherwise it is considered as invalid. The digital signature validation process is illustrated in line no (3-10).

At the same time, the devices under same hospital can mutually authenticate using mutual authentication token (MAT_{div}). The token is generated by hybrid computing layer using equation $MAT_{div} = ID_{div} \oplus H(P(ID_{hos}))$. The symbol $P()$ represents the permutation function which changes the order of bits in a given binary data. The token is assigned

Algorithm 1: Digital Signature Validation

Input: Device registration application identity (RI_{div}), Device private key (PR_{div})
Result: Device digital signature (DI_{sig}) generation and validation

// DI_{sig} generation
$Temp$= H(RI_{div})
DI_{sig} = $E(PR_{div}, Temp)$ + RI_{div}.
// DI_{sig} validation
Divide $E(PR_{div}, Temp)$ and RI_{div} from device's signature DI_{sig}
$Temp$ = $D(PU_{div}, Temp)$
$Temp_1$ = $H(RI_{div})$
if $Temp_1$= $Temp$ **then**
    DI_{sig} valid for RI_{div}
else
    DI_{sig} not valid for RI_{div}
end

to the device along with other identities after a successful registration process. This token is useful to establish device-to-device secure communication without disturbing the edge computer operations.

B. Device Authentication Process

The algorithm-2 describes the proposed device authentication process (DAP) in an abstract form. The DAP validates device authorization before device initiating the system services. A grey list is maintained by the hybrid computing to stop unauthorized devices to participate in the system services. The DAP considers multiple arguments like device identity (ID_{div}), ring identity (RI_{div}), service identity (SI_{div}), group identity (GI_{id}) and digital signature (DI_{Sig}) to simplify authentication process, service automation, and data aggregation. Our proposed DAP only allows authorised devices to perform data read, write, and download operations from blockchain based DDSS. A hospital level edge computing performs DAP for the internal devices and the cloud computing performs DAP for a remote patient devices. For each registered devices the hybrid computing maintains a status value in device registration database (DT_{reg}). The possible status value of a device is 0 for new device, 1 for authorised, and -1 for unauthorised respectively. The DAP reads the device status from the local registration database cache at respective edge computer. For new devices, it initiates device authentication by considering RI_{div}, GI_{id}, DI_{Sig} as inputs. The DAP validates device authenticity (line-4) by validating its digital signature using algorithm-1. In case, both the device status and digital signature are valid, then it sets new status value to 1 as shown in lines (5-6) and updates the database. The device service access rights are validated by using ring access control records with the help of function called CheckDeviceRights(ID_{div}, ID_{rin}) as shown in line-8. The devices access rights are assigned by the caretaker or patient in the initial stage of service automation using dynamic smart contract creation module, and grouped as one RightsSet{R,W,D} is represented in Fig. 6.

Algorithm 2: Device Authentication Algorithm

Input: ID_{div}, SI_{div}, GI_{id}, DI_{Sig}, ID_{rin}
Result: Device allowed to initiate file access operations

1 $status$ = Getstatus(ID_{div}, SI_{div}, GI_{id});
2 **if** *status==0* **then**
3 Validate ID_{div} and DI_{Sig};
4 **if** $ID_{div} \in DT_{reg}$ *and* DI_{Sig} *valid* **then**
5 SET device $status$ to 1 ;
6 UPDATE DT_{reg};
7 UPDATE Local cache;
8 $RightsSet$ = CheckDeviceRights(ID_{div}, ID_{rin});
9 **if** $RightsSet \neq \emptyset$ **then**
10 ALLOW access based on $RightsSet$ (R,W,D) values;
11 **end**
12 **else**
13 SetState(DT_{reg}, ID_{div}, $status$);
14 SET device $status$ to -1;
15 UPDATE DT_{reg};
16 UPDATE Local cache;
17 REQUEST the caretaker to validate Device authenticity;
18 **end**
19 **else if** *status==1* **then**
20 $RightsSet$=CheckDeviceRights(ID_{div}, ID_{rin});
21 **if** $RightsSet \neq \emptyset$ **then**
22 ALLOW access based on rights assigned (R,W,D);
23 **end**
24 **else**
25 Block ID_{div};
26 ALERT the caretaker about unauthorised file aceess attempt;
27 **end**

For the new devices the state is generated and assigned with the values using the function SetState(). For existed and known devices it uses local cache to speed-up the validation process. Simultaneously, a grey listed devices are identified and set the status value to -1 as shown in line-14. Every organization calculates a different useful scores related to their actors and devices, for example number of unsuccessful attempts in a given time, index value, etc. These calculated values are used to determine device access rights. The grey listed devices are only allowed for services after a manual validation process by the caretaker. This mechanism prevents fake devices participation in healthcare system activities. The devices under same hospital can exchange secure information using session secret key. The process starts with device-A, which sends a random number (RN_A) using a secure message $E(PU_B, [MAT_A||RN_A||ID_A||DI_{sig}])$ to device-B. The receiver side, device decrypts the

Ring ID (ID_{rin})	File Id (Fl_{req})	Patient ID (IDpat)	Doctor ID (ID_{doc})	Service ID (SI_{div})	Group ID (GI_{id})	Device ID (ID_{div})	Index Value	Read (2)	Write (1)	Download (4)
Ring 1	_File_ 20-12-20:12:54pm	H1P203	H1D653	H1S41	H1G789	H1DE12-17	335	6*H1P203,3*H1D653,5*H1G789,5*H1DE12-17		
Ring 2	_File_ 12-02-20:02:12am	H1P314	H1D123	H1S59	H1G124	H1DE88-94	278	5*H1P314,7*H1D123,5*H1G124,1*H1DE88-94		
Ring 3	_File_ 11-09-20:02:04pm	H3P245	H3D341	H3S12	H3G876	H3DE1-8	300	5*H3P245,5*H3D341,2*H3G876,1*H3DE1-8		
Ring N	_File_ 15-04-20:07:55pm	HNP232	HND234	HNS11	HNG123	HNDE123-134	302	2*HNP232,XXXX,XXXX,XXXX		
Hospital level ID space size - 256 bits, ID_{pat} – 32 bit, ID_{doc} - 32 bit, GI_{id} - 32 bit, SI_{div} - 32 bit, ID_{div} - 64 bit, ID_{rin} - 64 bit										

Fig. 6: The overview access control ring table structure.

$D(PR_B, [MAT_A||RN_A||ID_A]||DI_{sig})$ and computes the value of $(ID_A \oplus MAT_A)$. Simultaneously, it computes $(ID_B \oplus MAT_B)$ and compares the both outcome values. If both the values are equal then device-B accepts device-A. Subsequently, chooses a random number (RN_B) and computes $(RN_A)^{RN_B}$ as a mutual key. Further, a secure message $E(PU_A, mutualkey||ID_B||DI_{sig}||timestamp)$ is sent to the device-A to share the newly generated session key. Finally, both the devices establish a secure communication using a session key.

C. Selective Ring based Access Control (SRAC) Algorithm

The proposed SRAC algorithm depends on ring rules to control data access rights for every scenario as shown in Fig.6. Each record in the table is called as ring for a particular patient. The patient is allowed to create any number of static unique rings for different files using simple user-interface. At the same time, with the help of index value, a patient can create a dynamic ring to accept valid file accessing requests from the remote location. The patient sets a different index value for different category of files based on critical information. The SRAC compares index value of requester with required index value in the ring to provide secure read-only access to remote actors. Every hospital computes the local actors index values for dynamic file access control. The mechanism prevents the third-party actors from accessing files. The static rings are used for hospital level access control service whereas the dynamic rings are useful in handling remote location requests. We incorporated three file operations in SRAC mechanism known as: Read (R), Write (W), and Download (D) respectively. In ring 1 the second cell indicates the patient medical file name (_File_20-12-20:12:54pm) as shown in the table-6. However, the "_File_" is replaced by patient private identity before it is stored on local DDSS network. The third field is encrypted and protected by the edge computer to hide the patient public identity, and the last field indicates access rights for different devices and actors for that particular file. For example, the doctor (D653) in hospital (H1) have only read-write permissions for the _File_20-12-20:12:54pm.

Algorithm 3: Selective Ring based Access Control (SRAC) Algorithm

Input: ID_{div}, SI_{div}, GI_{id}, Requested File Name (FN_{req}), ID_{pat}, Identity (ID_{rin}), Requester Identity (ID_{req}), DI_{Sig}

Result: Requested File (FL_{req}), File address (FL_{add}), file access code (FL_{acc})

```
SRAC(ID_div, SI_div, GI_id, FN_req, ID_pat, ID_rin, ID_req, DI_Sig)
state = GetState(ID_div, SI_div, GI_id);
if state ==1 then
    if ID_div, ID_req ∈ local ID_rin database then
        READ AccessRights(ID_rin);
        READ IndexValue(ID_rin);
        if Access Right present in ID_rin for ID_div or ID_req then
            FL_add = GetResourceAddress(FN_req);
            FL_acc = GetResourceCode(FL_add);
            Buffer = GetFile(FL_add,FL_acc);
            ALLOW appropiate file actions for defind (R,W,D) values;
            ALERT ID_pat about File access;
        else if IV_doc ⩾ IndexValue then
            Do steps 8, 9, 10;
            ALLOW SecureRead(Buffer);
            ALERT ID_pat about File access;
            // provide review option for new activity
        else
            BLOCK request;
            SET status of ID_div to -1 in local database;
            ALERT ID_pat about unsuccesful file access;
        end
    else if ID_div, ID_req ∉ ID_rin then
        REQUEST ID_pat for File access rights ;
    else
        BLOCK request;
        SET status of ID_div to -1 in local database;
    end
end
```

The access rights database (rings) is distributed across the blockchain based DDSS networks in a secure mode. When an actor requested for a patient file, the edge computer checks the ring information in local cache for suitable ring record. However, if no such record found in local cache then it will fetch records from global data which is located at cloud computer. The algorithm-3 describes the SRAC mechanism in an abstract form. Every organization calculates internal actors index values by using equation-3 where (TN_{suc}/TN_{tot}) represents the successful file access ratio for the doctor, the (RR_{hos}/TR_{hos}) indicates the doctor's impact on hospital services, the (RC_{tot}/RG_{tot}) indicates ratio of new ring generation for the doctor, and the EV_{doc} indicates doctors experience

factor. Patients can create dynamic rings for remote doctors by using $GenerateRing()$ with appropriate index values. Further, the rings are updated locally as well as globally. Such index values are used as credibility score when a remote doctor requested patient data from remote location. Only actors with higher or equivalent index value are allowed to read the patient files.

$$IV_{doc} = ((TN_{suc}/TN_{tot}) + (RR_{hos}/TR_{hos}) + (RC_{tot}/RG_{tot}) + (EV_{doc})) * 100 \quad (3)$$

A device authentication status is considered for a device access rights validation. If the local cache contains records of a device, it reads access rights set R,W,D along with its index value. However, in few cases the dynamic rings may not contains the set values and only contains the index value. In such cases, SRAC compares the requester index value against the patient ring value for validating access rights. If the IV_{doc} greater than or equal to the patient's benchmark index value in ring, then SRAC permits the requester to perform read-only operation on the file. Due to this, SRAC initiates GetResourceAddress(), GetResourceCode( ) and GetFile() between the lines (8-10) to get secure copy of the requested file with access code from the blockchain-based DDSS. Finally, it decrypts the file with the access code and allows the requester to perform appropriate operation. If a requester with low index value tries to access the file, the SRAC blocks the request and adds the devices to the grey list. However, when no matching record is available for the request than SRAC sends a request to the patient to take decision. The patient can create a new ring for the requester or append the existing access rights for the requester.

D. Patient Anonymity

The algorithm-4 describes how proposed work establishes the anonymity in medical file management process. Patient registration is done by healthcare service providers like hospitals or by the patient using service providers application. The registration process validates device authenticity before it allowed to create a user profile (line-1). The registration process generates patient public identity (PU_{user}) for hospital level identification and private identity (PR_{user}) strings for patient medical data storage on DDSS. The function GenerateUserID() at line-2 takes $H()$ value of users unique identification data like fingerprint, iris, rim of ear, retina, toe print etc., for generating patient public identity. A unique public patient identity string generation explained as follows: The input of 256-bit hash string is divided into two equal parts named as left part (LP) and right part (RP). Then, undergoes an XOR operation at line-5. The result of XOR operation is a temporary 128-bits. The above steps are repeated until the resulted output is a 32-bit string. The string is handed over to the patient as a public identity. Eventually, it is passed to the function named $f()$ for private identity generation, as shown in line-10. Thereafter, the function $f()$ divides the input binary string into 4 equal size parts and generates a temporary 16-bit string (YZ) by considering the last two right-most bits from each part. In addition, it performs the XOR operation followed by bit shuffle operation on generated string as shown between the lines (13-15). The edge computer organises the patient medical files in a DDSS local directory which is created by the function $f()$. The directory and the medical data files are named by using the patient private identity string $H(D'C'B'A')$ concatenated by a timestamp TM_{sta}.

Algorithm 4: Patient Anonymity Algorithm

Input: H(PA_{inf})
Result: PR_{usr}

1 **while** *DeviceAuthentication(* RI_{div}, GI_{id}, DI_{Sig}*)* == *True* **do**
  `// patient public identity string generation function initiated by actor`
2 GenerateUserID(H(PA_{inf}))
3 **repeat**
4 divide the H(PA_{inf}) string into two equal parts Left Part (LP) and Right Part (RP)
5 X = LP $\oplus$ RP
6 PA_{inf} = X
7 **until** *till X becomes a 32-bit binary string*;
8 **return** *X as* PU_{usr} to patient and edge computer
  `// patient private identity string generation function initiated by hospital edge computer`
9 **while** PU_{usr} *presents in database* **do**
10 $f(PU_{usr})$
11 Generate a 16-bit string (YZ) from PU_{usr}
12 divide the PU_{usr} into two equal parts AB and CD
13 $A'B'$ = AB $\oplus$ YZ
14 $C'D'$ = CD $\oplus$ YZ
15 $A'B'C'D'$ shuffled to $D'C'B'A'$
16 **return** $H(D'C'B'A')$ as PR_{usr} to edge computer
17 **end**
  `// MapID() function`
18 MapID(PU_{usr})
19 **if** $PU_{usr} \neq$ *NULL* **then**
20 $f(PU_{usr})$
21 **end**
22 **end**

The function MapID(PU_{user}) at line-18 is a public function that any authorised actor can initiate. However, $f()$ is a private function which is only accessible to the hybrid computing layer devices. Therefore, the patient medical records are stored in folder with the name equivalent to $H(D'C'B'A')$ are safe from backtracking. Moreover, files are encrypted using edge computer public key before saving in blockchain based local DDSS folder $H(D'C'B'A')$. This mechanism provides full level confidentiality and anonymity to the patient medical records and also eliminates the relationship between patient's public identity with their medical data directory.

VII. Logical Analysis and Threat Modelling of Fortified-Chain

In this section, we have presented a logical analysis of the proposed Fortified-Chain model from a security and privacy

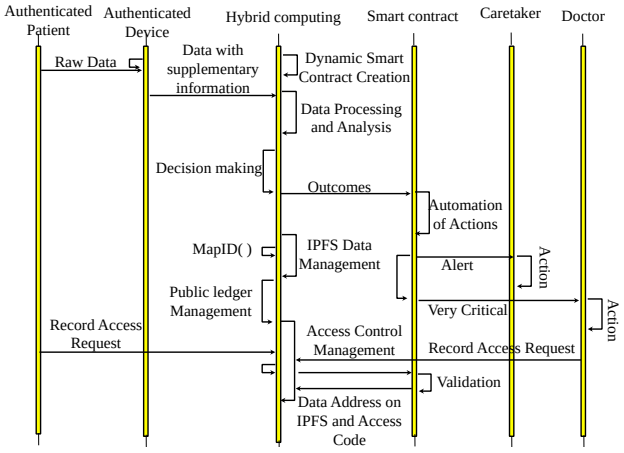


Fig. 7: Sequence Operational Flow

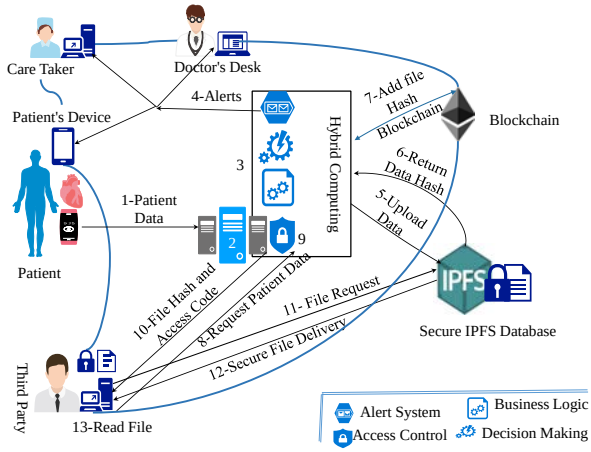


Fig. 8: Top Level Operational Flow of Healthcare System with Fortified-Chain Deployment.

perspective. Fig.8 represents a simple operational flow of the proposed architectural model, which represents the communications establishment between different elements, and Fig.7 demonstrates the operational flow in sequence diagram. The following subsection provides a logical analysis of Fortified-Chain for different security issues.

A. Logical Analysis of Fortified-Chain

a) Security: Every architecture must consider security requirements such as confidentiality, integrity and availability. Correspondingly, our proposed architecture needs to fulfil the above requirements. The confidentiality guarantees the data secrecy between authorised parties only. In our model, the patient data is only shared to the allowed authorised actors using algorithm-2 and algorithm-3. These two algorithms combination protect the data confidentiality by blocking unauthorised devices and actors access attempts. Further, all critical hospital level patient data is isolated from main DDSS network using separate local DDSS networks. All access records are fully encrypted by hybrid computing layer before storing in decentralized storage system. Additionally, the DTLS protocol provides communication layer level data confidentiality. The feature of integrity guarantees that no unauthorised data modification is done during data transmission. Our proposed model prevents unauthorised data modification at storage level and transmission level using blockchain and hash functions. In addition to that, the digital signature of sender guarantees data integrity at receiver side.

b) Privacy: The SRAC mechanism guarantees the medical record's privacy by selective sharing of file access codes to authorised devices and actors. In addition to that, the sensitive or critical patient data is securely isolated from other organizations using local DDSS networks. The access rings dose not reveal the patient real identity instead it uses a system generated secure digital identities. The $MapID()$ module eliminates direct relationship between the patient identity and its data on blockchain-based DDSS, which provides another layer of privacy for patient data at storage level.

c) Immutability: In traditional systems data is vulnerable to accidental and malicious attacks. In this proposed model all critical data is protected against such issues by incorporating blockchain-based public ledger mechanism. Once the data is added to the blockchain based DDSS in secure form, there is no way to alter the data without modifying the ledger which is practically impossible.

d) Availability: The blockchain based DDSS file management system at hybrid computing layer updates the local cache and global cache in frequent time interval to provide data availability. Our system allows authorized people to access data at any given time by using the DDSS client interface system. The two level secure backup can be used when one or more edge devices went down. However, in traditional systems, the data stored in a central location may affect data availability.

e) Traceability: In our Fortified-Chain, records every event on blockchain based DDSS by uploading transaction logs along with the data in an encrypted format. Such action provides system level event traceability for different actors. Moreover, the DDSS file system maintains file versioning mechanism to trace file modification activities. Moreover, in the traditional server centric H-CPS systems, the file altering is untraceable.

f) Anonymity: The $MapID()$ modules at edge computer provides the data anonymity by eliminating the relationship between the patient and their data using public and private identities. Moreover, the patient real information is not used in any of the system service operations instead a hash value of unique identification feature is used as a public patient identity. The records maintain only hash values of patients identities instead of real identities.

g) User-control: In our proposed algorithm SRAC, rings rules and dynamic priority indexed values provide more control over the patients own data. This mechanism allows the patient to share their data based on their own choice, and also have rights to block the data access at any time.

h) Scalability: The combination of blockchain-based DDSS, decentralized SRAC access control mechanism and distributed hybrid computing framework allows the system to expand its infrastructure as per real time requirements. Any authorised organization can become a member of this

framework with few modification in its existing infrastructure. Table IV illustrates the comparison between our system and related works.

B. Threat Modelling and Analysis Process of Fortified-Chain

We have considered a simple system communication flow for threat modelling which is represented in Fig. 8. The threat modelling contains the definition of security, threat selection, threat mitigation and threat assessment steps respectively. Every security system needs to fulfil the properties like confidentiality, integrity, availability to consider it as a secure system. In order to identify the suitable threats related to the proposed architecture, we have chosen Microsoft thread modelling tool which generates a list of possible threats using Spoofing, Tampering, Repudiation, Information disclosure, Denial of Service, Elevation of Privilege (STRIDE) methodology. In the threat selection process, the tool suggests the following threats to mitigate: spoofing of cache and data, tampering of data, actuators-configuration tampering, non-repudiation, information-disclosure at storage level and transmission level, distributed denial of service (DDoS), and finally escalating actress privileges.

The mitigation process explains how the proposed system mechanisms mitigate the above-said threats. In order to prevent cache or patient data spoofing attacks the proposed architecture limits the trust relationships between the cloud and hospital-level edge computers. Both parties believe in blockchain public ledgers for updating their local caches and data sharing. Sensors and actuators are not modifiable because they operate on smart-contracts and public ledgers. The DTLS and digital signatures guarantee the data integrity and authenticity at the communication level. Whereas, blockchain and public ledgers assures the data integrity at the storage level. In the proposed system only authorized actors can access the data, however the malicious activities are easily identifiable with the help of public ledgers and system-level event logs. The information disclosure threats are mainly related to the system local cache and patient data disclosure. In our system, the local cache is maintained by trusted edge devices and allows other trusted edge devices in the same hospital to read the cache. The critical information of patient is isolated using local DDSS networks and access control mechanisms. Further, one patient's data is isolated from the other patients using MapID() function. Moreover, the data is encrypted before it is published to the blockchain-based DDSS network.

The DoS attack is well known possible threat for any critical system. The proposed architecture distributes the data and the process modules among different peers in the hybrid computing layer that guarantees the data and service availability even some nodes went down. It is impractical to shutdown majority of the nodes inside hospital's private network because it is isolated and controlled by the hybrid computing access rights. Moreover, the architecture is designed based on a REST full architecture model which does not maintains any state information during communication. This mechanism prevents the SYN flood attack. Even if the cloud services went down the hospital level hybrid computing works without any effect on hospital internal services. The privilege of any device or actor is hold by two different databases, one is device registration database and other one is access control rings. As the rings and public ledgers are published to blockchain based DDSS networks they are tamper resistance. Ones the rings are added to the database then it becomes read-only for authorised system modules. The ring information indicates the number of allocated devices for each patient. Algorithm-2 and algorithm-3 provides protection against information disclosure.

A formal verification of proposed secure protocol is validated using Scyther tool which is a secure protocol verification tool. The formal verification results are showcased in the Fig. 9. Form the above result it is clear that our communication protocol guaranteed for all the claims included in Scyther related to the device-to-device and device-to-hybrid computing data confidentiality and finds no attacks. The climes Nisynch and Niagree are used to detect man-in-the-middle attacks and replay attacks. The Session-secret key are the parameters that claim confidentiality. The related documents available at [45].

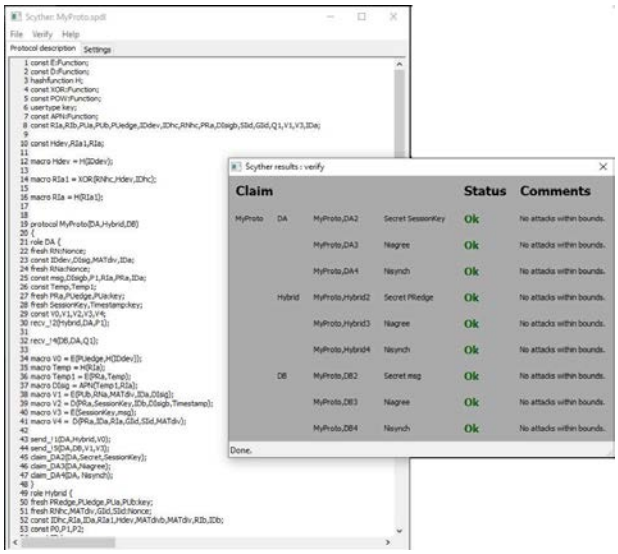


Fig. 9: Formal Verification of Fortified-Chain.

VIII. Experimental Analysis of Fortified-Chain

We have investigated the proposed architecture's prototype file sharing performance using platforms like Ethereum, Infura, Metamask, and Remix application stack. The IPFS has chosen as a distributed data storage system for the proposed architectures prototype performance evaluation. We have also suggested a system analysis from the security and privacy perspective by highlighting the benefits of the proposed architecture over the classical healthcare systems.

A. A Specific System Setting for Experimental Validation

We have deployed our prototype file storage and sharing on Infura infrastructure, which provides connectivity to ethereum and IPFS test node networks via HTTPS and WebSocket application interface. The smart-contracts compiled and deployed to the test network using Remix IDE which is a

TABLE IV: Our Fortified-Chain is Automated while Performing Faster

Research Work	Access Control	SPoF	Low Latency	Support for Automation
Yingate et al. [14]	Cloud-centric Attribute-based Encryption	Yes	No	No
Xia et al. [15]	Cloud-centric Blockchain-based smart-contracts	Yes	No	No
Nguyen et al. [17]	Cloud-centric Blockchain-based smart-contracts	No	No	No
Wang et al. [18]	Cloud-centric Attribute-based Encryption	Yes	No	No
Zhang et al. [19]	Cloud-centric Multiple Access Control Contracts (ACCs)	Yes	No	No
Zheng et al. [16]	Cloud-centric Attribute-based Cryptosystem	Yes	No	No
Xu et al. [20]	Cloud-centric Federated Capability-based Access Control	Yes	No	No
Alkhushayni et al. [30]	Cloud Blockchain-centric Data sharing techniques	Yes	No	No
Pace et al. [26]	Decentralized Edge Computing-centric General model	No	Yes	No
Gonalves et al. [29]	Decentralized Semi Blockchain-centric data sharing	No	No	No
Fortified-Chain	Decentralized Selective Ring based Access Control	No	Yes	Yes

web based integrated development environment (IDE). All our smart-contracts are written in the solidity language [46]. However, ethereum wallet accounts are created and accessed using Troffile, Ganache, and Metamask plug-in. The metamask is a browser extension which allows us to interact with the ethereum blockchain. The experiments are performed on rinkeby test network to evaluate the prototype performance in-terms of response time and storage cost. However, a local IPFS client domain is initiated from a MacBook Air which runs on macOS 10.12 operating system having 1.8 GHz dual-core and Intel Core i5 processor. Two PC are deployed, with configuration of intel i5, 4Gb RAM and windows 10 operating system as edge computers to perform hybrid computing layer tasks. These two systems are connected through test network. We have designed a web application using Nodejs, npm library and web3.js library [47] to interact with the test network. Moreover, an android mobile with 4Gb RAM is used for uploading sensor dataset using the web application. The same android device is used for device authentication and validation process. The transaction details of file uploading different dataset on the test network is shown in Fig.10.

B. Testing Methodology

Based on the above defined specification terms of hardware and software configuration, we have measured the latency in terms of file uploading time and downloading time by comparing our prototype with google cloud service. In our proposed prototype, we have considered only an authorized user profile for prototype checking with a chosen dataset files.

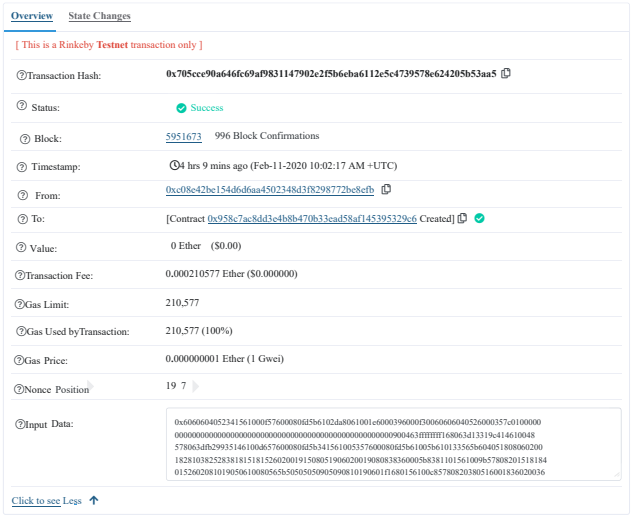


Fig. 10: Transaction Details of Fortified-Chain on Rekbay.

The first operation is uploading files to the IPFS is followed by storing file hash on the blockchain. Secondly, we generate download file request by fetching file hash and access codes from the blockchain based access control ring database. The test networks is configured with auto mining mode for automatic block validation. The Fortified-Chain uses local edge cache for frequent service requests where the response time decreases and eliminates unnecessary blockchain operations.

C. Datasets for Experiments

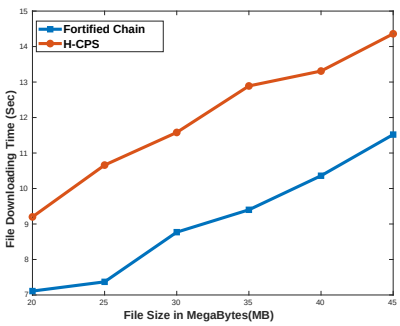
We have chosen few heart disease related dataset [48, 49] for validating system functionalities and capabilities.

Our prototype storage system using smart contract address "0xEcc20d265D5C12Fe9C3B42D0735F7361c650520B" to deal with IPFS database. All the transactions are initiated using Metamask account "0xc08e42be154d6d6aa4502348d3f8298772be8efb". We have used developer tools and communication protocol network information of client-side browser application. From the above results it is clear that data storage and retrieval operations of datasets produce less latency. At the same time, Table V represents datasets storage details whereas Table. VI illustrates the corresponding Fortified-Chain DDSS storage cost and response time intervals.

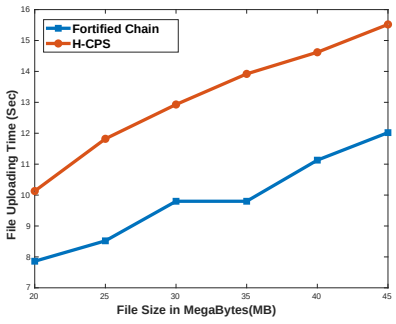
D. Experimental Results

In this section, we have compared our proposed prototype which is based on Selective Ring based Access Control (SRAC) file system over classic cloud centric blockchain-base Healthcare-Cyber Physical System (H-CPS). It is important to note that the google cloud drive application is used as a benchmark for file accessing time related results, so other work in future can compare the performance to the proposed solution. Several experiments have been done to evaluate different performance measurements.

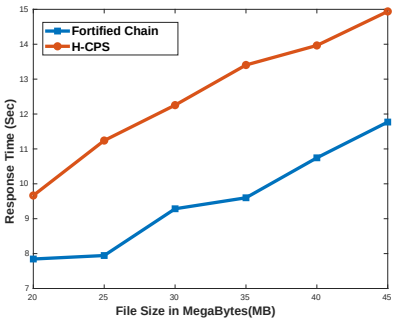
To evaluate the network file operations latency, we measured the average time required for the blockchain-based IPFS system to upload and download each file operation with different



(a) File Downloading on Fortified-Chain Prototype



(b) File Uploading on Fortified-Chain Prototype



(c) File Operation Response Time of Fortified-Chain

Fig. 11: A Selected Experimental Results of Fortified-Chain

TABLE V: Healthcare Data Securely Stored in the Proposed Fortified-Chain

Dataset Name	File Size	IPFS Hash	Transaction Hash
Processed. hungarian. data.csv	10 KB	QmYnuoFJvis9n5 S7FZiv87F9eZEC rKTDwN2tnARYU PJKPC	"0x174724c7946 43f999869f45cb 7e7e36ab02a3d 1ee82f4779cb06 773de0ab7158"
Processed. cleveland. data.csv	18 KB	QmTSQNWmueD Kr9YG7aRDMgKk YcJtVfBdPYXptPB Wk94fvg	"0x37445f2f264 66f88f1757111b 086fca7373635 6e1dd11ce1f66 94d33ca53bb73"
Switzerland. data.csv	25 KB	QmY4fQDzab2p KMoTw9DfqabsQ WriwqdUhpMyKg ZjLhvtY8	"0x9d9420ff53f2 4c2d183f04f074 3b9d89ebeaa80 813061155c052 6e26c8842be1"
Long-beach -va.data.csv	40 KB	QmTQ9okXFkh NWTUGjwupWs Em9qJsuWoA9X pGNYEzWXTx13	"0x5cdcd6304b7 6df3e08656ed76 b80dcec019ef23 5fa472ad962d6c aa28dd12c7b"
Cleveland- data.csv	64 KB	QmP4ZFX8UzqS cHKrj26LqUTM8S De4ELygsjnMiVsL aTQ8L	"0x479bca0114b 32beae89a6f53c c08985b85597d7 aa0083dcd29733 ea30c9e65e3"
New-data .csv	189 KB	QmS3BxnU7XfU B6Bbz2wmfuwYb JNzVgtEgXzSgVrL Ff38g3	"0x1488e0e2810 fa787a8075ccda efe5e1dfe7712ee 10b08c0e411668 9380f6def9"
New.data .csv	390 KB	QmXcW49P3hG 7ZuuiH5uC3Qt3Q qM5GJutaRHnRU CfV8ePSn	"0x3afeafea8473 4cd88748037759 383673a2c925e1 eff2a7b986edea8 2dc984d20"

TABLE VI: Healthcare Datasets File Storage Operation Cost and Time interval

Transaction Fee	Upload Time In (ms)	Download Time In (ms)	Response Time In (ms)
0.000010537 Ether	86	432	259
0.000010537 Ether	86	432	259
0.000011233 Ether	103	660	381.8
0.000011348 Ether	131	709	420
0.000011826 Ether	166	939	552.5
0.000012114 Ether	184	1060	622
0.000019309 Ether	395	1820	1107.5
0.000020016 Ether	528	2210	1369

file sizes. The experiment results show that our proposed model can deliver faster file operations compared to the classic cloud centric blockchain-based H-CPS for multiple concurrent requests. To support this, we performed the same file operations on the google cloud and the results are compared with our proposed prototype model. With this analysis, it is clear that our model has produced low latency in respective file operations when compared to the classic cloud centric blockchain-based H-CPS as shown in Fig. 11.

Our proposed prototype maintains a local cache that reduces the ethereum transaction and storage cost also minimizes latency. In our Fortified-Chain, the file storage which is based on the blockchain-based IPFS is represented in Table VII. From the experiment analysis, we can conclude that storage in classic cloud centric H-CPS based blockchain has more ethereum cost compared to the Fortified-Chain. The results demonstrated that our Fortified-Chain is feasible, and provides low latency and less ethereum cost compared to the classical cloud centric blockchain-based H-CPS. We have performed 10 concurrent connection load test for different file sizes ranging from 100 KB to 800 KB and computed a mean value for each file size. The resulted values are compared with Pace et al. [26] to showcase Fortified-Chain behaviour as shown in Fig. 12. We observed minor delays in our model because of robust features of blockchain based DDSS and security mechanisms.

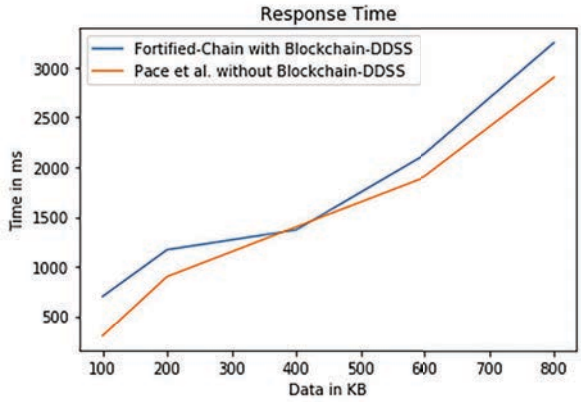


Fig. 12: Pace et al. [26] Vs Proposed Fortified-Chain

TABLE VII: Comparison of Storage Cost

Cost (Ether)	20MB	25MB	30MB	35MB	40MB	45MB
Traditional blockchain H-CPS Database	15.22	18.31	20.152	21.86	23.318	25.672
Fortified-Chain (Current Paper)	0.017	0.031	0.053	0.076	0.089	0.13

IX. Conclusions and Future Work

In this work, we have elucidated a novel approach to solve the problems related to latency, data security, privacy, anonymity, and traceability in decentralized IoMT based smart healthcare systems. Moreover, it showcases the leverage blockchain, DDSS, and hybrid computing to deliver architecture level solutions to the discussed issues. The system level traceability is achieved through blockchain-based tamper-proof public ledgers. The SRAC and other proposed cryptography techniques assure the medical data security and privacy. On the other hand, smart contract automates the medical emergency alerting and primary medical services. Simultaneously, the proposed architecture provides a platform for different stakeholders in the healthcare industry to make digital agreements. In the logical analysis, our system exhibited expected functionalities like low latency in data sharing for critical situations.

In the future work, we will explore the techniques to leverage the intelligence to our system by using AI/ML technology. Our focus will be on future generation critical patient monitoring and assisting system framework requirements to deal with different types of pandemics. Aim of our future work is to provide a robust system to enhance healthcare services capability along with quality of services (QoS). Moreover, we will develop a full level prototype with all the proposed capabilities in real time scenario. In addition, the future work will be able to detect and alert all stake holders about pre-pandemic identifications related to a particular area in real time.

Acknowledgments

The Authors would like to thank the Science and Engineering Research Board (SERB) for supporting this work, Grant number TAR/2019/000286.

References

[1] Y. Sun, F. P. Lo, and B. Lo, "Security and Privacy for the Internet of Medical Things Enabled Healthcare Systems: A Survey," *IEEE Access*, vol. 7, pp. 183 339–183 355, 2019.

[2] V. P. Yanambaka, S. P. Mohanty, E. Kougianos, and D. Puthal, "PMsec: Physical Unclonable Function-Based Robust and Lightweight Authentication in the Internet of Medical Things," *IEEE Transactions on Consumer Electronics*, vol. 65, no. 3, pp. 388–397, 2019.

[3] N. Neshenko, E. Bou-Harb, J. Crichigno, G. Kaddoum, and N. Ghani, "Demystifying IoT Security: An Exhaustive Survey on IoT Vulnerabilities and a First Empirical Look on Internet-Scale IoT Exploitations," *IEEE Communications Surveys Tutorials*, vol. 21, no. 3, pp. 2702–2733, 2019.

[4] T. Tekeste, H. Saleh, B. Mohammad, and M. Ismail, *IoT for Healthcare: Ultra Low Power ECG Processing System for IoT Devices*. Cham: Springer International Publishing, 2019, pp. 7–12.

[5] Y. Zhang, M. Qiu, C. Tsai, M. M. Hassan, and A. Alamri, "Health-CPS: Healthcare Cyber-Physical System Assisted by Cloud and Big Data," *IEEE Systems Journal*, vol. 11, no. 1, pp. 88–95, 2017.

[6] X. Xu, X. Zhang, H. Gao, Y. Xue, L. Qi, and W. Dou, "BeCome: Blockchain-Enabled Computation Offloading for IoT in Mobile Edge Computing," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4187–4195, 2020.

[7] S. Biswas, K. Sharif, F. Li, and S. P. Mohanty, "Blockchain for E-Healthcare Systems: Easier Said Than Done," *IEEE Computer*, vol. 53, no. 7, pp. 57–67, 2020.

[8] S. Biswas, K. Sharif, F. Li, S. Maharjan, S. P. Mohanty, and Y. Wang, "PoBT: A Lightweight Consensus Algorithm for Scalable IoT Business Blockchain," *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 2343–2355, 2020.

[9] E. Bhaskara Santhosh, S. Priyanka, and A. K. Pradhan, "SHPI: Smart Healthcare System for Patients in ICU using IoT," in *Advanced Networks and Telecommunications Systems*, 2019, pp. 1–6.

[10] T. M. Fernández-Caramés and P. Fraga-Lamas, "A Review on the Use of Blockchain for the Internet of Things," *IEEE Access*, vol. 6, pp. 32 979–33 001, 2018.

[11] F. Klemm, S. Girdzijauskas, J. Le Boudec, and K. Aberer, "On routing in distributed hash tables," in *Seventh IEEE International Conference on Peer-to-Peer Computing*, 2007, pp. 113–122.

[12] T. T. A. Dinh, J. Wang, G. Chen, R. Liu, B. C. Ooi, and K. L. Tan, "BLOCKBENCH: A Framework for Analyzing Private Blockchains," in *Proceedings of the ACM International Conference on Management of Data*, 2017, p. 1085–1100.

[13] M. H. Ghahramani, M. Zhou, and C. T. Hon, "Toward cloud computing QoS architecture: analysis of cloud systems and cloud services," *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 1, pp. 6–18, 2017.

[14] Z. Ying, L. Wei, Q. Li, X. Liu, and J. Cui, "A Lightweight Policy Preserving EHR Sharing Scheme in the Cloud," *IEEE Access*, vol. 6, pp. 53 698–53 708, 2018.

[15] Q. Xia, E. B. Sifah, K. O. Asamoah, J. Gao, X. Du, and M. Guizani, "Medshare: Trust-less medical data sharing among cloud service providers via blockchain," *IEEE Access*, vol. 5, pp. 14 757–14 767, 2017.

[16] Y. Zhang, D. He, and K.-K. R. Choo, "BaDS: Blockchain-Based Architecture for Data Sharing with ABS and CP-ABE in IoT," *Wireless Communications and Mobile Computing*, vol. 2018, pp. 1–9, 2018.

[17] D. C. Nguyen, P. N. Pathirana, M. Ding, and A. Seneviratne, "Blockchain for Secure EHRs Sharing of Mobile Cloud Based E-Health Systems," *IEEE Access*, vol. 7, pp. 66 792–66 806, 2019.

[18] S. Wang, Y. Zhang, and Y. Zhang, "A Blockchain-Based Framework for Data Sharing With Fine-Grained Access Control in Decentralized Storage Systems," *IEEE Access*, vol. 6, pp. 38 437–38 450, 2018.

[19] Y. Zhang, S. Kasahara, Y. Shen, X. Jiang, and J. Wan, "Smart Contract-Based Access Control for the Internet of Things," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1594–1605, 2019.

[20] R. Xu, Y. Chen, E. Blasch, and G. Chen, "BlendCAC: A BLockchain-Enabled Decentralized Capability-Based Access Control for IoTs," in *Proc. IEEE International Conference on Internet of Things (iThings)*, 2018, pp. 1027–1034.

[21] O. Novo, "Blockchain Meets IoT: An Architecture for Scalable Access Management in IoT," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1184–1195, 2018.

[22] X. Huang, R. Yu, J. Kang, and Y. Zhang, "Distributed Reputation Management for Secure and Efficient Vehicular Edge Computing and Networks," *IEEE Access*, vol. 5, pp. 25 408–25 420, 2017.

[23] A. H. Sodhro, Z. Luo, A. K. Sangaiah, and S. W. Baik, "Mobile edge computing based QoS optimization in medical healthcare applications," *International Journal of Information Management*, vol. 45, p. 308–318, 2019.

[24] T. Gong, H. Huang, P. Li, K. Zhang, and H. Jiang, "A Medical Healthcare System for Privacy Protection Based on IoT," in *2015 Seventh International Symposium on Parallel Architectures, Algorithms and Programming*, 2015, pp. 217–222.

[25] P. Zhang, M. A. Walker, J. White, D. C. Schmidt, and G. Lenz, "Metrics for assessing blockchain-based healthcare decentralized apps," in *Proc. IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom)*, 2017, pp. 1–4.

[26] P. Pace, G. Aloi, R. Gravina, G. Caliciuri, G. Fortino, and A. Liotta, "An Edge-Based Architecture to Support Efficient Applications for Healthcare Industry 4.0," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 1, pp. 481–489, 2019.

[27] A. Azaria, A. Ekblaw, T. Vieira, and A. Lippman, "Medrec: Using blockchain for medical data access and permission management," in *2016 2nd International Conference on Open and Big Data (OBD)*, 2016, pp. 25–30.

[28] G. G. Dagher, J. Mohler, M. Milojkovic, and P. B. Marella, "Ancile: Privacy-Preserving Framework for Access Control and Interoperability of Electronic Health Records Using Blockchain Technology," *Sustainable Cities and Society*, vol. 39, pp. 283 – 297, 2018.

[29] B. B. Gonçalves, "Secure storage and sharing of health data in a Blockchain environment," 2018, pp. 1–74, nOVA University Lisbon. [Online]. Available: https://run.unl.pt/bitstream/10362/58228/1/Goncalves_2018.pdf

[30] S. Alkhushayni, D. Al-Zaleq, and N. L. G. Kengne, "Blockchain Technology applied to Electronic Health Records," in *International Conference on Computer Applications in Industry and Engineering*, ser. EPiC Series in Computing, vol. 63. EasyChair, 2019, pp. 34–42.

[31] R. Angeles, "Blockchain-Based Healthcare: Three Successful Proof-of-Concept Pilots Worth Considering," *Journal of International Technology and Information Management*, vol. 27, pp. 47–83, 2019.

[32] K. Ito, K. Tago, and Q. Jin, "i-Blockchain: A Blockchain-Empowered Individual-Centric Framework for Privacy-Preserved Use of Personal Health Data," 2018, pp. 829–833.

[33] S. Niu, L. Chen, J. Wang, and F. Yu, "Electronic Health Record Sharing Scheme With Searchable Attribute-Based Encryption on Blockchain," *IEEE Access*, vol. 8, pp. 7195–7204, 2020.

[34] X. Yang, T. Li, X. Pei, L. Wen, and C. Wang, "Medical Data Sharing Scheme Based on Attribute Cryptosystem and Blockchain Technology," *IEEE Access*, vol. 8, pp. 45 468–45 476, 2020.

[35] V. Ramani, T. Kumar, A. Bracken, M. Liyanage, and M. Ylianttila, "Secure and Efficient Data Accessibility in Blockchain Based Healthcare Systems," in *Proc. IEEE Global Communications Conference (GLOBECOM)*, 2018, pp. 206–212.

[36] X. Liang, J. Zhao, S. Shetty, J. Liu, and D. Li, "Integrating blockchain for data sharing and collaboration in mobile healthcare applications," in *International Symposium on Personal, Indoor, and Mobile Radio Communications*, 2017, pp. 1–5.

[37] Y. Chen, H. Li, K. Li, and J. Zhang, "An improved P2P file system scheme based on IPFS and Blockchain," in *Proc. IEEE International Conference on Big Data (Big Data)*, 2017, pp. 2652–2657.

[38] T. N. Gia, M. Jiang, A. Rahmani, T. Westerlund, P. Liljeberg, and H. Tenhunen, "Fog Computing in Healthcare Internet of Things: A Case Study on ECG Feature Extraction," in *Proc. IEEE International Conference on Computer and Information Technology*, 2015, pp. 356–363.

[39] S. Biswas, K. Sharif, F. Li, S. S. Kanhere, Z. Latif, and S. P. Mohanty, "Interoperability and Synchronization Management of Blockchain Based Decentralized e-Health Systems," *IEEE Transactions on Engineering Management*, no. 10.1109/TEM.2020.2989779, pp. 1–14, 2020, accepted on 15 April 2020.

[40] L. Rachakonda, A. K. Bapatla, S. P. Mohanty, and E. Kougianos, "SaYoPillow: Blockchain-Enabled Privacy-Assured Framework for Stress Detection, Prediction and Control Considering Sleeping Habits in the IoMT," *arXiv Computer Science*, no. arXiv:2007.07377, July 2020.

[41] R. A. Rahman and B. Shah, "Security analysis of IoT protocols: A focus in CoAP," in *3rd MEC International Conference on Big Data and Smart City (ICBDSC)*, 2016, pp. 1–7.

[42] J. Park, H. Kwon, and N. Kang, "IoT–Cloud collaboration to establish a secure connection for lightweight devices," *Wireless Networks*, vol. 23, p. 681–692, 2016.

[43] M. Borysov, "Building Private IPFS Network with IPFS-Cluster for Data Replication," accessed: 20-Jun-2020. [Online]. Available: https://labs.eleks.com/2019/

03/ipfs-network-data-replication.html.
[44] D. J. Bernstein and T. Lange, "Faster Addition and Doubling on Elliptic Curves," *IACR Cryptology ePrint Archive*, vol. 4833, pp. 29–50, 2007.
[45] B. Santhosh, "Fortified-chain," https://github.com/BhaskaraSanthosh/Fortified-Chain, 2020.
[46] S. Bragagnolo, H. Rocha, M. Denker, and S. Ducasse, "SmartInspect: solidity smart contract inspector," in *Proc. International Workshop on Blockchain Oriented Software Engineering (IWBOSE)*, 2018, pp. 9–18.
[47] Web3js, "web3.js - Ethereum JavaScript API - web3.js 1.0.0 documentation," 2019.
[48] DataWorld, "Electronic Health Record (EHR) datasets." [Online]. Available: https://data.world/datasets/ehr
[49] kaggle, "Electronic Health Record (EHR) datasets." [Online]. Available: https://www.kaggle.com/datasets

Bhaskara S. Egala received his bachelor's degree in Information Technology from JNTU-K University, in 2011. He has received Post Graduation Diploma in IT Infrastructure, Systems and Security (PG-DITISS) from Centre for Development of Advanced Computing, Pune, 2013. He then commenced his master's in Cyber Security from JNTU-K University, in 2016. Now, he is pursuing Ph.D. degree in SRM University, Amaravati, AP. His current research interest covers security and privacy concerns in the context of the Internet of Things (IoT) and Smart Healthcare systems.

Ashok K. Pradhan is currently working as an Assistant Professor in the Department of Computer Science & Engineering, School of Engineering and Applied Science at SRM University, Amaravati, AP. He has received his M. Tech degree in the Department of Computer Science and Engineering from the National Institute of Technology (NIT), Rourkela, India, 2010. He has received his Ph.D. degree in the Department of Computer Science and Engineering NIT Durgapur, India, 2015. His areas of interest and research includes Optical Communication and Networks, Internet of Things (IoT), Blockchain Technology, Network Security & Privacy, Cloud Computing, Edge Computing, Fog Computing, and Computer Algorithms.

Venkata R. Badarla is currently working as an Associate Professor in the Department of Computer Science & Engineering at Indian Institute of Technology, Tirupati, AP. He has received his M.E degree in the Department of Information Systems from the Birla Institute of Technology and Science, Pilani, India, 1997. He has received his Ph.D. degree in the Department of Computer Science and Engineering, from Indian Institute of Technology, Madras, India, 2007. His areas of interest and research includes Wireless Networks, Cloud Computing, and Internet of Things (IoT). He has published more than 20 research papers in reputed peer-reviewed journals/conferences with high impact factors.

Saraju P. Mohanty (SM'08) received the bachelor's degree (Honors) in electrical engineering from the Orissa University of Agriculture and Technology, Bhubaneswar, in 1995, the master's degree in Systems Science and Automation from the Indian Institute of Science, Bengaluru, in 1999, and the Ph.D. degree in Computer Science and Engineering from the University of South Florida, Tampa, in 2003. He is a Professor with the University of North Texas. His research is in "Smart Electronic Systems" which has been funded by National Science Foundations (NSF), Semiconductor Research Corporation (SRC), U.S. Air Force, IUSSTF, and Mission Innovation. He has authored 350 research articles, 4 books, and invented 4 granted and 1 pending patents. His Google Scholar h-index is 39 and i10-index is 149 with 6800 citations. He is regarded as a visionary researcher on Smart Cities technology in which his research deals with security and energy aware, and AI/ML-integrated smart components. He introduced the Secure Digital Camera (SDC) in 2004 with built-in security features designed using Hardware-Assisted Security (HAS) or Security by Design (SbD) principle. He is widely credited as the designer for the first digital watermarking chip in 2004 and first the low-power digital watermarking chip in 2006. He is a recipient of 12 best paper awards, Fulbright Specialist Award in 2020, IEEE Consumer Technology Society Outstanding Service Award in 2020, the IEEE-CS-TCVLSI Distinguished Leadership Award in 2018, and the PROSE Award for Best Textbook in Physical Sciences and Mathematics category in 2016. He has delivered 10 keynotes and served on 11 panels at various International Conferences. He has been serving on the editorial board of several peer-reviewed international journals, including IEEE Transactions on Consumer Electronics (TCE), and IEEE Transactions on Big Data (TBD). He is the Editor-in-Chief (EiC) of the IEEE Consumer Electronics Magazine (MCE). He has been serving on the Board of Governors (BoG) of the IEEE Consumer Technology Society, and has served as the Chair of Technical Committee on Very Large Scale Integration (TCVLSI), IEEE Computer Society (IEEE-CS) during 2014-2018. He is the founding steering committee chair for the IEEE International Symposium on Smart Electronic Systems (iSES), steering committee vice-chair of the IEEE-CS Symposium on VLSI (ISVLSI), and steering committee vice-chair of the OITS International Conference on Information Technology (ICIT). He has mentored 2 post-doctoral researchers, and supervised 12 Ph.D. dissertations, 26 M.S. thesis, and 10 undergraduate projects.