BIoTHR: Electronic Health Record Servicing Scheme in IoT-Blockchain Ecosystem

Partha Pratim Ray, *Senior Member, IEEE*, Biky Chowhan, *Member, IEEE*, Neeraj Kumar, *Senior Member, IEEE*, and Ahmad Almogren, *Senior Member, IEEE*

***Abstract*—The pervasiveness of newly introduced Internet-of-Things (IoT) devices has opened up new opportunities in healthcare systems, for example in facilitating remote patient monitoring. There are, however, security and privacy considerations in the transmission of data from these devices to the backend server, and across heterogeneous IoT networks. In this study, we propose a novel privacy-preserving scheme which is based on blockchain and swarm exchange techniques to facilitate seamless and secure transmission of user data (e.g., electronic health record (EHR)-related information) through secured swarm nodes of peer-to-peer communications. BIoTHR refers to the proposed scheme on the private blockchain-assisted EHR management using IoT. Specifically, new blockchain and swarm exchange infrastructures are suggested as a backbone of the proposed scheme to ensure secure and reliable data transmission and timely monitoring of data sent across IoT networks. An autonomous encryption–decryption mechanism is also utilized, along with a dynamic and modular server assistance technology to deploy EHR transmission in a secure manner. Moreover, several swarm-listen, announcement, peer open and peer closing algorithms are incorporated to employ the actual power of pervasive EHR transmission for better e-healthcare service provisioning. The proposed scheme is developed using the open-source tools of GnuPG, IPFS, and Golang. Proposed study simulates a number of heterogeneous IoT-based health sensor nodes, namely, body temperature, pulse rate, and oxygen saturation, i.e., SPO2, galvanic skin response, and blood glucose in blockchain-assisted swarm exchange framework. The results reveal that the proposed scheme, in terms of blockchain-IoT, swarm exchange and EHR transmission, outperforms several peer techniques.**

***Index Terms*—Blockchain, healthcare systems, Internet of Things (IoT), privacy preservation, service management.**

Manuscript received December 18, 2020; accepted January 6, 2021. Date of publication January 11, 2021; date of current version June 23, 2021. This work was supported by the Deanship of Scientific Research, King Saud University, through the Vice Deanship of Scientific Research Chairs. *(Corresponding author: Ahmad Almogren.)*

Partha Pratim Ray is with the Department of Computer Applications, Sikkim University, Gangtok 737102, India (e-mail: ppray@ieee.org).

Biky Chowhan is with the National Institute of Electronics and Information Technology, Gangtok 737101, India (e-mail: bikychowhan.slg@hotmail.com).

Neeraj Kumar is with the Department of Computer Science and Engineering, TIET, Patiala 147004, India, also with the School of Computing, University of Petroleum and Energy Studies, Dehradun 248007, Uttarakhand, also with the Department of Computer Science and Information Engineering, Asia University, Taichung City 41354, Taiwan, and also with the Department of Informatics, King Abdulaziz University, Jeddah 21589, Saudi Arabia (e-mail: neeraj.kumar@thapar.edu).

Ahmad Almogren is with the Chair of Cyber Security, Department of Computer Science, College of Computer and Information Sciences, King Saud University, Riyadh 11633, Saudi Arabia (e-mail: Ahalmogren@ksu.edu.sa).

Digital Object Identifier 10.1109/JIOT.2021.3050703

I. Introduction

With the emergence of the Internet-of-Things (IoT) healthcare systems, a large volume of medical data is recorded and transferred through IoT networks [1]. Remote patient monitoring (RPM) is one of many such technologies within an IoT healthcare network. Specifically, RPM facilitates the monitoring and treatment of patients outside the conventional healthcare setting [2], [3], [40], using data collected from different IoT devices and other sources. For example, IoT wearable devices sense patients' data and transmit them to hospitals or medical institutions. The data are then stored as part of the patient's electronic health records (EHRs) at backend databases and storage systems. There are clearly security and privacy considerations when patient data is sent through heterogeneous IoT networks [2], [41]–[44].

The healthcare sector has been served by multiple IoT manufacturing and service providers for offering IoT-enabled healthcare sensors, actuators, System-on-Chips (SoC), cloud services and data analytics [5]. Although IoT-enabled healthcare systems improve services to patients and medical institutions, in terms of time and cost, security and privacy remain two key challenges (e.g., how to protect heterogeneous healthcare data sources and transfer them in a secure manner). For example, healthcare and many other critical systems should contain essential protection requirements that include secure transmission, tamper-proof monitoring, data storage, integrity, and privacy-aware authentication [4], [6].

Blockchain is a relatively new technology that could be used to enhance security and privacy levels in the domain of IoT healthcare systems [3]. Because of the immutable feature of blockchain, e-health data can be stored at secure medical blocks to assert the privacy and integrity of data [2], [4], [45]. Due to its decentralized nature, blockchain could be used to achieve some level of security without relying on another intermediary [5]. For example, there have been efforts to integrate blockchain into the design of IoT systems in domains such as smart homes and cities [6]. However, in a healthcare setting, data sources of EHRs involve many structured and unstructured data types, such as text and image, which can be significantly large and impractical to transmit over existing wireless network systems. Existing blockchain and IoT solutions may require systematic adaptations to fit the heterogeneous norm of EHRs, which may also introduce new security and privacy concerns [4]–[6].

In this article, we propose a novel privacy-preserving blockchain-based scheme that relies on the concept of swarm exchange to facilitate smart and secure EHR handling, transmission and monitoring, i.e., BIoTHR. Specifically, the swarm exchange paradigm is designed on the top of the underlying blockchain-IoT schema to improve the security and privacy of IoT healthcare systems. In our approach, we use a private blockchain, whereby patients' EHR data is transferred to authorized stakeholders (e.g., medical doctors) using the swarm exchange paradigm. Doctors can transmit medication or diagnosis reports to patients using the same secure infrastructure. Moreover, the swarm exchange paradigm facilitates EHR data transmission by associating security services to EHR blocks, and using a content-addressable network protocol. This work is a representation of simulation-based preliminary study on the significance of heterogeneous IoT-sensor originated data for concatenation into EHR and efficient processing within the blockchain and swarm framework.

The remainder of this article is organized as follows. The next two sections present relevant background materials relating to blockchain and IoT healthcare systems and their related studies, respectively. Section IV explains the conceptual and implementation details of the proposed BIoTHR scheme. Section V discusses our results and evaluations. Finally, the study is concluded in the last section.

II. Background

A. Basics of Blockchain

Privacy-preserving approaches protect original and confidential data against being accessed by unauthorized third parties [41]. The objectives of privacy-preserving approaches are to alter, convert, distribute and conceal sensitive information, such as the EHRs of IoT healthcare systems, to prevent breaching the original data during processing by third-party systems [42], [43]. IoT networks of healthcare system include wearable devices, such as Fitbit, Fuelband and pacemakers, embedded with software, sensors, and connectivity which enables the wearable devices to connect and transmit data with RPMs and their backend EHRs. Protecting EHRs has become essential in IoT healthcare networks to prevent unauthorized users from disclosing sensitive information of patients [20].

The technology of blockchain could be a perfect solution for protecting EHRs in IoT healthcare networks, which is a chain of blocks that contain information, where a genesis block is the first block in the blockchain created by the first miner who wants to start the chain [4]. The essential attributes of a block are basically the block header, which is composed of a pointer to the previous block, the timestamp, nonce, Merkle root, and the block body that contains transactions. Merkle root is a hash of the entire nodes of a Merkle tree [10]. Merkle trees are widely used to validate the large data structures securely and efficiently, which are commonly used to allow efficient verification of transactions [11].

Blockchain can be divided into three main types: 1) public; 2) private; and 3) consortium [7]. A public blockchain allows any stakeholder, such as miners and users, to access the blocks and transactions. In a private blockchain, stakeholders need to be granted prior consent to join the blockchain, thus it is more restricted. Finally, the consortium blockchain is suitable for enterprises or large business applications where a group of people can grant or revoke access to the blockchain.

The key elements of blockchain are explained as follows.

1) Block is the fundamental building unit of a blockchain. It comprises data, hash values, timestamp, nonce and other customized elements.
2) Node is any digital device in a blockchain. Such devices either store complete or partial ledger in a system to perform several tasks, including block creation, mining of other blocks, communicating with other peer nodes.
3) Hash functions play an important role in blockchain to ensure the integrity of blocks. They are used to create a linkage between previous and next block via a chaining mechanism. Popular hash algorithms used in blockchain are secure hash algorithm (SHA)-1,-2,-3, Whirlpool and RIPEMD [8].
4) Transaction refers to the transmission of a crypto-value from one node to another in blockchain.
5) Miner is a mechanism used to extract blocks' information. Tasks of the miners can be summarized as follows: a) Synching up with the network; b) Transaction validation; c) Block validation; d) Create a new block; e) Perform consensus mechanism; and f) Fetch rewards [8].
6) Peer-to-peer (P2P) network is the backbone of any blockchain system, which allows one node of blockchain to talk with others directly, without a need of any intermediary node.
7) Consensus protocol is a mechanism deployed at different nodes, whereby miners take common decisions. Most popular Consensus mechanisms in IoT-based scenarios are summarized as follows.
    a) Proof of Stake (PoS)—depends on the amount of stake held by any miner. The More stake has the more opportunity to mine a new block [9].
    b) Delegated PoS (DPoS)—is a representative democratic process that is faster in transaction processing but incurs more cost toward centralization [10].
    c) Leased PoS (LPoS)—solves the centrality problem in PoS, enables the nodes with low balances, leases contract and shares the reward with the wealth holders [11].
    d) Proof of Elapsed Time (PoET)—is an improvement over the Proof of Work (PoW) with and extremely low energy consumption capability, where the winning miner is randomly chosen based on a random wait time [12].

B. Benefits of Blockchain in Healthcare

There are multiple benefits of decentralizing IoT Healthcare systems using blockchain technology, as listed in the following elements [13].

1) Interoperability—since various blockchain nodes can generate different block contents, content-wise interoperability becomes an issue to resolve. Fortunately, IoT can provide that facility to convey the same from one

node to a recipient node. Data sources of EHRs can be greatly benefited by such incorporation as many types of EHRs could require to exchange within a blockchain network.

2) Cost Reduction—incorporation of smart EHR transmission mechanisms and blockchain-IoT ecosystem would be helpful in reducing overhead costs that generally occur in mitigating traditional healthcare systems.
3) Enhanced Data CIA triad—data sources of EHRs can be stored at local nodes of the blockchain in in a decentralized and secure manner. Data confidentiality can be achieved by applying encryption techniques, data integrity can be accomplished using hash values of blockchains, and data availability becomes easier than existing health systems [14].
4) Faster Healthcare—blockchain can offer faster IoT healthcare facilities to the patients. This can be accomplished by using wearable IoT devices and configuring cellular technology, such as 5G and low power wide area networking (LPWAN) [14].
5) Immutability—data sources of EHRs are very important to diagnose the patients' health condition. The EHR data must be kept so that no change can be done. the Immutability feature of blockchain can assist in retaining the EHR within an intact block [15].
6) Transparency—a transparent scheme is imposed by the blockchain that can assimilate the genuineness of ongoing e-healthcare services [16]. Patients can participate in the entire process to determine the validation of transparency.

C. Basics of Swarm and Swarm Exchange

Swarm is a paradigm that is used for P2P decentralized file transfer. The content of the shared files is addressed by respective hash, i.e., content-addressable. Swarm is used to store numerous types of files written in JavaScript, HTML, CSS etc., on top deployed decentralized systems. Swarm is a good option if small chunks of files are needed to be transfer and shared on decentralized platform. The main advantage of swarm is "built-in" incentivization in a low-latency plethora. Significance of swarm lies into the idea that existing nodes in a decentralized network like blockchain can fetch the files from a designated source until the source node is alive in the network. The need of server-based hosting services is minimized rather diminished which also leverage location-independent file access facility. Thus, a node in the blockchain infrastructure can obviate the necessity for singular hosting of asked data even if other nodes are not connected to the underlying decentralized network. It results a provision to incentivize other nodes of the network to replicate and store the contents at decentralized locations. Thus, data availability is largely harnessed.

Swarm exchange is the policy that deals with the mechanism of incentivization of various nodes in the decentralized network, communication, and realization of self-stable peer pool. It has far reaching consequences which can alleviate the proof of retrievability while mining a piece of data. This policy has a potential anti-censorship plethora in the content agnostic service provisioning. Swarm exchange is inherently safe against the plausible denial of accountability thus making the system stronger in terms of security and privacy. Swarm exchange policy support data upload/download where two nodes can function from own network protocol, topology and heterogeneity. Such facility encourages to use IoT as a tool to implement any decentralized application.

When the blockchain technology is integrated with the paradigm of swarm exchange, they reliably communicate between nodes of blockchains while transferring data between various peers over IoT healthcare networks. Swarm stands for a bunch of nodes in a blockchain network that communicates with each other in a clustered manner, especially for decentralized storing of data in localized distributed ledgers. Swarm exchange is a process in which swarm nodes establish P2P communications via swarm core API, swarm command line interface (CLI) parser, swarm CLI dispatcher, and associated modules [17]. Swarm nodes are very useful elements for P2P decentralized transfer of data source of EHRs, where a swarm community takes decisions by themselves about what to do, i.e., listen, announce, open or close connection with peers. In this work, the swarm nodes are deployed to exchange and transmit EHRs via blocks of the private blockchain. This helps in uploading and downloading EHR data sources to and from the underlying infrastructure scheme.

D. Trusted Parties

In this work, we utilize private blockchain platform. Thus, all stakeholders in the prescribed scenario are assumed to be trusted parties. Even if an unauthorized or malicious agent tries to intrude the system it won't be successful. Underlying specifications of rigid foundation made on top of swarm core supports the system to work in purely decentralized manner with incentivization. Thus, any stakeholder (e.g., patient, care giver, doctor, medical professional, government agencies, and insurance organizations) having complete authorization can be able to participate and access the EHRs. No EHR tampering is allowed in the proposed system. The complete process is performed in the vicinity of stringent swarm exchange policy that provides transparent data sharing and exchange facility.

III. Related Work

A. Review of Works

A simulation model was being performed to ensure the privacy, security and integrity of EHRs during their data distribution in blockchain [18]. In their proposed system, a transparent, decentralized network had been demonstrated to share EHR data in a medical system through blockchain. A MedBlocks application was developed to efficiently and securely transmit EHRs across network systems. The work was extended to create a fee payment scenario in a decentralized manner [19].

A decentralized mathematical model was developed to harness the EHR service a provisioning capability, where A record relationship contract (RRC) was simulated against a summary contract (SC). The whole process provided a tamper-proof

EHR management system for providing benefits to patients and other stakeholders [20].

An initiative was proposed to resolve the research availability tradeoff and privacy-aware EHR data mitigation by utilizing the American PHIPAA regulation. This initiative discussed the EHR storage network modeling, permission-less availability and operational cost minimization aspect [21].

In [22], an Ethereum-based mobile-cloud-blockchain system was proposed to share the EHR simulations by using a proof of concept integration while allowing IPFS to interact with the AWS EC2 cloud service. A smart contract was designed for testing and validating the proposed data integrity, data privacy, and data availability features. A BHEEM framework was designed to analyze the required decentralization in the EHR stakeholder management and maintenance by paving the healthcare 4.0 [23].

A novel authentication scheme was suggested to enhance the privacy and security aspects of the blockchain-based EHR facility mitigation [24]. Another multiauthority-based blockchain was deployed at the cloud EHR management system using an attribute-based encryption (ABE) scheme to simulate the telemedicine aspect [25].

Guo *et al.* [26] proposed an advanced bi-linear Diffie-Hellman algorithm to provide a secure attribute-based signature authentication scheme in EHR managerial system. In the study, a decentralized e-healthcare architecture was also proposed to analyze the EHR service mitigation using the Exonum blockchain under an open information system model. Further, a big data classification approach was developed to counter various diseases and related e-healthcare problems related to patients' data [27].

This was followed by a study performed in the Estonian system to validate various applications of blockchain technology in the e-healthcare domain [28]. A framework was designed to improve the efficiency and security aspects of e-healthcare systems by implying advanced data storage, data sharing, and decentralized approaches [29]. In more detail, an architecture was designed to facilitate secure access and identity management in EHR transactions. The blockchain size, in this case, was approximated at 3.8 MB for serving the interactive response of 2.5 s [30].

Liu *et al.* [31] proposed a BPDS framework to efficiently store and manage EHRs in a tamper-proof cloud-based consortium blockchain ecosystem. A discussion was carried out to seek the answer to the applicability of the blockchain to emerge as a panacea for the existing cloud-based EHR data security as well as privacy [32].

An emergency access control EHR data management system was developed by combining a hyper ledger composer logic for offering distributed laser centric services to patients, doctors, and emergency EHR data accessibility [33]. EHR building blocks were securely incorporated for the regulation of enhanced circular e-healthcare jurisdictions by involving smart contracts, patient records, and EHR data models [34]. Talukder *et al.* [35] proposed a novel consensus algorithm, namely, Proof of Disease (PoD), to reduce the overhead risk related to the EHR in the blockchain. A blockchain architecture was proposed to manage diabetes-related EHR data in a pervasive manner incorporated with IoT devices [36].

In [46], a LoRAWAN-assisted proof-of-concept implementation is investigated where IoT data is stored in swarm exchange service without using any private blockchain network. Smart contract is made to just access the data stored in the swarm exchange service. However, this work failed to show internal state diagrams and analysis of the system accurately. Here, data is sent or received between IoT node and data analyst thus proper healthcare scenario is not justified. As the work does not use private blockchain for store any information about the scenario, this makes the system more vulnerable against the cyberattacks. Storing data in private blockchain improves security of the system.

Islam and Shin [47] investigated unmanned aerial vehicle (UAV)-based swarm to access IoT data by using blockchain. Before the operation is performed, the UAV swarm shares a shared key for talking with IoT devices. By using π-hash bloom filter and digital signature the validation of the sender is carried. Thus, a two-phase secure system is developed. However, this article does not discuss how their system is stringent enough against the cyberattack. Further, lack of private blockchain makes the system more access friendly that could pave conflict of interest among the users.

In [52], trusted third party auditor enabled EHR framework is devised. Ajayi *et al.* [53] proposed a secure architecture for EHR transmission for health care scenario. IOTA-Tangle is used in [54] to connect EHR with the Tangle platform. Bosri *et al.* [55] proposed the HIDEchain architecture for edge centric health services. Another framework is proposed in [56] for the wearable device analytics. In [57], the BBEHR architecture is proposed to empower biometric-based blockchain service.

The comparison of the proposed scheme with other related works is listed in Table I. The terms of the blockchain-IoT scenario, swarm exchange scheme, EHR transmission and study type are used in the comparison to demonstrate the high capability of the proposed privacy-preserving scheme. There are various reasons for the high performance of the proposed scheme are considered from the time analysis of uploading and downloading HER files while running the scheme.

B. Research Gaps in Existing Literature

Existing works mainly focus on blockchain-based implementation of healthcare security and privacy provisioning. Selective articles show partial IoT alignments in their works. Most of the works are aimed at formulation of mathematical modeling and abstract. Minimal investigations are performed to actually deploy EHRs transmission in an IoT scenario. Swarm exchange scheme is found in none of the papers. Thus, it becomes difficult to justify whether these works can really showcase user privacy, data integrity, ethnicity, and security of highly sensitive EHRs. Moreover, these works do not conform to mitigating the assessment of trusted or untrusted parties into their models.

TABLE I
COMPARISON OF THE PROPOSED SYSTEM WITH RELATED WORKS

Article	Year	Blochchain-IoT	Swarm Exchange Scheme	EHR Transmission	Contributions
R. N. Nortey *et al.* [18]	2019	Only Blockchain	No	Yes	Privacy Modeling
A. Cirstea *et al.* [19]	2018	Only Blockchain	No	No	Platform-based Implementation
G. Yang *et al.* [20]	2018	Only Blockchain	No	Yes	Architecture Modeling
G. Magyar [21]	2017	Only Blockchain	No	No	Theoretical Study
D. C. Nguyen *et al.* [22]	2019	Partial	No	Yes	Proof of Concept
J. Vora *et al.* [23]	2018	Only Blockchain	No	Yes	Platform-based Implementation
F. Tang *et al.* [24]	2019	Only Blockchain	No	Yes	Mathematical Modeling
R. Guo *et al.* [25]	2019	Only Blockchain	No	Yes	Mathematical Modeling & Simulation
R. Guo *et al.* [26]	2018	Only Blockchain	No	Yes	Mathematical Modeling
I. Kotsiuba *et al.* [27]	2018	Only Blockchain	No	Yes	Architecture Modeling
A. Ekn *et al.* [28]	2018	Only Blockchain	No	Yes	Theoretical Discussion
N. Kshetri [29]	2018	Only Blockchain	No	Yes	Theoretical Discussion
T. Mikula *et al.* [30]	2018	Only Blockchain	No	Yes	Platform-based Implementation
A. R. Rajput *et al.* [33]	2018	Only Blockchain	No	Yes	Architecture Modeling
K. R. Ozyilmaz *et al.* [46]	2019	Partial	Yes	No	Preliminary Structure
A. Islam *et al.* [47]	2019	Partial	Yes	No	Mathematical Modeling
R. Jabbar *et al.* [52]	2019	Only Blockchain	No	yes	Third Party Framework Modeling
O. Ajayi *et al.* [53]	2020	No	No	No	Architecture Modeling
E. Saweros *et al.* [54]	2020	Only Blockchain	No	No	Case Study Modeling
R. Bosri *et al.* [55]	2020	No	Yes	No	Architecture Modeling
R. M. Patil *et al.* [56]	2020	No	No	No	Analytics Framework Modeling
M. A. Baqari *et al.* [57]	2020	Only Blockchain	No	No	Biometric System Modeling
Proposed Work	2020	Both IoT and Blockchain	Yes	Yes	Framework Design, Modeling, Implementation and Validation

C. Novelty of Proposed Work

Novelty of our proposed scheme are as follows.

1) Blockchain-centric full Support with EHR utilization
2) Full IoT-based implementation while inclusion of swarm exchange scheme
3) Conceptual architecture modeling, UML activity modeling, novel implementation, swarm private blockchain-IoT infrastructure modeling and deployment
4) We also present the notions of trusted and untrusted parties into our scheme

IV. PROPOSED SCHEME

In this section, we explain the system design and methodology of implementing the proposed privacy-preserving scheme based on blockchain and swarm exchange, i.e., BIoTHR, as depicted in Fig. 1. To develop the proposed scheme, three open-source tools: 1) GPG [37]; 2) IPFS [38]; and 3) Golang [39], are used. Brief details of the tools are discussed below.

A. Tools Used

1) GnuPG—is an open source, RFC 4880 compliant software that uses a hybridized encryption–decryption scheme [37]. It supports both symmetric and asymmetric key generation algorithms. Furthermore, GnuPG facilitates access to public key repositories with the help of an enriched key management system. In this work, we used GnuPG for encryption and decryption of EHR and diagnostic EHR while transferring data between patients and their doctors for accomplishing the EHR data confidentiality.
2) IPFS—is a network-designed P2P protocol that supports content-addressable infrastructure in a distributed environment [38]. It leverages the block-storage scheme where content-addressable hyperlinks can be managed. In this work, we used IPFS for hosting the swarm exchange scheme for efficient EHR data transmission and RPM for achieving data privacy and availability.
3) Golang—is statically-typed concurrency that processes a high-level language developed for granting memory safety and performing garbage collection tasks in an efficient manner [39]. Its inherent virtual inheritance and lightweight go routines help to channelize the static-linked binary files to a production host system. In this article, we used Go language for hosting the IPFS server in swarm nodes and allowing the security of EHR data sources.

As shown in Fig. 1, the scheme comprises (a) the patient to doctor EHR data transmission via the private blockchain and (b) the doctor to patient diagnosis EHR data transmission via the same infrastructure. The swarm exchange paradigm plays the most vital role in assisting the functions of (a) and (b) in a securely and efficiently bidirectional manner.

B. Use of IoT Virtual Nodes

In this scheme, we employ five types of IoT-based health sensor nodes, such as body temperature, pulse rate, SPO2 (blood oxygen level), galvanic skin response (GSR), and blood glucose sensors as the proof-of-concept simulated nodes for generation of virtual patient health data for propagation to the doctor. Such five types of heterogeneous IoT-sensor centric health data are aggregated before encrypting or sending to the blockchain by using an IoT-based data aggregator module. Upon successful aggregation of heterogeneous health data, the IoT-based aggregator module prepares an EHR. We used following forms of generated EHR to investigate the effectiveness

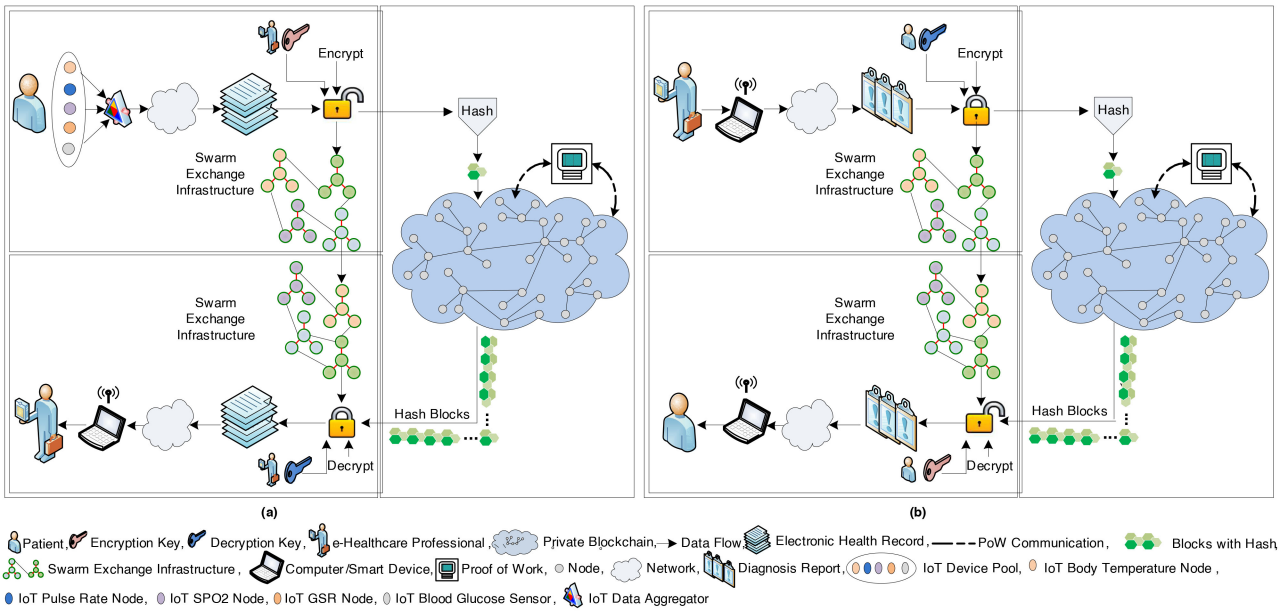

Fig. 1. BIoTHR: Proposed privacy-preserving blockchain-based scheme for EHR using a swarm exchange paradigm. (a) Patient to EHR medical professional transmission system. (b) Medical professional to patient diagnosis report transmission.

of proposed swarm scheme to validate that includes text (.txt), regular docx (.docx), and pdf (.pdf). The revised EHR format is then fit into the proposed swarm scheme and blockchain infrastructure for testing purpose.

Use of aforementioned IoT-based sensor nodes allow to understand how heterogeneity in the IoT-based environment could be seamlessly augmented for improved health service mitigation under the blockchain-assisted ecosystem. IoT-based aggregator concatenates the sensor data to form a string with predefined details of patient's name, age, sex, and address. It also adds a time stamp in the EHR for keeping record of the instance. IoT-based sensors and aggregator performs the data sensing and aggregation in 10 min interval. However, such jobs could be changed with prior setup. In some cases, we imposed some extra information about health including more textual details about the patient's case history or sample images for making the EHR a bit complex and heavy in terms of file model.

C. Design Principle

In Fig. 1(a), first, a patient uses a swarm node to encrypt EHR data with a prescribed doctor's public key using a hybridized key encryption technique. The encrypted file is then uploaded to the listening swarm exchanger module to announce the reception of other swarm nodes in the swarm exchange infrastructure. Simultaneously, the EHR file gets hashed using a standard SHA-3 algorithm and insert its value inside a block for further mining process. The miners of the private blockchain perform mining over this block and after successful mining, the block added it in the chain. There are two output files from the process above: 1) encrypted EHR file and 2) a hash value of the encrypted EHR file in two systems: a) swarm exchange infrastructure and b) private blockchain, respectively.

A designated doctor needs to perform decryption using a hybridized key decryption technique where three inputs are given, namely, an encrypted EHR file in the swarm exchange infrastructure, hash on a block of the private blockchain and own private key. If the hash of the block matches with the input encrypted-EHR file's hash then only the EHR file is downloaded from the swarm exchange infrastructure to the doctor's swarm node. Otherwise, an attacker could sometimes be able to download the encrypted EHR file, but cannot open the EHR file as the hash does not match. In this essence, the proposed scenario provides two benefits: 1) security to the EHR file and 2) privacy of block generation using the hash value of the EHR file. This results in dual-layer security and privacy to the scenario where swarm nodes would act as IoT healthcare-enabled nodes. In this novel scheme, by the help of a swarm exchange paradigm, all the swarm nodes are updated with the inclusion of newly encrypted EHR files into the system and as per the requirement, the EHR file is handed over to the specific swarm node. Blocks of EHR-hashes provide dual-layer over the earlier mentioned paradigm, where immutability and transparency are achieved by the private blockchain where EHR data is only accessible within the prespecified group of nodes.

D. Swarm Exchange Scheme

This proposed scheme presents a novel swarm exchange paradigm in which the entire swarm nodes either listen or announce about upload or download of EHR file within the infrastructure. This results in a robust and tamper-proof system for leveraging a dual-layer protection mechanism for the EHR file. In Fig. 1(b), a reverse-engineering scenario of (a) takes place, whereby a doctor starts the encryption of the diagnosis report that has been prepared to counter the patient's EHR data. All the swarm exchange and hashing of the diagnosis report seamlessly traverses in a similar fashion of (a) within

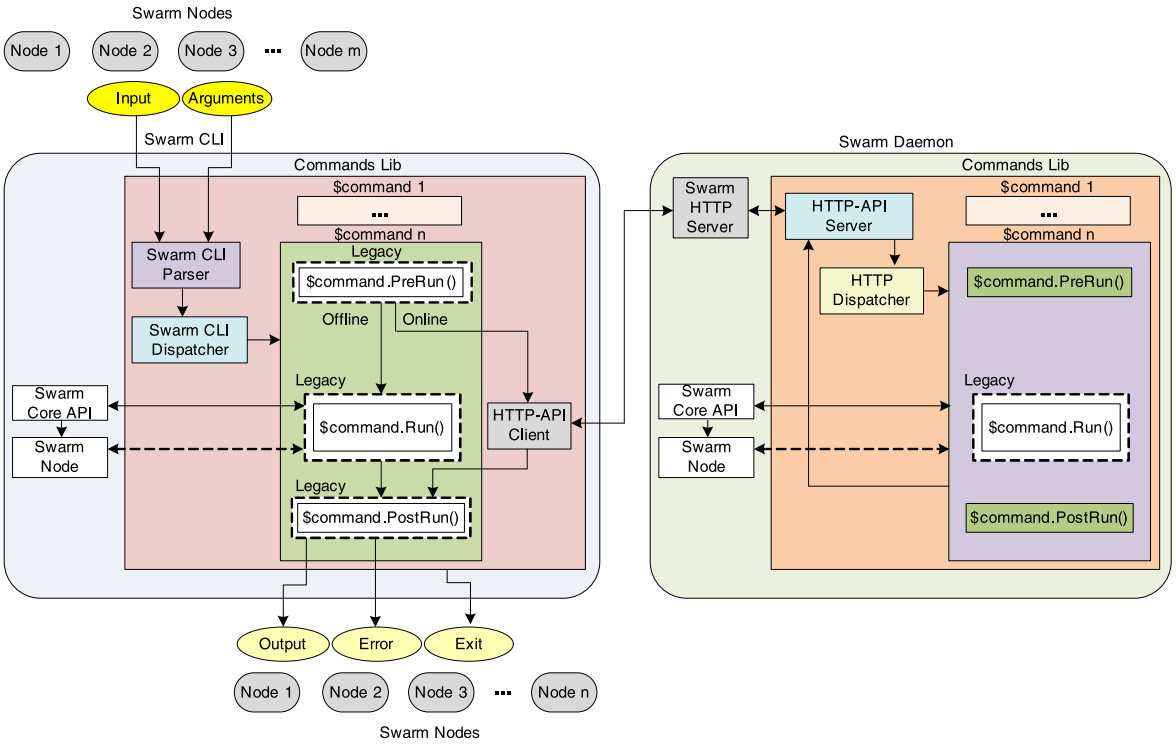

Fig. 2. Core model of swarm exchange paradigm.

the system. Finally, the patient downloads and reads the diagnosis EHR data that was sent by the designated doctor to take necessary actions to treat the disease or health condition. Both (a) and (b), the dual-layer protection mechanism seamlessly operates to enhance the security, privacy, immutability, and transparency of the EHR data sources. The conjunction of swarm exchange paradigm, secure hash block creation, and use of hybridized encryption–decryption techniques that allow the proposed scheme to be very stringent, rigid and tamper-proof in nature.

The swarm exchange paradigm consists of two parts: 1) swarm CLI and 2) swarm Daemon, as shown in Fig. 2. Listening swarm nodes provide a valid EHR related input and associated arguments to the swarm CLI, where the inputs are first parsed in a proper command format, followed by swarm CLI dispatcher that dispatches the inputs to the command module. The command module can process *n* number of commands via legacy *PreRun(), Run()*, and *PostRun()* methods. The most important aspect of this command module is that it can run in an online or offline mode. During the offline mode, EHR files directly sent from the swarm CLI dispatcher to the output segments for reaching out other swarm nodes. In the online mode, swarm daemon activates the swarm HTTP server to serve various EHR files via a similar command module. In both cases, swarm core API and core swarm nodes communicate with each other in a P2P communication to leverage the expected dissemination of EHR data. The HTTP API server and dispatcher support swarm daemon to efficiently perform the swarm exchange operations. This paradigm helps the swarm nodes greatly to provide dual options for facilitating the flexible task modeling. Thus, patient and doctors' swarm nodes can communicate with each other in high level of security and availability, resulting smoothing transmission of EHR files.

E. Proposed Algorithms of Scheme

Multiple algorithms are proposed to demonstrate EHR file transmission in a secure and effective manner. The algorithms are discussed as follows.

Algorithm 1 is used to create a private blockchain for authenticating data exchange of EHRs. The SHA-3 algorithm is used to generate hash values of the encrypted EHR files. Once a swarm node announces a new EHR file, the hash value is promptly taken by the miners to append in the private EHR-healthcare chain after validating the nonce value over the newly created hash block. After a successful mining operation, the hash block is updated in all the nodes' ledgers of the blockchain for further using it.

Algorithm 2 demonstrates warm peers to whom the current node is connected. The *SwarmPeer()* method initializes the verbose, latency, streams, and direction based on the preliminary set up requirement. A connection direction is decided, whether it is from a patient to a doctor or vice-versa. The latency of connection is determined for the successful functioning of stream processing of the EHR files. Finally, the encoding of the outcomes is executed for efficient packetization of the EHR files.

Algorithm 3 is used to list the set of swarm peers about whom the current swarm node is aware of. Initially, it maps the swarm local address that listens to each other. Then, it runs SwarmAddr() to aware of the current swarm node about the list of local swarm nodes that are known to it. Once, current swarm node knows about the local swarm nodes, it then prepares a

Algorithm 1: Private Blockchain Creation

```
Data: Initialize self, no, nonce, data, hashcode, prev
Result: Blockc creation using no, nonce, self.data,
        hashcode, prev
if len self.chain == 0 then
    prev = "0"
else
    prev = self.chain[-1].hashcode
myHash = convert hashlib to .hexdigest
Call block with data no, nonce, data, myHash, prev
self.chain.append block
for no in range self.chain do
    chainDict[no] = self.chain no to getStringVal
    Check wheather the chain is broken
    Mining Block: block.getStringVal
    self.mineBlock with block
while myHash[0:4] ! = self.prefix do
    myHash = hashlib with
     sha256(str(str(nonce)+str(block.data)) .to hexdigest
myHash = self.chain[block.no].hashcode
self.chain[block.no].nonce = nonce
if block.no ! len(self.chain) − 1 then
    myHash = self.chain[block.no + 1].prev
def chechIfBroken(self)
for no in range(len(self.chain)) do
    if self.chain[no].hashcode[0:4] == self.prefix then
        pass
    else
        self.chain[no]
        initialize self, no, data
        self.chain[no].data = data
        self chain[no].hashcode = hashlib with sha256(str
         (str(nonce) + str(block, data)).to hexdigest)
```

Algorithm 2: Swarm Listing

```
Data: Import context, fmt, io, path, sort, sync, time
Result: List Set of Peers to Whom Current Node is
        Connected
Create Swarm peer variable:: SwarmPeer
Run Swarm::SwarmPeer ()
if No error then
    Initialize verbose, latency, streams, direction
for no in range self.chain do
    Initilize con_i := conInfo addr := con_i.adress, peer
     := con_i.ID if verbose||direction then
        Initilize con_i := con_i.direction
if verbose||latency then
    if error := con_i.latency is Nil then
        if lat_vaue == 0 then
            con_i.latency := lat_value
if verbose||streams then
    if error := con_i.streams is Nil then
        if lat_vaue == 0 then
            con_i.latency := lat_value
    for s in range strs do
        con_i.streams := con_i.streams + protocol stream
         info
    return out_peers + con_i
Encode latency, peers, direction, stream info
```

Algorithm 3: Swarm Awareness

```
Data: Import context, fmt, io, path, sort, sync, time
Result: List Set of Peers about Whom Current Node is
        Aware of
Create Swarm address variable:: SwarmAddr
Map Local Swarm address variable:: SwarmAddrLocal
Map Listening Swarm address variable::
 SwarmAddrListen
Run Swarm::SwarmAddr ()
if No error then
    Initialize error := SwarmAddr.Known
for pAddr in range addr do
    for a in range pAddr do
        out := append (out + a)
    return SwarmAddr + out
Encode SwarmAddr
```

refined list of swarm nodes that it is aware of. Thus, whenever an incident takes place in the current swarm node's vicinity via an append operation, it is instantly made aware of the situation to take necessary actions.

Algorithm 4 generates a list of local swarm listening addresses. *SwarmAddrLocal()* method initializes the context-awareness feature of the current swarm node. It runs a continuous loop until all listening swarm nodes are enlisted into the designated panel. A joining operation is performed to join each of the swarm nodes that are currently in a listening mode. Such listening swarm nodes upon enlisting get announced by the current swarm node. It helps other swarm nodes to get ready about the nodes that are listing to the node's periphery.

Algorithm 5 illustrates how the current node gets the list of interfaces listening swarm nodes for announcing their presence to the other swarm nodes. In the algorithm, the current swarm node lists interface listening addresses for the announcement. It runs SwarmAddrInterface() method to append local listening addresses with the current node's interface listening list.

Algorithm 6 demonstrates the opening swarm connections that assist one swarm node to open a connection for another peer, i.e., P2P. A direct connection path is opened between two swarm nodes for transmission of EHR files. A connect() operation is performed to open a connection between such nodes.

Algorithm 4: Swarm Local Listening Address Announcement

Data: Import context, fmt, io, path, sort, sync, time
Result: List Local Listening Address

```
Run Swarm::SwarmAddrLocal ()
if No error then
    Initialize error := SwarmAddrLocal.Context
for addr in range maddr do
    Initialize saddr := maddr if showid is not Nil then
        saddr := join(saddr+ID)
    return SwarmAddrLocal + saddr
Encode SwarmAddrLocal
```

Algorithm 5: Swarm Interface Listening Address Announcement

Data: Import context, fmt, io, path, sort, sync, time
Result: List Interface Addresses the Current Node is Listening on

```
Run Swarm::SwarmAddrInterface ()
if No error then
    Initialize error := SwarmAddrInterface.Context
for addr in range maddr do
    addrs := append(addr + assrs)
    return SwarmAddrLocal + addrs
Encode SwarmAddrInterface
```

Algorithm 6: Swarm Connection Opening

Data: Import context, fmt, io, path, sort, sync, time
Result: Opens Direct Connection to a Swarm Peer

```
Run Swarm::SwarmAddrConnOpen ()
if No error then
    Initialize error := parseAddr.Context
for pi in range pis do
    output := connect(pi + ID)
    return SwarmAddrConnOpen + output
Encode SwarmAddrConnOpen
```

Algorithm 7: Swarm Connection Closing

Data: Import context, fmt, io, path, sort, sync, time
Result: Closes Connection to a Swarm Peer

```
Run Swarm::SwarmAddrConnClose ()
if No error then
    Initialize error := parseAddr.Context
output := addr.len()
for ainfo in range addrs do
    for addr in range maddrs do
        message := disconnect(addr + ainfo.ID) message
          := append(message + output)
    return SwarmAddrConnOpen + output
Encode SwarmAddrConnOpen
```

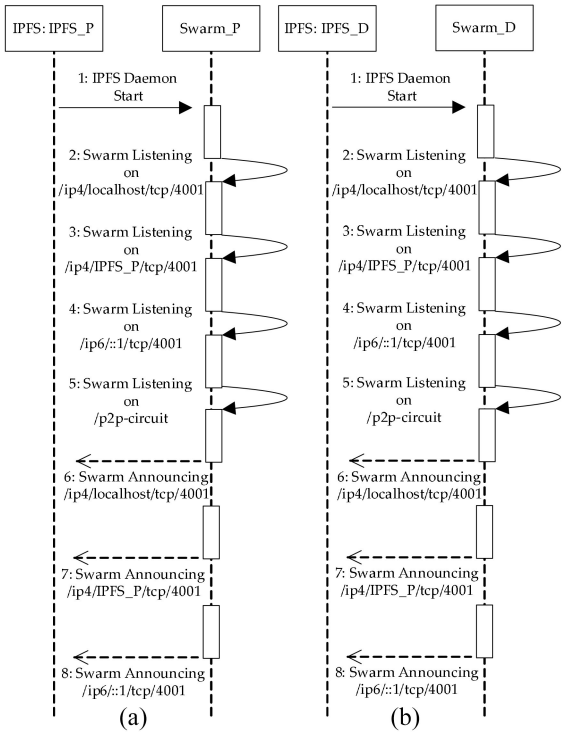

Fig. 3. UML sequence chart for swarm listening and announcement. (a) Patient's swarm activity. (b) Doctor's swarm activity.

Algorithm 7 presents the mechanism as opposed to what is presented in Algorithm 6 that closes the P2P connection which was earlier opened. In this case, the context of the current swarm node is parsed to check whether any error has occurred or not. It runs two for loops to find the P2P connections which are open and subsequently, terminated with disconnect() method. To reflect the closing of connections, the closed swarm nodes are appended with a message that is sent to other swarm peers to inform about the closing of connections.

F. UML Activities of Privacy-Preserving System

The UML activities of the proposed privacy-preserving system assist in understanding how various entities of the system perform different activities to provide security and privacy services to IoT-healthcare networks. The detail process of UML is shown in Figs. 3–5, explained below.

The activity starts with a key pair generation using the GPG-based hybridized key generation mechanism that creates key pairs for both patient and doctor. Initially, both the patient and doctor access GPG in local swarm nodes to create instances of GPG_P and GPG_D, respectively. Such instances then generate key fingerprints for both of them. However, the entropy must be kept equal or higher than the threshold levels for the creation of key pairs. The entropy depends on the provided input characters to generate internal random numbers which makes the key generation process harder to break by any intruder. When the process is completed, public and

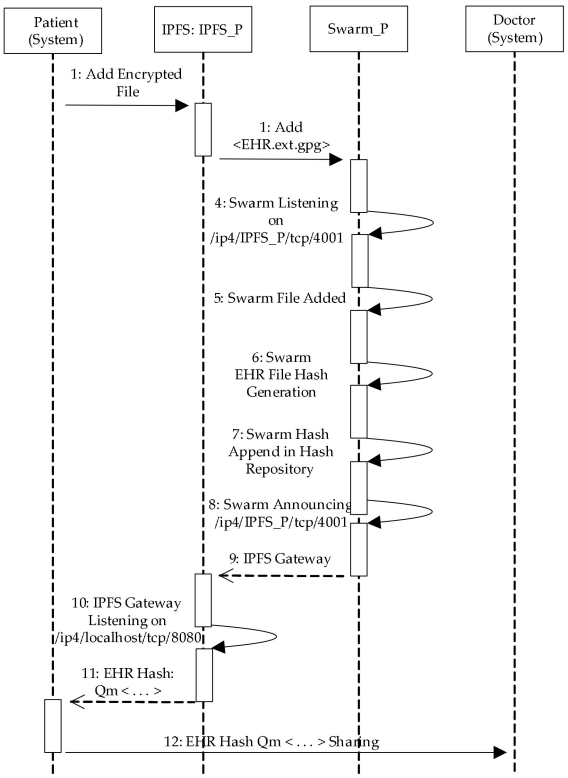

Fig. 4. UML sequence chart for patient's encrypted EHR uploading to IPFS by using the swarm exchange scheme.

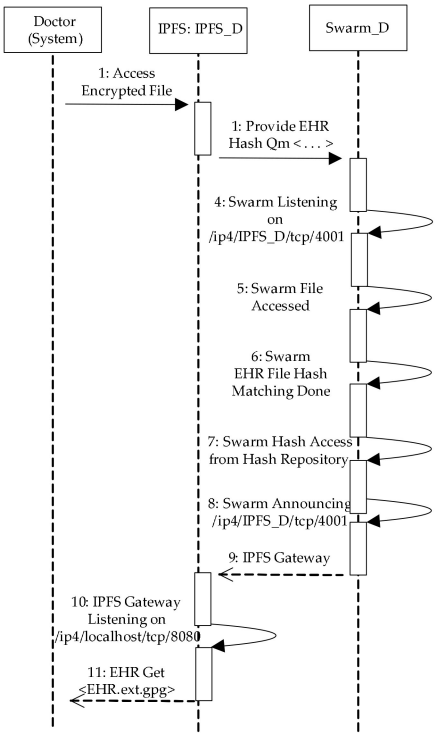

Fig. 5. UML sequence chart for Doctor's access to the encrypted EHR by using shared hash using the swarm exchange scheme.

secret keys are created and signed to be used as fingerprints of individual nodes.

Before the doctor begins participation in the EHR access process, key-ring generation steps must be performed. In this case, doctor exports own key to the GPG_D instance via armored blob and public key inclusion. GPG_D in turn, creates a $<pubkey.asc>$ file in the doctor's swarm node which must be shared secretly with the patient before proceeding to the next stages. We assume that this $<pubkey.asc>$ file-sharing is processed in a secured and trusted environment, thus diminishing the chance of information leakage. The patient then starts the key-ring generation process by importing the $<pubkey.asc>$ into own trusted key list in the swarm node. Later, the patient could determine the validation procedure, where $<pubkey.asc>$ is successfully enlisted into the key-ring of the own key list.

Following that, the patient and doctor have Golang instances Go_P and Go_D, respectively, in *wn* swarm node. When the Golang instances are ready, $<main.go>$, they are executed in both nodes of the patient and doctor. This results in the creation of a listening port on 8080 on both the swarm nodes, thus the respective nodes are ready to the host local swarm exchanger server using IPFS. The patient and doctor create IPFS instances, such as IPFS_P and IPFS_D to go to further stages.

When the servers are up in both swarm nodes of the patient and doctor, the interaction between the node server and swarm instances need to be established, as depicted in Fig. 3. The IPFS_P and IPFS_D instances start the swarm daemon process to communicate with the respective swarm instances Swarm_P and Swarm_D. Both Swarm_P and Swarm_D respond to the respective IPFS instances using swarm listening on /ip4/localhost/tcp/4001, /ip4/IPFS_P/tcp/4001, /ip4/IPFS_D/tcp/4001, /ip6/::1/tcp/4001, and /p2p-circuit. The swarm instances announce the ports via /ip4/localhost/tcp/4001, /ip4/IPFS_P/tcp/4001, /ip4/IPFS_D/tcp/4001, /ip6/::1/tcp/4001. Multiple options are created to listen for announcement purposes. Further, the API server is opened on /ip4/localhost/tcp/4001, the WebUI port is opened on http://localhost:5001/webui, and the gateway (read-only) server listens on /ip4/localhost/tcp/8080.

After successfully completing the phases above, the patient initiates the encryption of the EHR file using a hybridized key encryption mechanism, resulting in a generation of encrypted $<EHR.ext.gpg>$ file. It is worth mentioning that .ext denotes the file extensions, such as .txt, .docx, .pdf, .png, .jpg, and .tiff. As depicted in Fig. 4, the patient provides the encrypted $<EHR.ext.gpg>$ file to the IPFS_P instance. This instance, in turn, calls the Swarm_P instance to upload the $<EHR.ext.gpg>$ file into the swarm exchange infrastructure. The swarm nodes lists on /ip4/IPFS_P/tcp/4001 that promptly add the file into the system and generates the hash value, which is followed by the inclusion of the hash into the hash repository for swarm access. Swarm announcing nodes on /ip4/IPFS_P/tcp/4001 announces the information via the gateway for other swarm nodes to proceed further. The gateway then returns back the patient with the listening port on /ip4/localhost/tcp/8080 and shows the hash of the EHR beginning with $Qm<\cdots>$. Later, the $Qm<\cdots>$ hash value is shared with the designated doctor's swarm node.

The designated doctor can access the encrypted <*EHR.ext.gpg*> file using IPFS_D instance and Swarm_D instance, as shown in Fig. 5. The doctor first provides the Qm$<\cdots>$ hash of the requested EHR to the Swarm_D instance, where swarm nodes listen on /ip4/IPFS_D/tcp/4001. The file is promptly served to the doctor's swarm node, and the same information is updated within the swarm infrastructure by announcing on /ip4/IPFS_D/tcp/4001. The doctor receives the encrypted EHR file via the gateway on /ip4/localhost/tcp/8080.

The decryption process of the doctor's swarm node is involved in the <*EHR.ext.gpg*> file. On the successful reception of the file, the doctor calls GPG_D instance to decrypt the file using the hybridized decryption mechanism. Initially, the doctor needs to remove the <*.gpg*> from the <*EHR.ext.gpg*> file and provide only <*EHR.ext*> to the instance. Subsequently, the doctor provides the Qm$<\cdots>$ hash value to it, along with the doctor's secret key, i.e., a password. If the hash matches with the encrypted hash of the EHR file, the <*EHR.ext*> is opened for delivering a further diagnosis in a human-readable format. To sum up, the doctor can initiate to transmit the diagnosis EHR file to the patient by following the aforementioned steps in a reverse direction of the steps represented in Fig. 1(b) to send the support to the needy patient in a secure and efficient manner.

V. Results and Evaluation

This section discusses the experimental results of the proposed privacy-preserving scheme. The blockchain and swarm exchange techniques are demonstrated to reveal their capability of securing EHR files in IoT-healthcare systems.

A. Analysis of Swarm Loading Time

The swarm exchange paradigm is applied to a blockchain-IoT scenario of healthcare systems, where the EHR files were sent from a patient to a doctor and vice versa, as the doctor sent the diagnosis EHR to the patient. In both cases, the EHR files were uploaded while sending from a swarm node to another and were downloaded while receiving the earlier EHR sent from the sender swarm node.

Two types of the EHR files: 1) text files (.txt, .pdf, and .docx) and 2) image files (.png, .jpg, and .tiff), are utilized in the experiments. Because the EHR file could be of any type, for example, a simple text file, word file, containing X-Ray or MRI scan images. Both types were simulated in this study for incorporating various use cases of EHR files with different sizes. It is observed that .docx takes more processing time than .pdf or .txt while uploading or downloading them into the swarm exchange infrastructure. The EHR files of the .txt extension took lesser processing than other formats. As shown in Fig. 6(a) and (b), the uploading and downloading time of the files vary between 0.1 and 6.0 MB. The .tiff image of the EHR files consumed more time while uploading and downloading. The EHR files having the .png extension costs less time processing compared with the .tiff image. Fig. 7(a) and (b) present the uploading and downloading time of the EHR files which vary between 0.1 to 6.0 MB.

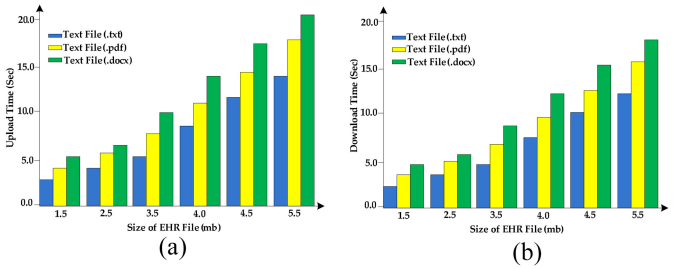

Fig. 6. Swarm loading time of different text file sizes in the swarm exchange infrastructure. (a) Uploading time of text file sizes. (b) Uploading time of text file sizes.

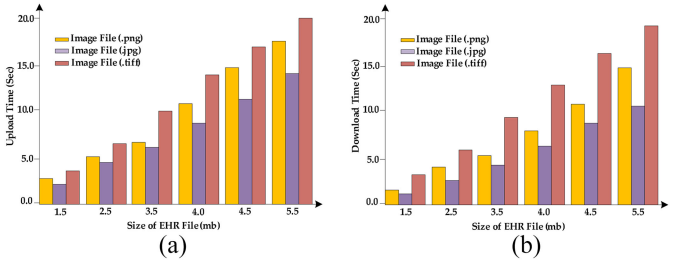

Fig. 7. Loading time of different image file sizes in the swarm exchange infrastructure. (a) Uploading time of image file sizes. (b) Downloading time of image file sizes.

B. Analysis of Swarm Exchange Time

It is noticed that the average sizes of EHR files with their different types have a variation in terms of uploading and downloading times within the swarm exchange infrastructure. Fig. 8(a) and (b) present the uploading and downloading time that show the variations of average EHR size which is between 0.1 and 6.0 MB in the swarm exchange scenario.

The more the average size of the EHR file, the more time it took while loading the swarm exchange scheme. During the uploading swarm exchange time duration, the gradual growth of time was observed, with respect to the increase of the average EHR file size. However, it was not the same case during downloading the EHR file from the swarm exchange infrastructure. There was a horizontal a *S* shaped form in the graph that resembles the sinusoidal curve, whereby the large file size took more time in the downloading phase.

C. Analysis of Swarm Listening Time

Swarm listing is a vital task of the swarm exchange infrastructure, where every swarm node listens to another swarm. Fig. 9(a) and (b) presents the upload time of average EHR file size, and the download time of average EHR file size, respectively. The response time of swarm listening capability is illustrated using the average EHR file sizes in a range of 0.1 and 6.0 MB. It is noted that the swarm listing time increases, along with the increase of average EHR file sizes and their types.

The swarm listening time is represented at the X-axis after a certain threshold of EHR file size. The prespecified range of EHR size is kept constant to understand the impact of swam listing time in IoT and blockchain scenarios. It is also observed that the swarm listening time during download duration is a *U*

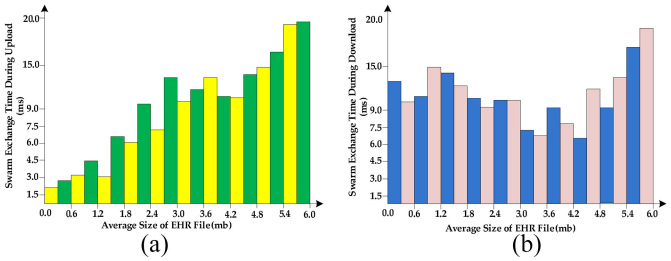

Fig. 8. Swarm exchange time duration. (a) Upload time for average EHR file size. (b) Download time for average EHR file size.

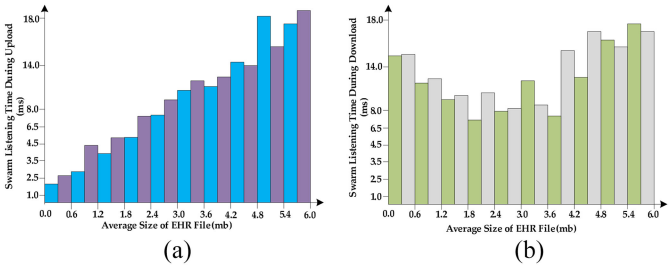

Fig. 9. Swarm listening time duration. (a) Upload time for average EHR file size. (b) Download time for average EHR file size.

shaped form. This results in predicting that the swarm listening time is minimal for average EHR files that have 1.2 to 3.8 MB. This reveals that the EHR file in this range could be quickly downloaded.

D. Analysis of Swarm Announcement Time

Swarm announcement is another vital operation of the swarm exchange infrastructure that is very crucial for handling the EHR files. The swarm announcement feature of the proposed scenario is experimented in this work to observe their impact while processing the EHR files. The swarm announcement time analysis is presented in Fig. 10, where Fig. 10(a) shows the upload time of average EHR file size while Fig. 10(b) represents the download time of the average EHR file size.

It is noticed that the swarm announcement time does not significantly change with the increase of the average EHR file size while uploading, along with a slight upward slope was noticed in the figure. The slope seems to be static after reaching a 4.8-MB cut-off point. In the downloading phase, the swarm announcement was not stable, due to the swarm announcement is randomly executed during the downloading phase. The range of 1.2 to 3.8 MB average EHR size was found to be minimal in the downloading phase.

E. Analysis of Swarm Availability Time

Swarm availability is also an essential factor in the swarm exchange infrastructure. The swarm availability feature is evaluated to determine their effect of handling the EHR files. The swarm availability time analysis is presented in Fig. 11(a) for the uploading time and Fig. 11(b) for the downloading time of the average EHR file size, respectively.

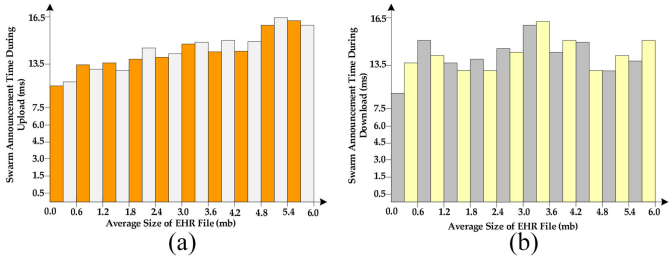

Fig. 10. Swarm announcement time duration. (a) Upload time for average EHR file size. (b) Download time for average EHR file size.

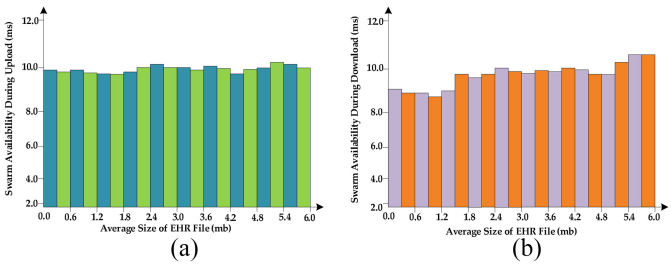

Fig. 11. Swarm availability time duration. (a) Upload time for average EHR file size. (b) Download time for average EHR file size.

It is noted that the swarm availability time does not significantly change with the increase of average EHR file size while uploading. With the change of the average EHR file size, the swarm availability was almost constant, i.e., 10 ms. In the downloading phase, swarm availability constantly fluctuated toward the increase of the average EHR file size because swarm availability is slightly dependent on the size of the EHR file in the downloading stage in an average EHR size of 1.6 to 4.6 MB.

F. Analysis of IoT Elements

We investigate simulated IoT-sensor data centric health data generation time and IoT-based aggregator oriented EHR generation.

Five different types of heterogeneous IoT-based health sensor data simulated as if it were connected to the patient's body physically. We find moderate values of sensor data generation within 1%–2% variance of specific sensor types. We see that temperature, pulse rate, SPO2, GSR, and glucose monitoring data lies in the range as follows, 10%–12%, 11%–13%, 14%–16%, and 29%–31%, respectively. We set interval time at 15 min so that rest of the swarm and blockchain related jobs could be well performed. Fig. 12(a) presets the analysis.

All the IoT-based health sensor data need to be concatenated to form the EHR before sending to the doctor. Thus, we investigate the timing efficiency of the IoT-based sensor data aggregator module in this study. We find EHR generation time lies within the range of 95–103 s while covering all five IoT sensor originated values into one EHR file. Fig. 12(b) shows five heterogeneous IoT-sensor oriented data aggregation at the IoT aggregator module.

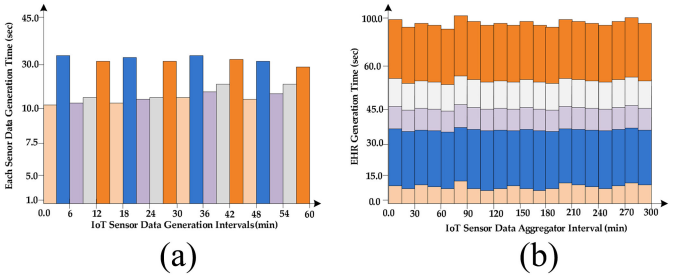

Fig. 12. IoT Elements activity. (a) IoT sensor data generation. (b) IoT aggregator EHR generation.

G. Discussions

This study proposes a novel privacy-preserving scheme that incorporates the blockchain-IoT ecosystem using the swarm exchange paradigm for secure EHR transmission in healthcare systems. The development of the proposed privacy-preserving system explained in Section IV could be implemented real-world EHR systems in IoT healthcare networks for enhancing security and privacy levels of the systems [40].

The key security and privacy features included in the proposed scheme are explained as follows.

1) Protection of data privacy of patients—patients' data in the proposed aspect are highly protected from the data breach and leakage while transferring EHR data to doctors across IoT healthcare networks. Using a hybridized encryption–decryption mechanism has significantly improved the data privacy of patients' data sources.
2) Protection of EHR against fraud—the immutability of EHR data using the blockchain and swarm exchange techniques is a promising feature that enhances CIA triads and privacy services of IoT healthcare systems. The Proposed scheme leverages a stringent scenario of protecting EHR data within the blockchain-IoT infrastructure against fraudulent behaviors that attempt to illegally modify or steal sensitive data of EHR files.
3) Transparency and security—blocks are usually shared with other nodes through the IoT healthcare network, this allows the proposed system to be transparent and secure. A group of trusted and secured nodes, i.e., swarm nodes act as the master of the system to access any service within nodes. The transparency and security aspects are leveraged in the proposed scheme for handling EHR data using blockchain and encryption mechanisms.
4) Interoperability of EHR data formats—the proposed swarm exchange paradigm can deal with various data types during transactions and transmissions over IoT networks. Textual and image EHR data files are effectively handled by the proposed privacy-preserving scheme to test the interoperability concern. The proposed scheme successfully can handle the interoperability aspect highly heterogeneous IoT healthcare environment [50].
5) Access control of data exchange—the proposed swarm exchange paradigm enables the blockchain-IoT ecosystem to control access under the private orientation of EHR data transmissions. Being a private blockchain, the deployed system model succeeds to provide regulated control of blockchain EHR data access in a seamless and secure manner.
6) Pseudonymity—although the proposed scheme is a private-blockchain to assert the privacy issues of HER files, it is not completely anonymous. It was intentionally levied on the proposed system to mitigate the pseudonymity of patients and doctors in situations, where some sort of anonymity is required to be held [51].
7) Full decentralization—the proposed privacy-preserving system is completely decentralized, whereby it facilitates the pervasiveness of underlying IoT protocols and lightweight services to the patients while transmitting their EHR files.
8) High availability—since the proposed privacy-preserving system is based on a private blockchain and hybridized cryptographic-wise P2P swarm nodes, the system becomes highly available to the patients and doctors in IoT healthcare networks. Even if some nodes leave the network or become inaccessible, the proposed system can support the network to continue its operations, making it highly available. The availability of the EHR data to all the swarm nodes in almost real-time is fully observed while implementing the proposed system.
9) Simplification of current paradigms—using the swarm exchange handler greatly supports the proposed system to be simplified over existing IoT healthcare systems. The utilization of swarm exchange significantly enriches the simplification process of EHR data flow within the proposed system.
10) Low cost—as no trusted third party is required in the proposed system; the swarm exchange paradigm can massively eliminate overhead costs in the form of the fees which are paid to such parties. We assume that upon direct implementation of the swarm exchange scheme, the proposed blockchain-IoT system could be useful to reduce the cost involved in the regular EHR data handling.
11) IoT sensor nodes with blockchain—in this study, we simulate IoT-based sensor nodes to generate body temperature, pulse rate, SPO2, GSR, and blood glucose values of a patient who is connected to the proposed system. Besides, such IoT sensors, add-on patient health data in form of text and images have been merged with the EHR for processing inside the blockchain ecosystem [48], [49].
12) IoT data aggregator—herein this work, we envisage an IoT data aggregator module that concatenates five different IOT sensor data in form of string of EHR. Such EHR is later on optionally augmented with other necessary details about the patient ready for blocked within the private chain and swarm platform.
13) IoT sensor heterogeneity—heterogeneity is one of the most crucial aspect of IoT ecosystem. Thus, in this article we involve five heterogeneous IoT-based health

sensor nodes simulated to harness the effectiveness of EHR conglomeration in a private blockchain-assisted swarm infrastructure. This work justifies the significance of heterogeneity of IoT-based sensor nodes for possible use in the proposed framework.

The swarm loading time in the uploading phase of the proposed system was found 64% average growth in term of increase of file size. .docx file took more time than .txt file around 93%, whereas .pdf uploading time was 100% more than .txt EHR file. Image EHR files also reflected similar results. It is noticed that the lowest upload and download time for .jpg EHR files, whereas, .tiff took the maximum and .png EHR files were in mid times of all EHR file types. In the swarm exchange phase, there was moderate steep growth of exchange time during the uploading phase, but not as sharp during the download stage.

It is also observed that small EHR files took less swarm exchange time du47ring upload and EHR files ranging from 3.3 to 4.6 MB that took minimal time during uploading. In the swarm listening experiments, very steep growth in listening time against the increase of EHR files. It is found that a slight sinusoidal growth of EHR files while decreasing the average EHR file size. Swarm announcement showed a gradual slow tangent of swarm announcement time in the uploading stage, but a random behavior was noticed in downloading duration. Swarm availability was surprisingly constant during the uploading time. A moderate change was noticed in the downloading duration of the files.

VI. Conclusion

This study introduced a privacy-preserving scheme based on blockchain and swarm exchange paradigms for protecting IoT healthcare, i.e., BIoTHR. The proposed scheme offers a dual-layer blockchain-IoT system that governs, monitors and controls the flow of EHRs between patients and doctors using a hybridized encryption/decryption mechanism. A swarm exchange paradigm using blockchain technology is used to enhance the EHR facilitation without the need of the third party while pouring high availability, flexibility, transparency, privacy, and security into the proposed scheme. The results showed promising performance of the proposed scheme in various aspects of blockchain-IoT scenarios, swarm exchange, and EHR transmission compared with several research studies. This suggested the reliability of the scheme in preserving and securing EHRs in IoT healthcare networks. In the future, the functions of the proposed scheme will be integrated with other security systems such as anomaly detection and statistical monitoring of data for providing a solid line of defense in IoT healthcare networks.

Since blockchain is a relatively new technology, we face some issues getting used to working on it, for example the blockchain concept is not widespread yet and there are only a few successful initiatives based on this technology at this time. That is a major hurdle because we do not have many successful blockchain models to follow which creates an uncertain situation. Even when going through the repositories of the Github there were not many projects which fully followed the concept of blockchain. Also, the setting up of different environment for different applications such as the docker and webjs was an issue and compatibility was missing for the programs written in different versions of the solidity. Though the overall work seems to be successful in running over a local environment it is still has not been successfully deployed over a larger network in which more work needs to be done in the future [50]. Resource trading [51] can be investigated along with the industrial IoT to deliver EHR messages as resource for business ecosystem development.

Though the work is a success, it is far from perfect. The blockchain created is for small time use and is not feasible for a large-scale network. Also, the whole process is being run via the terminal so the patients may have a problem with working around with it therefore we need to add an interface for making work easier. Currently the block is mined instantaneously by the virtual nodes itself, which we later need to improve by adding miners to the picture. Therefore, there is still place for improvement and potential in this current part of the extended work. We aim at integrating more lightweight IoT devices, protocols and platforms with the proposed system in future.

References

[1] 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 Trans. Consum. Electron.*, vol. 65, no. 3, pp. 388–397, Aug. 2019.

[2] D. Puthal and S. P. Mohanty, "Proof of authentication: IoT-friendly blockchains," *IEEE Potentials Mag.*, vol. 38, no. 1, pp. 26–29, Jan./Feb. 2019.

[3] P. Mishra, D. Puthal, M. Tiwary, and S. P. Mohanty, "Software defined IoT systems: Properties, state of the art, and future research," *IEEE Wireless Commun. Mag.*, vol. 26, no. 6, pp. 64–71, Dec. 2019.

[4] V. Gatteschi, F. Lamberti, C. Demartini, C. Pranteda, and C. Santamaría, "To blockchain or not to blockchain: That is the question," *IT Prof.*, vol. 20, no. 2, pp. 62–74, Mar./Apr. 2018.

[5] S. S. Roy, D. Puthal, S. Sharma, S. P. Mohanty, and A. Y. Zomaya, "Building a sustainable Internet of Things: Energy-efficient routing using low-power sensors will meet the need," *IEEE Consum. Electron. Mag.*, vol. 7, no. 2, pp. 42–49, Mar. 2018.

[6] M. A. Sayeed, S. P. Mohanty, E. Kougianos, and H. Zaveri, "A fast and accurate approach for real-time seizure detection in the IoMT," in *Proc. 4th IEEE Int. Smart Cities Conf. (ISC2)*, Kansas City, MO, USA, 2018, pp. 1–5.

[7] S. Wang, L. Ouyang, Y. Yuan, X. Ni, X. Han, and F.-Y. Wang, "Blockchain-enabled smart contracts: Architecture, applications, and future trends," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 11, pp. 2266–2277, Nov. 2019, doi: 10.1109/TSMC.2019.2895123.

[8] T. Salman, M. Zolanvari, A. Erbad, R. Jain, and M. Samaka, "Security services using blockchains: A state of the art survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 858–880, 1st Quart., 2019, doi: 10.1109/COMST.2018.2863956.

[9] R. Yang, F. R. Yu, P. Si, Z. Yang, and Y. Zhang, "Integrated blockchain and edge computing systems: A survey, some research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1508–1532, 2nd Quart., 2019, doi: 10.1109/COMST.2019.2894727.

[10] T. T. A. Dinh, R. Liu, M. Zhang, G. Chen, B. C. Ooi, and J. Wang, "Untangling blockchain: A data processing view of blockchain systems," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 7, pp. 1366–1385, Jul. 2018, doi: 10.1109/TKDE.2017.2781227.

[11] Z. Yu, X. Liu, and G. Wang, "A survey of consensus and incentive mechanism in blockchain derived from P2P," in *Proc. IEEE 24th Int. Conf. Parallel Distrib. Syst. (ICPADS)*, Singapore, 2018, pp. 1010–1015, doi: 10.1109/PADSW.2018.8645047.

[12] W. Yang, E. Aghasian, S. Garg, D. Herbert, L. Disiuta, and B. Kang, "A survey on blockchain-based Internet service architecture: Requirements, challenges, trends, and future," *IEEE Access*, vol. 7, pp. 75845–75872, 2019, doi: 10.1109/ACCESS.2019.2917562.

[13] L. Mertz, "(Block) chain reaction: A blockchain revolution sweeps into health care, offering the possibility for a much-needed data solution," *IEEE Pulse*, vol. 9, no. 3, pp. 4–7, May/Jun. 2018, doi: 10.1109/MPUL.2018.2814879.

[14] J. Al-Jaroodi and N. Mohamed, "Blockchain in industries: A survey," *IEEE Access*, vol. 7, pp. 36500–36515, 2019, doi: 10.1109/ACCESS.2019.2903554.

[15] J. Xie *et al.*, "A survey of blockchain technology applied to smart cities: Research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2794–2830, 3rd Quart., 2019, doi: 10.1109/COMST.2019.2899617.

[16] J. Liu and Z. Liu, "A survey on security verification of blockchain smart contracts," *IEEE Access*, vol. 7, pp. 77894–77904, 2019, doi: 10.1109/ACCESS.2019.2921624.

[17] *Swarm Architecture*. Accessed: Jul. 5, 2019. [Online]. Available: https://github.com/ipfs/specs/ tree/master/architecture

[18] R. N. Nortey, L. Yue, P. R. Agdedanu, and M. Adjeisah, "Privacy module for distributed electronic health records(EHRs) using the blockchain," in *Proc. IEEE 4th Int. Conf. Big Data Anal. (ICBDA)*, Suzhou, China, 2019, pp. 369–374, doi: 10.1109/ICBDA.2019.8713188.

[19] A. Cirstea, F. M. Enescu, N. Bizon, C. Stirbu, and V. M. Ionescu, "Blockchain technology applied in health the study of blockchain application in the health system (II)," in *Proc. 10th Int. Conf. Electron. Comput. Artif. Intell. (ECAI)*, Iasi, Romania, 2018, pp. 1–4, doi: 10.1109/ECAI.2018.8679029.

[20] G. Yang and C. Li, "A design of blockchain-based architecture for the security of electronic health record (EHR) systems," in *Proc. IEEE Int. Conf. Cloud Comput. Technol. Sci. (CloudCom)*, Nicosia, Cyprus, 2018, pp. 261–265, doi: 10.1109/CloudCom2018.2018.00058.

[21] G. Magyar, "Blockchain: Solving the privacy and research availability tradeoff for EHR data: A new disruptive technology in health data management," in *Proc. IEEE 30th Neumann Colloquium (NC)*, Budapest, Hungary, 2017, pp. 000135–000140, doi: 10.1109/NC.2017.8263269.

[22] 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. 66792–66806, 2019, doi: 10.1109/ACCESS.2019.2917555.

[23] J. Vora *et al.*, "BHEEM: A blockchain-based framework for securing electronic health records," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Abu Dhabi, UAE, 2018, pp. 1–6, doi: 10.1109/GLOCOMW.2018.8644088.

[24] F. Tang, S. Ma, Y. Xiang, and C. Lin, "An efficient authentication scheme for blockchain-based electronic health records," *IEEE Access*, vol. 7, pp. 41678–41689, 2019, doi: 10.1109/ACCESS.2019.2904300.

[25] R. Guo, H. Shi, D. Zheng, C. Jing, C. Zhuang, and Z. Wang, "Flexible and efficient blockchain-based ABE scheme with multi-authority for medical on demand in telemedicine system," *IEEE Access*, vol. 7, pp. 88012–88025, 2019, doi: 10.1109/ACCESS.2019.2925625.

[26] R. Guo, H. Shi, Q. Zhao, and D. Zheng, "Secure attribute-based signature scheme with multiple authorities for blockchain in electronic health records systems," *IEEE Access*, vol. 6, pp. 11676–11686, 2018, doi: 10.1109/ACCESS.2018.2801266.

[27] I. Kotsiuba, A. Velvkzhanin, Y. Yanovich, I. S. Bandurova, Y. Dyachenko, and V. Zhygulin, "Decentralized e-Health architecture for boosting healthcare analytics," in *Proc. 2nd World Conf. Smart Trends Syst. Security Sustain. (WorldS4)*, London, U.K., 2018, pp. 113–118, doi: 10.1109/WorldS4.2018.8611621.

[28] A. Ekİn and D. Ünay, "Blockchain applications in healthcare," in *Proc. 26th Signal Process. Commun. Appl. Conf. (SIU)*, Izmir, Turkey, 2018, pp. 1–4, doi: 10.1109/SIU.2018.8404275.

[29] N. Kshetri, "Blockchain and electronic healthcare records [cybertrust]," *Computer*, vol. 51, no. 12, pp. 59–63, Dec. 2018, doi: 10.1109/MC.2018.2880021.

[30] T. Mikula and R. H. Jacobsen, "Identity and access management with blockchain in electronic healthcare records," in *Proc. 21st Euromicro Conf. Digit. Syst. Design (DSD)*, Prague, Czech Republic, 2018, pp. 699–706, doi: 10.1109/DSD.2018.00008.

[31] J. Liu, X. Li, L. Ye, H. Zhang, X. Du, and M. Guizani, "BPDS: A blockchain based privacy-preserving data sharing for electronic medical records," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Abu Dhabi, UAE, 2018, pp. 1–6, doi: 10.1109/GLOCOM.2018.8647713.

[32] C. Esposito, A. De Santis, G. Tortora, H. Chang, and K.-K. R. Choo, "Blockchain: A panacea for healthcare cloud-based data security and privacy?" *IEEE Cloud Comput.*, vol. 5, no. 1, pp. 31–37, Jan./Feb. 2018, doi: 10.1109/MCC.2018.011791712.

[33] A. R. Rajput, Q. Li, M. T. Ahvanooey, and I. Masood, "EACMS: Emergency access control management system for personal health record based on blockchain," *IEEE Access*, vol. 7, pp. 84304–84317, 2019, doi: 10.1109/ACCESS.2019.2917976.

[34] S. Alexaki, G. Alexandris, V. Katos, and E. N. Petroulakis, "Blockchain-based electronic patient records for regulated circular healthcare jurisdictions," in *Proc. IEEE 23rd Int. Workshop Comput.-Aided Model. Design Commun. Links Netw. (CAMAD)*, Barcelona, Spain, 2018, pp. 1–6, doi: 10.1109/CAMAD.2018.8514954.

[35] A. K. Talukder, M. Chaitanya, D. Arnold, and K. Sakurai, "Proof of disease: A blockchain consensus protocol for accurate medical decisions and reducing the disease burden," in *Proc. IEEE SmartWorld Ubiquitous Intell. Comput. Adv. Trusted Comput. Scalable Comput. Commun. Cloud Big Data Comput Internet People Smart City Innovat. (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, Guangzhou, China, 2018, pp. 257–262, doi: 10.1109/SmartWorld.2018.00079.

[36] K. Azbeg, O. Ouchetto, S. J. Andaloussi, L. Fetjah, and A. Sekkaki, "Blockchain and IoT for security and privacy: A platform for diabetes self-management," in *Proc. 4th Int. Conf. Cloud Comput. Technol. Appl. (Cloudtech)*, Brussels, Belgium, 2018, pp. 1–5, doi: 10.1109/CloudTech.2018.8713343.

[37] *The GNU Privacy Guard*. Accessed: Jul. 1, 2019. [Online]. Available: https://gnupg.org/

[38] *IPFS is the Distributed Web*. Accessed: Jun. 19, 2019. [Online]. Available: https://ipfs.io/

[39] *The Go Programming Language*. Accessed: Jun. 28, 2019. [Online]. Available: https://golang.org/

[40] J. Xu *et al.*, "Healthchain: A blockchain-based privacy preserving scheme for large-scale health data," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8770–8781, Oct. 2019, doi: 10.1109/JIOT.2019.2923525.

[41] M. Keshk, E. Sitnikova, N. Moustafa, J. Hu, and I. Khalil, "An integrated framework for privacy-preserving based anomaly detection for cyber-physical systems," *IEEE Trans. Sustain. Comput.*, early access, Mar. 25, 2019, doi: 10.1109/TSUSC.2019.2906657.

[42] M. Keshk, N. Moustafa, E. Sitnikova, and B. Turnbull, "Privacy-preserving big data analytics for cyber-physical systems," *Wireless Netw.*, pp. 1–9, Dec. 2018. [Online]. Available: https://doi.org/10.1007/s11276-018-01912-5

[43] M. Keshk, N. Moustafa, E. Sitnikova, and G. Creech, "Privacy preservation intrusion detection technique for SCADA systems," in *Proc. IEEE Military Commun. Inf. Syst. Conf. (MilCIS)*, Canberra, ACT, Australia, 2017, pp. 1–6.

[44] N. Moustafa, B. Turnbull, and K.-K. R. Choo, "An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4815–4830, Jun. 2019.

[45] J. Li, Y. Ji, K.-K. R. Choo, and D. Hogrefe, "CL-CPPA: Certificate-less conditional privacy-preserving authentication protocol for the Internet of Vehicles," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10332–10343, Dec. 2019.

[46] K. R. Ozyilmaz and A. Yurdakul, "Designing a blockchain-based IoT with Ethereum, swarm, and LoRa: The software solution to create high availability with minimal security risks," *IEEE Consum. Electron. Mag.*, vol. 8, no. 2, pp. 28–34, Mar. 2019.

[47] A. Islam and S. Y. Shin, "Bus: A blockchain-enabled data acquisition scheme with the assistance of UAV swarm in Internet of Things," *IEEE Access*, vol. 7, pp. 103231–103249, 2019.

[48] P. P. Ray, D. Dash, and D. De, "Internet of Things-based real-time model study on e-Healthcare: Device, message service and dew computing," *Comput. Netw.*, vol. 149, pp. 226–239, Feb. 2019.

[49] A. Chaer, K. Salah, C. Lima, P. P. Ray, and T. Sheltami, "Blockchain for 5G: Opportunities and challenges," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Waikoloa, HI, USA, 2019, pp. 1–6, doi: 10.1109/GCWkshps45667.2019.9024627.

[50] S. Tuli *et al.*, "Next generation technologies for smart healthcare: Challenges, vision, model, trends and future directions," *Internet Technol. Lett.*, vol. 3, no. 2, p. e145, 2020. [Online]. Available: https://doi.org/10.1002/itl2.145

[51] H. Yao, T. Mai, J. Wang, Z. Ji, C. Jiang, and Y. Qian, "Resource trading in blockchain-based industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 15, no. 6, pp. 3602–3609, Jun. 2019, doi: 10.1109/TII.2019.2902563.

[52] R. Jabbar, N. Fetais, M. Krichen, and K. Barkaoui, "Blockchain technology for healthcare: Enhancing shared electronic health record interoperability and integrity," in *Proc. IEEE Int. Conf. Inf. IoT Enabling Technol. (ICIoT)*, Doha, Qatar, 2020, pp. 310–317, doi: 10.1109/ICIoT48696.2020.9089570.

[53] O. Ajayi, M. Abouali, and T. Saadawi, "Secure architecture for inter-healthcare electronic health records exchange," in *Proc. IEEE Int. IOT Electron. Mechatronics Conf. (IEMTRONICS)*, Vancouver, BC, Canada, 2020, pp. 1–6, doi: 10.1109/IEMTRONICS51293.2020.9216336.

[54] E. Saweros and Y.-T. Song, "Connecting personal health records together with EHR using tangle," in *Proc. 20th IEEE/ACIS Int. Conf. Softw. Eng. Artif. Intell. Netw. Parallel/Distrib. Comput. (SNPD)*, Toyama, Japan, 2019, pp. 547–554, doi: 10.1109/SNPD.2019.8935646.

[55] R. Bosri, A. R. Uzzal, A. Al Omar, M. Z. A. Bhuiyan, and M. S. Rahman, "HIDEchain: A user-centric secure edge computing architecture for healthcare IoT devices," in *Proc. IEEE INFOCOM Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Toronto, ON, Canada, 2020, pp. 376–381, doi: 10.1109/INFOCOMWKSHPS50562.2020.9162729.

[56] R. M. Patil and R. Kulkarni, "Universal storage and analytical framework of health records using blockchain data from wearable data devices," in *Proc. 2nd Int. Conf. Innovat. Mech. Ind. Appl. (ICIMIA)*, Bangalore, India, 2020, pp. 311–317, doi: 10.1109/ICIMIA48430.2020.9074909.

[57] M. Al Baqari and E. Barka, "Biometric-based blockchain EHR system (BBEHR)," in *Proc. Int. Wireless Commun. Mobile Comput. (IWCMC)*, Limassol, Cyprus, 2020, pp. 2228–2234, doi: 10.1109/IWCMC48107.2020.9148357.

Partha Pratim Ray (Senior Member, IEEE) received the B.Tech. degree in computer science and engineering and the M.Tech. degree in electronics and communication engineering, with specialization in embedded systems, from the West Bengal University of Technology, Kolkata, India, in 2008 and 2011, respectively.

He is currently a full-time Assistant Professor with the Department of Computer Applications, Sikkim University, Gangtok, India. He has published papers in IEEE ACCESS, IEEE SYSTEMS JOURNAL, IEEE TRANSACTIONS ON CONSUMER ELECTRONICS, IEEE/ACM TRANSACTIONS ON COMPUTATIONAL BIOLOGY AND BIOINFORMATICS, *Computer Networks*, *Journal of Network and Computer Applications* and IEEE GLOBECOM 2019. His research interests include Internet of Things, dew computing, blockchain and pervasive biomedical informatics.

Dr. Ray has been awarded with Young Engineers Award by the Institution of Engineers India from 2019 to 2020. He is guest editor in the special issues of the IEEE INTERNET OF THINGS JOURNAL, Elsevier, *Sensors* (MDPI).

Biky Chowhan (Member, IEEE) received the B.C.A. degree from the Sikkim Manipal Institute of Technology, Gangtok, India, in 2016, and the M.C.A. degree from Sikkim University, Gangtok, in 2019.

He is currently with the National Institute of Electronics and Information Technology, Gangtok. His area of interest includes IoT and blockchain. He has contributed in the research and training of knowledge dissemination related to Raspberry Pi, Arduino in alignment with the IoT. He has also provided training on the Python.

Neeraj Kumar (Senior Member, IEEE) received the M.Sc., M.Tech. degrees and the Ph.D. degree in CSE from Shri Mata Vaishno Devi University, Katra, India, in 1999, 2000, and 2009, respectively.

He was a Postdoctoral Research Fellow with Coventry University, Coventry, U.K. He is currently a Full Professor with the Department of Computer Science and Engineering, Thapar University, Patiala, India. He has published more than 400 technical research papers in leading journals and conferences in top cited journals, such as the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE TRANSACTIONS ON CLOUD COMPUTING, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE TRANSACTIONS ON CONSUMER ELECTRONICS, IEEE NETWORK, *IEEE Communications Magazine*, IEEE WIRELESS COMMUNICATIONS LETTERS, IEEE INTERNET OF THINGS JOURNAL, IEEE SYSTEMS JOURNAL, *Future Generation Computing Systems*, *Journal of Network and Computer Applications*, and ComCom.

Ahmad Almogren (Senior Member, IEEE) received the Ph.D. degree in computer science from Southern Methodist University, Dallas, TX, USA, in 2002.

He is a Professor with the Computer Science Department, College of Computer and Information Sciences (CCIS), King Saud University (KSU), Riyadh, Saudi Arabia, where he is currently the Director of Cyber Security Chair. He worked as the Vice Dean for the Development and Quality with CCIS. He also served as the Dean for the College of Computer and Information Sciences and the Head of Academic Accreditation Council with Al Yamamah University, Riyadh. His research areas of interests include mobile-pervasive computing and cyber security.

Prof. Almogren served as the General Chair for the IEEE Smart World Symposium and a Technical Program Committee member in numerous international conferences/workshops, such as IEEE CCNC, ACM BodyNets, and IEEE HPCC.