Journal of Network and Computer Applications 217 (2023) 103696

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

A novel secure and distributed architecture for privacy-preserving healthcare system☆

Rakib Ul Haque [a], A.S.M. Touhidul Hasan [b,c,*], Apubra Daria [c], Abdur Rasool [d], Hui Chen [e,f], Qingshan Jiang [d], Yuqing Zhang [g,**]

[a] School of Computer Science & Technology, University of Chinese Academy of Sciences, Beijing 100049, China
[b] Division of Computing, Analytics and Mathematics, School of Science and Engineering, University of Missouri-Kansas City, Kansas City, MO 64110, United States of America
[c] Institute of Automation Research and Engineering, Dhaka 1205, Bangladesh
[d] Shenzhen Key Laboratory for High Performance Data Mining, Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China
[e] School of Computer Engineering, Shenzhen Polytechnic, Shenzhen 518055, Guangdong, China
[f] Institute of Applied Mathematics, Hebei Academy of Sciences, Hebei 050081, China
[g] National Computer Network Intrusion Protection Center, University of Chinese Academy of Sciences, Beijing 100049, China

ARTICLE INFO

Keywords:
Health monitoring system
Secure data sharing
Local differential privacy
Machine learning
Blockchain

ABSTRACT

Patients often visit several hospitals to obtain medication, generating a significant volume of data. Moreover, hospitals use different data analytic techniques to improve healthcare services, leading to patient data privacy. However, no integrated architecture has a trustworthy data repository and secure communication protocols to store and share the data with various parties (e.g., hospitals and data analysts) for different healthcare services. This study proposes an innovative three-tier secure and distributed privacy-preserving healthcare architecture that addresses the aforementioned challenges. The first tier introduces a trustworthy data repository called custodian, where the data owner stores encrypted data and offer real-time privacy-preserving health monitoring service. The second tier provides Elliptic Curve Cryptography-based authentication for secure data exchange between the data owner and the hospital. The third tier utilizes smart contracts and local differential privacy (LDP) for secure machine learning model training. All transactions are recorded on the blockchain and managed by a smart contract. Security analysis shows that the proposed framework ensures privacy, security, and data integrity. Performance analysis is done based on score metrics, scalability metrics, and formal analysis. A transaction takes 0.00121 s in the first tier, and in the second tier, it takes 0.1267 s. The third tier uses $\epsilon - LDP$ with a privacy budget of $\epsilon = 3$ on Random Forest and achieves 97%, 83%, and 74% accuracy on breast cancer, heart disease, and diabetic datasets. This accuracy is higher than the previous state-of-the-art methods. Moreover, the proposed healthcare framework ensures privacy, security and outperforms the previous state-of-the-art methods.

1. Introduction

The Internet of Things (IoT) refers to the interconnection of physical devices, vehicles, buildings, and other items embedded with electronics, software, sensors, and network connectivity that enable these objects to collect and exchange data (Laghari et al., 2021). The application of IoT in healthcare, known as the Internet of Medical Things, has the potential to revolutionize patient care, enhance clinical outcomes, and reduce costs (Rahman and Hossain, 2021). The IoT enables remote patient monitoring, telemedicine, and real-time tracking of medical devices and supplies, among other applications. For instance, wearable devices such as fitness trackers and smartwatches can collect data on physical activity, heart rate, sleep patterns, and other health metrics that can be used to personalize treatment plans and

☆ The work was supported by The National Key Research and Development Program of China under Grant: 2021YFF1200100, 2021YFF1200104 and Hebei Academy of Sciences under Grant: 22602.

* Corresponding author at: Division of Computing, Analytics and Mathematics, School of Science and Engineering, University of Missouri-Kansas City, Kansas City, MO 64110, United States of America.
** Corresponding author.
E-mail addresses: rakibulhaqueraj@mails.ucas.ac.cn (R.U. Haque), ahrzd@umkc.edu (A.S.M.T. Hasan), apubra@iar-e.com (A. Daria), rasool@siat.ac.cn (A. Rasool), hui.chen1@siat.ac.cn (H. Chen), qs.jiang@siat.ac.cn (Q. Jiang), zhangyq@ucas.ac.cn (Y. Zhang).

https://doi.org/10.1016/j.jnca.2023.103696
Received 15 October 2022; Received in revised form 18 May 2023; Accepted 28 June 2023
Available online 4 July 2023

improve patient engagement. Similarly, medical devices such as blood glucose monitors, pacemakers, and insulin pumps can transmit real-time data to healthcare providers, enabling them to monitor patient health based on machine learning (ML) and respond to changes in conditions promptly (Ayvaz and Alpay, 2021). However, the collection and transmission of sensitive health data through IoT devices raise concerns about privacy and security (Paul et al., 2023). The healthcare industry is subject to strict regulations, such as the Health Insurance Portability and Accountability Act, which mandate the protection of patient information (Anderson et al., 2023). IoT devices that collect, store, or transmit health data must comply with these regulations and ensure that patients' privacy and security are protected. Healthcare organizations and IoT device manufacturers must implement appropriate security measures to safeguard patient data from unauthorized access, theft, or alteration. These measures include data encryption, access control, secure authentication, and regular security audits. Some of these measures can be ensured with blockchain technology.

Blockchain and smart contract technology are increasingly being recognized as potential solutions for enhancing privacy and security in IoT healthcare data (Issa et al., 2023). Blockchain is a decentralized and distributed ledger technology that provides a secure and tamper-proof mechanism for recording and sharing data. On the other hand, smart contracts are self-executing digital contracts that automate the execution of contract terms and conditions. The use of blockchain in IoT healthcare data offers several benefits, including enhanced security, transparency, and data integrity. By using a distributed ledger system, blockchain eliminates the need for a central authority to manage the data, thereby reducing the risk of data breaches and unauthorized access (Pasdar et al., 2023). Moreover, the blockchain's immutable nature ensures that once data is recorded on the blockchain, it cannot be altered, providing a high level of data integrity and reliability. Smart contract-based blockchains offer an added layer of security by automating the execution of contract terms and conditions. Regardless, blockchain cannot cover all the security parameters required for the entire healthcare system.

This study analyzes the healthcare system from the patient/data owner ($\mathcal{O}$) perspective to determine all the privacy and security requirements. Firstly, $\mathcal{O}$ holds a massive amount of data generated from its IoT devices. This dataset also consists of medication from doctors. $\mathcal{O}$ needs to store this dataset in any medical cloud storage, which is costly and failed to ensure privacy. Moreover, $\mathcal{O}$ does not have any trustworthy repository to store its dataset. For that reason, $\mathcal{O}$ faces several unnecessary difficulties such as gathering hard copies, and maintaining several formalities for collecting past health data from hospitals $\mathcal{H}$ and claiming insurance from the insurance company. Secondly, numerous devices of $\mathcal{O}$ are often connected to the trusted third-party (TTP) server for sharing patients' sensitive information with the doctors from any $\mathcal{H}$. Here, TTP utilization is also costly and risky. Further, the design methodology for sharing data is complicated and lacks authorization. Moreover, patient data have privacy and security threats because data are in plain text format. Thirdly, whenever a data analyst $\mathcal{A}$ wants to train any model, there is a risk of data privacy breach for the $\mathcal{O}$. These issues of health care require proper solutions.

Instead of handling the above situations separately, this study combines popular technologies, lightweight public-key cryptography (i.e., elliptic curve cryptography ECC), ML, and differential privacy (DP) to propose a secure privacy-preserving three-tier framework for the Health care system. In $Tier$-1 of the proposed system, $\mathcal{O}$ is the data owner, and all data has been stored in the data custodian $\mathcal{C}$. $\mathcal{C}$ is a decentralized data repository, such as the interplanetary file systems ($IPFS$). The $\mathcal{C}$ will get a small amount of transaction fee every time the $\mathcal{O}$ wants to add new data in the $\mathcal{C}$, and it will not have any control over the data. $Tier$-2 focuses on the lightweight authentication of $\mathcal{O}$ and $\mathcal{H}$, where blockchain and ECC are employed for data security. $Tier$-3 focuses on $\mathcal{A}$ who wants to train their ML model with the data of various $\mathcal{O}$. This process has privacy issues. Therefore, this study employs smart-contract-based ML with differential privacy for privacy-preserving model training. Here, researchers will never perceive the sensitive data of the participants. As all the transactions are registered in the blockchain, in the future, no parties can deny any transaction. These recorded transactions in the blockchain also work as evidence for claiming health insurance from the insurance company.

The main contributions are as follows:

- The concept of a data custodian is proposed in $Tier$-1, which will act as a trustworthy data repository and decentralized storage for patients. All patients must register with the data custodian and authenticate before storing their encrypted data. $Tier$-1 will also provide real-time health monitoring services.
- The $Tier$-2 system incorporates an integrated authentication and secure data-sharing mechanism for patients and hospitals. Patients will grant the hospital view access to their encrypted data held by the respective data custodian, using a session key established through symmetric key cryptography. Based on the health record, the doctor at the hospital will administer medication, with the resulting data being added to the data custodian of the corresponding data owner.
- In $Tier$-3, a novel approach is introduced for privacy-preserving machine learning, leveraging smart contact-based technology and local differential privacy. Once authentication is complete, patients and data analysts can engage in secure interactions. Notably, the training of the model does not require the involvement of a trusted third party.

The structure of the paper is outlined as follows: In Sections 2–4, the related works, preliminaries, and system overview are presented, respectively. Section 5 covers the model construction, while Section 6 focuses on performance evaluation. Lastly, the paper concludes in Section 7.

2. Related work

This section provides a $Tier$-wise analysis of previous work. An overview of existing research on privacy-preserving healthcare systems is presented in Table 1. It categorizes the studies based on tiers. It also identifies the technologies, and cyber attacks covered by the previous studies.

Several studies have addressed the issue of data storage in healthcare systems (Miyachi and Mackey, 2021; Deepa and Pandiaraja, 2021; Ghayvat et al., 2021). These studies have explored various aspects such as the structure of off-chain systems for healthcare information management (Miyachi and Mackey, 2021), privacy-preserving report retrieval (Deepa and Pandiaraja, 2021), and access control for healthcare medical reports (Ghayvat et al., 2021). However, none of these studies have adequately addressed the lack of a concrete data repository that provides patients with full control over their data or covers all types of data. To overcome these challenges, $Tier$-1 introduces the concept of $\mathcal{C}$, which leverages the idea of $IPFS$. This solution ensures decentralized and off-chain storage of all data, while also incorporating a real-time health monitoring system.

Additionally, there are studies that specifically address privacy and security concerns associated with data transfer (Wu et al., 2017; He et al., 2015; Merabet et al., 2020). Wu et al. (2017) developed a hybrid protection system that combines symmetric and asymmetric key cryptographic methods. The system focuses on maintaining the anonymity attribute while lacking support for progressive controller junction computing and pharmaceutical equipment extension. He et al. (2015) proposed various authentication protocols for an Ambient Assisted Living system utilizing ambient intelligence. These protocols facilitate the observation of medical information and the implementation of telemedical care assistance. Merabet et al. (2020) examined the Machine-to-Cloud and Machine-to-Machine communication methods necessary in healthcare, specifically in IoT applications. However,

Table 1
An overview of existing research on privacy-preserving healthcare systems.

Study	Year	*Tier*-1	*Tier*-2	*Tier*-3	Technologies and security parameters covered														
					$\mathcal{V}_1$	$\mathcal{V}_2$	$\mathcal{V}_3$	$\mathcal{V}_4$	$\mathcal{V}_5$	$\mathcal{V}_6$	$\mathcal{V}_7$	$\mathcal{V}_8$	$\mathcal{V}_9$	$\mathcal{V}_{10}$	$\mathcal{V}_{11}$	$\mathcal{V}_{12}$	$\mathcal{V}_{13}$	$\mathcal{V}_{14}$	$\mathcal{V}_{15}$
He et al. (2015)	2015	N/A	✓	N/A	✓	✗	✗	✗	✗	✓	✓	✓	✓	✗	✗	✗	✗	✓	✓
Wu et al. (2017)	2017	N/A	✓	N/A	✗	✓	✗	✗	✗	✗	✓	✓	✓	✗	✓	✗	✗	✓	✗
Shen et al. (2019)	2019	N/A	N/A	✓	✗	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✗
Merabet et al. (2020)	2020	N/A	✓	N/A	✓	✗	✗	✓	✗	✗	✗	✓	✓	✗	✗	✗	✗	✓	✗
Khan et al. (2020)	2020	N/A	✓	N/A	✓	✗	✗	✓	✗	✗	✗	✓	✓	✗	✓	✗	✗	✓	✓
Liu et al. (2020)	2020	N/A	N/A	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗
Jia et al. (2020)	2020	N/A	N/A	✓	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓
Le Nguyen et al. (2020)	2020	N/A	N/A	✓	✓	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓
Miyachi and Mackey (2021)	2021	✓	N/A	N/A	✗	✗	✓	✓	✗	✓	✓	✓	✗	✗	✓	✗	✓	✗	✓
Deepa and Pandiaraja (2021)	2021	✓	N/A	N/A	✓	✗	✗	✗	✓	✓	✗	✗	✗	✗	✓	✗	✗	✗	✓
Zhu et al. (2021)	2021	N/A	N/A	✓	✗	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✗
Ghayvat et al. (2021)	2021	✓	N/A	N/A	✓	✓	✓	✓	✗	✗	✗	✓	✓	✗	✗	✗	✓	✗	✓
Zhang et al. (2022)	2022	N/A	N/A	✓	✗	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗
Akter et al. (2022)	2022	N/A	N/A	✓	✗	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✗
Tang et al. (2022)	2022	N/A	N/A	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓
Yu et al. (2022)	2022	N/A	N/A	✓	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✓
Zhao et al. (2023)	2023	N/A	N/A	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗
This study	2023	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Here, $\mathcal{V}_1$: lightweight data hiding; $\mathcal{V}_2$: heavyweight data hiding; $\mathcal{V}_3$: blockchain-based solution; $\mathcal{V}_4$: *IoT*-based solution; $\mathcal{V}_5$: replay attack; $\mathcal{V}_6$: man-in-the-middle attack; $\mathcal{V}_7$: impersonation attack; $\mathcal{V}_8$: mutual authentication; $\mathcal{V}_9$: key agreement; $\mathcal{V}_{10}$: formal security analysis; $\mathcal{V}_{11}$: eavesdropping attack; $\mathcal{V}_{12}$: data integrity; $\mathcal{V}_{13}$: authenticity; $\mathcal{V}_{14}$: confidentiality; $\mathcal{V}_{15}$: availability; ✓: "a scheme is secure or it supports an attribute"; ✗: "a scheme is insecure or it does not support an attribute"; N/A: not applicable in a scheme.

their proposed protocols do not support progressive controller node computing and pharmaceutical equipment extension. Khan et al. (2020) has focused on lightweight registration and authentication using ECC. Again, none of the aforementioned research addresses privacy concerns related to data sharing between patients and hospitals in the medical system. To address these issues, $Tier$-2 employs lightweight cryptography to generate a session key, ensuring secure connections and data transfer.

On the other hand, When multiple individuals are involved in training a ML model using IoT data, privacy concerns emerge. The objective is to protect the data privacy of each entity, preventing it from being accessed by other participants during the ML model training process. This topic has garnered considerable attention, and this study also focuses on it in $Tier$-3. By leveraging a Homomorphic Cryptosystem (HC) for encrypted data, ML training can offer stronger privacy guarantees. HC allows computations to be performed on ciphertexts while preserving the accuracy of the information. Several privacy-preserving techniques based on HC have been proposed for various training ML algorithms such as Support Vector Machine (SVM), k-means, and others (Shen et al., 2019; Zhang et al., 2022; Akter et al., 2022; Liu et al., 2020; Zhu et al., 2021). Here, authors develop secure protocols for performing mathematical operations securely, relying on partially HC like Paillier. However, HC-based solutions have their own limitations. Firstly, they are computationally expensive and limited to integer computations. Secondly, they exhibit higher time complexity, and thirdly, they require substantial memory resources. $Tier$-3 resolves the hurdles by utilizing smart contract-based local differential privacy LDP in ML algorithms. Existing studies on LDP based ML do not include smart contract based operation (Tang et al., 2022; Jia et al., 2020; Le Nguyen et al., 2020; Zhao et al., 2023; Yu et al., 2022).

Studies discussed above focused on sub-domains of the health sector, i.e., data sharing, intrusion detection, malware protection, etc. There is still an emptiness for a secure framework that can focus on all realms of healthcare. This study introduces a secure three-tier structure for the healthcare system based on blockchain. $Tier$-1 is based on $\mathcal{C}$s, which act as a trustworthy repository for $\mathcal{O}$s and it also provides real-time healthcare with the help of smart contracts. $Tier$-2 focuses on lightweight authentication and registration for secure data sharing between $\mathcal{O}$ and $\mathcal{H}$. In $Tier$-3, smart contract based LDP is applied in ML algorithms incorporated with blockchain for ensuring privacy. We used LDP as it is faster, and spends less time and space than HC.

3. Preliminaries

This section represents all preliminaries related to this study. Symbols illustrated in Table 2 will be used throughout the rest of the paper. $\mathcal{ID}$ stands for the identity of an entity such as patients/data owner, hospital and researcher, etc. $\mathcal{PK}$, $\mathcal{SK}$, $\mathcal{SEK}$ and $\mathcal{DS}$ represents the public key, private key, session key, and digital signature respectively. m stands for messages. There are various functions, which are used in this paper such as the encryption function $\mathcal{EN}(\ )$, the hash function $h(\ )$, and the decryption function $\mathcal{DN}(\ )$, etc. An unordered dataset D with the size s, where x_i, y_i is the ith record in D and l_i the output label. This research used ECC as the cryptographic system and LDP. $[[m]]$ is an encrypted message.

3.1. Elliptic curve cryptography (ECC)

Currently, the following two problems are used frequently in ECC. The first one is the Elliptic curve discrete logarithm problem (ECDLP). 10-bit ECDLP is employed in cryptosystems. Here, $\mathcal{AD}$ unable to compute u. For $\mathcal{Q}, \mathcal{P} \in \mathcal{E}(\mathcal{F}_p)$ is $u \in \mathcal{Z}_q^*$ and $\mathcal{Q} = u\mathcal{P}$. Again, the second one is the Elliptic curve computational Deffie–Hellmen problem (ECDH). 160-bit ECDLP is protected and still unsolvable by hackers. Therefore $\mathcal{AD}$ will not be able to compute $uv\mathcal{P}$ ($u, v \in \mathcal{Z}_q^*$ and $u\mathcal{P}, v\mathcal{P} \in \mathcal{E}(\mathcal{F}_p)$) (Wu et al., 2018).

3.2. Local differential privacy (LDP)

It is essential to have perturbation by randomization in order to assure privacy through credible deniability. The standard concept of differential privacy is defined in the Erlingsson et al. (2014). The definition of $\varepsilon - DP$ is defined in Definition 1.

Table 2
Notations.

Sign	Meanings	Sign	Meanings	Sign	Meanings	Sign	Meanings
IoT	internet of things	ML	machine learning	DP	differential privacy	ECC	elliptic curve cryptography
$\mathcal{MA}$	message	r, u, v	nonce	LDP	local DP	$IPFS$	interplanetary file systems
$\mathcal{O}$	owner/patent	$\mathcal{C}$	custodian	$\mathcal{H}$	hospital	TTP	trusted third-party
$\mathcal{A}$	analyst	$\mathcal{SP}$	service provider	$\mathcal{ID}$	identity	$\mathcal{DE}()$	decryption function
$\mathcal{PK}$	public key	$\mathcal{SK}$	private Key	p, q	two large primes	Φ	registration protocol
$\mathcal{AD}$	adversaries	$\mathcal{E}()$	elliptic curve	$\mathcal{F}_p$	finite field	Γ	authentication protocol
$\mathcal{Z}_q^*$	multiplicative group	$\mathcal{P}$	a generator	$h()$	hash function	$\mathcal{EN}()$	encryption function
Med	medication	$\mathcal{SEK}$	session key	DS	digital signature	HC	homomorphic cryptosystem

Definition 1. $\varepsilon- DP$ can be achieved by a randomized function K if all datasets D_1 and D_2 varying on at most unit component, and all $S \subseteq Range(K)$,

$$\frac{\mathcal{P}r[\mathcal{K}(D_1) \in S]}{\mathcal{P}r[\mathcal{K}(D_2) \in S]} \leq e^{\varepsilon} \tag{1}$$

The main issue with the centralized notion of $\varepsilon- DP$ is that it needs a reliable centralized authority such as the database maintainer in order to keep its privacy.

The notion of LDP does not need a reliable centralized authority (Erlingsson et al., 2014). The $\varepsilon - LDP$ is defined in Definition 2.

Definition 2. $\varepsilon- LDP$ can be achieved by algorithm π, where $\varepsilon > 0$ if for any input v and v'

$$\forall_y \in Range(\pi) : \frac{\mathcal{P}r[\pi(v) = y]}{\mathcal{P}r[\pi(v') = y]} \leq e^{\varepsilon} \tag{2}$$

here, $Range(\pi)$ indicates all potential outcomes of the algorithm π.

Formula for $\varepsilon - LDP$ is: $m' = m + \mathcal{LAP}(\frac{\delta}{\varepsilon})$, where, ε privacy budget, m plain text, function Laplace $\mathcal{LAP}(\,)$, δ predefined sensitivity, and m' noise data.

3.3. Blockchain and smart-contracts

A blockchain is a distributed ledger technology that enables secure and decentralized transactions or data storage. It consists of a chain of blocks, where each block contains a list of transactions or data records, along with a cryptographic hash that links it to the previous block (Nakamoto, 2008). The mathematical equation that represents the structure of a blockchain can be defined as $Blockchain = Block_1, Block_2, Block_3, \ldots, Block_n$, where $Block_1$, $Block_2$, ..., $Block_n$ represents individual blocks in the chain. Computer applications operating over the blockchain network can express business logic, statuses, and triggers to enable transactions, which are complex programs known as Smart contracts.

Smart contracts are self-executing contracts with the terms of the agreement directly written into code. They automatically execute predefined actions once specific conditions encoded in the contract are met. The mathematical equation for a smart contract can be represented as $SmartContract = Conditions, Actions$, where Conditions represent the predefined conditions that need to be satisfied, and Actions denote the actions to be performed once the conditions are met. This study employs Hyperledger as it is private and permissioned. Any transactions taking place on the network are only visible to the authorized members. Hyperledger is faster than Ethereum (Gorkhali et al., 2020).

3.4. Interplanetary file systems ($IPFS$)

InterPlanetary File System is a decentralized protocol and network designed to create a permanent and distributed method of storing and accessing files. It aims to replace the traditional client–server model of the web with a peer-to-peer network. $IPFS$ is employed for storing the documents (off-chain) in a decentralized way (Jayabalan and Jeyanthi, 2022). All documents would be too costly for on-chain storage. Consequently, decentralized and secure stowing of this content is compulsory. Public and distributed storage is $IPFS$. Therefore, any data stowed on $IPFS$ must be encrypted. Authorized individuals should be allowed to access the content. In this study custodian, $\mathcal{C}$ is acting as $IPFS$.

- Content-Addressable Storage: $IPFS$ uses content-addressable storage, where each piece of data is identified by its unique cryptographic hash.
- Distributed Hash Table (DHT): $IPFS$ utilizes a distributed hash table to store and retrieve content. A DHT is a decentralized key–value store that allows efficient lookup and retrieval of data across the network.
- MerkleDAG (Roy et al., 2022): $IPFS$ organizes data into a structure called a MerkleDAG (Directed Acyclic Graph). A MerkleDAG is a tree-like structure where each node represents a piece of content, and the edges represent relationships between the nodes.

4. System overview

This section illustrates the system model, the threat model, and the security definitions used in this paper.

4.1. System model

We envisage a data-driven three-tier ecosystem for medical health, shown in Fig. 1. Individual tier has a different purpose and is related to each other. At first, each $\mathcal{O}$ needs to complete its registration from a service provider $\mathcal{SP}$. After the completion of the registration, each $\mathcal{O}$ can authenticate itself and establish a secure connection throughout tiers. The three tiers are described below:

4.1.1. Tier-1

$Tier$-1 introduces data custodian $\mathcal{C}$. It serves as a trustworthy decentralized repository. $\mathcal{O}$ is the owner of their IoT data and this data are stored in the $\mathcal{C}$ by the smart contract after authentication. The $\mathcal{C}$ gets a fee every time while $\mathcal{O}$ inserts new data in the custodian. At first, the $\mathcal{O}$ gathers all the data from wearable IoT sensors and send them to their own smart contract. It pre-processes them and uses symmetric-key encryption on them with their own secret key $\mathcal{SK}$, which makes sure the security of the data. After encryption, all the data are stored in the $\mathcal{C}$. Further, the proposed system permits multiple parties to access the content on the servers while maintaining confidentiality such as doctors, researchers, and so on Hasan et al. (2020). Therefore, It is paramount to have a method, where content should be shared based on the permission of the $\mathcal{O}$. $\mathcal{C}$ only stores the encrypted data and the smart contracts hold the hash. Therefore, data are private to their owner $\mathcal{O}$. Fig. 2 illustrates the entire process.

The smart contract also provides a real-time health monitoring service to $\mathcal{O}$. It provides a suggestion to $\mathcal{O}$ to go to the hospital for a check-up in case of an abnormal measurement of body parameters (temperature, pressure, etc.). Smart contract deploys the rule-based method in order to perform this task. For general suggestions, rule-based methods are good enough in terms of time complexity, space complexity, and threshold-based abnormality identification in the human body. Entities involved in this tier are:

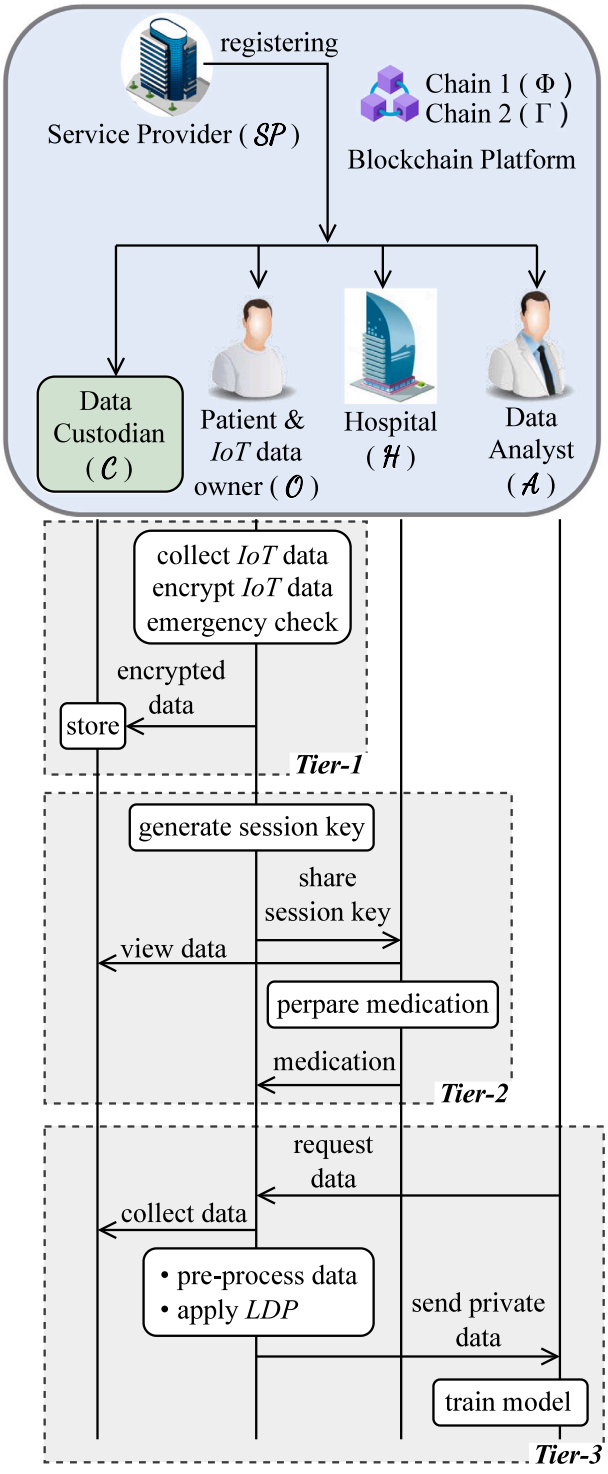


Fig. 1. Data-driven three-tier ecosystem for medical health.

- IoT **Devices:** are accountable for sensing and transferring important IoT data by wired and wireless networks.
- **Data Owners** $\mathcal{O}$: gather all pieces of IoT data from the IoT devices.
- **Data Custodian** $\mathcal{C}$: hospital or any cloud data storage service provider can serve as a trustworthy repository.

Formally, the proposed system in $Tier$-1 consist of n patients/data owners $\mathcal{O}_i$ $(i \in 1, \ldots, n)$ and a is the number of data custodians $\mathcal{C}_i$ $(i \in 1, \ldots, a)$. Each $\mathcal{O}_i$ holds a $\mathcal{C}$ and dataset $\mathcal{D}_i$. The dataset contains sensitive medical information of $\mathcal{O}$. The $\mathcal{O}$ store their data in the $\mathcal{C}$ by executing protocol $\mathcal{P}_{Tier\text{-}1}$.

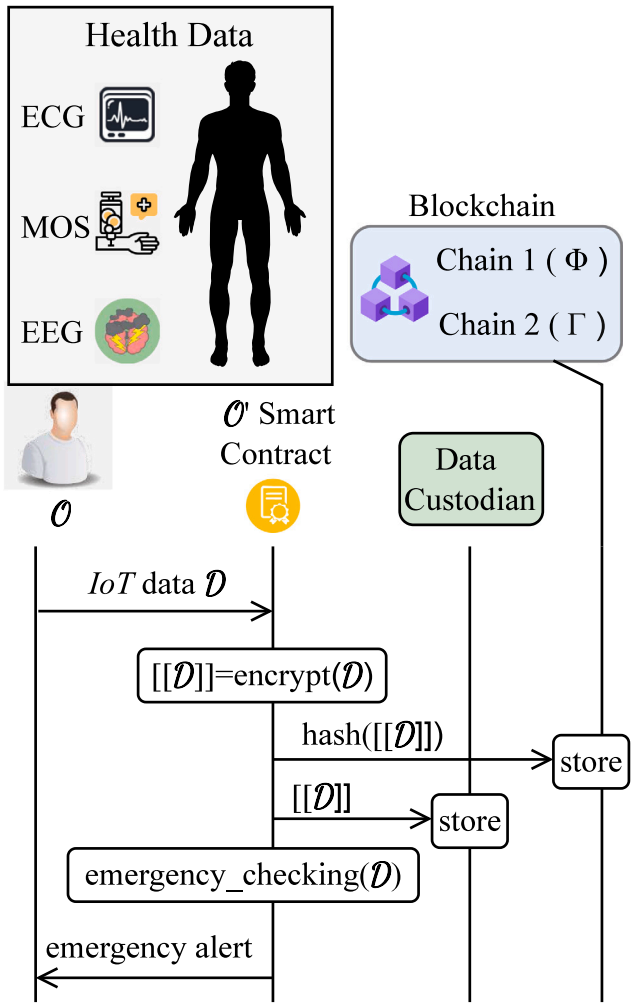


Fig. 2. Introduction of the trustworthy repository ($Tier$-1).

4.1.2. $Tier$-2

$Tier$-2 mainly focuses on the authentication of $\mathcal{O}$ and data sharing between $\mathcal{O}$ and $\mathcal{H}$. The entire process is shown in Fig. 3. The $\mathcal{O}$ visits the $\mathcal{H}$ and ask for medication. The $\mathcal{H}$ requests all the previous data of the $\mathcal{O}$. All the data is in the $\mathcal{C}$, and for proper medication, the physicians need to see the original data. The designed schema enables multiple parties to access the $IPFS$ or custodian $\mathcal{C}$ content confidentially. In simple terms, the $\mathcal{O}$ needs to recreate a different session key ($\mathcal{SEK}$) for each receiver from its own private $\mathcal{SK}$. This own private $\mathcal{SK}$ has been used in $Tier$-1 to encrypt the dataset before storing it in the $\mathcal{C}$.

To generate a $\mathcal{SEK}$ from the own private $\mathcal{SK}$, this study uses the key derivation function (KDF) or a cryptographic hash function. This study chooses Password-Based Key Derivation Function 2 ($PBKDF2$) and the key length is set to 255-bit. The parameters for generating the session key are the secret key, a salt value (random or user-specific), an iteration count (to slow down the derivation process), and the key length. After applying the $PBKDF2$ function, the required session key is generated. Now the role-based access control mechanism has been implemented. Here, the smart contract of $\mathcal{O}$ assigns roles to the hospital and defines permissions associated with each role. When a doctor logs in with a session key, verify their role and grant only view permissions if appropriate. Section 5.5 describes the entire process in detail.

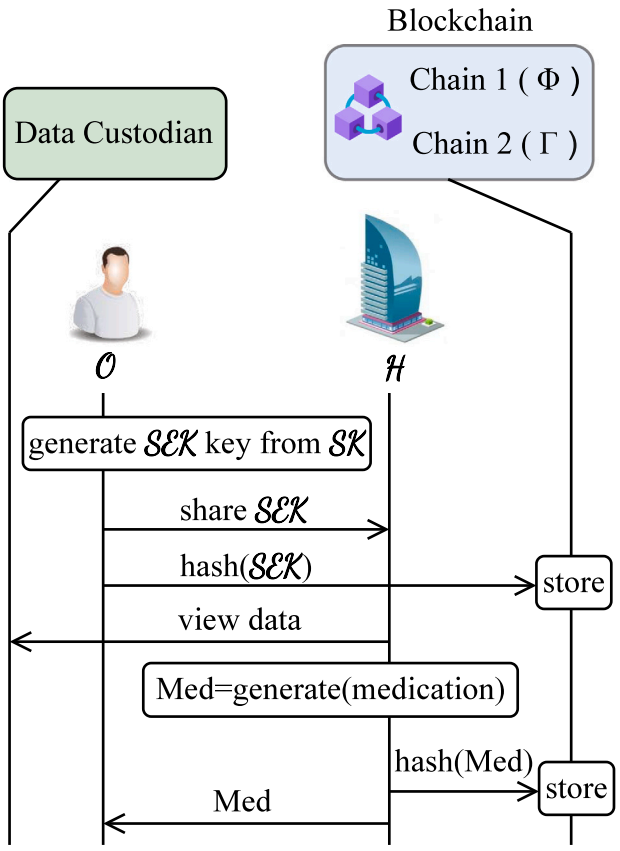


Fig. 3. Authentication and data sharing between $\mathcal{O}$ and $\mathcal{H}$ ($Tier$-2).

$\mathcal{O}$ encrypts the $\mathcal{SEK}$ with the public key of the hospital $\mathcal{PK}_{\mathcal{H}}$. The $\mathcal{O}$ sends the key to $\mathcal{H}$. The $\mathcal{H}$ then decrypts it using its private key $\mathcal{SK}_{\mathcal{H}}$ to get the session key. The receiver finally uses the session key to decrypt the content on $IPFS$ $\mathcal{C}$. Therefore, the $\mathcal{O}$ will provide a session key $\mathcal{SEK}$, which will allow the $\mathcal{H}$ to view the data from the $\mathcal{C}$. After viewing the data, the $\mathcal{H}$ will provide medication to the $\mathcal{O}$, and the newly given medication will also be added to the $\mathcal{C}$. Entities related to this tier are:

- **Data Owners/Patient** $\mathcal{O}$: responsible for sending an encrypted message $[[m]]$ and $\mathcal{SEK}$ to the $\mathcal{H}$.
- **Hospital:** responsible for sending medication and laboratory exam reports to the $\mathcal{O}$ and $\mathcal{C}$. $\mathcal{O}$ authentication is also done by the $\mathcal{H}$.
- **Data Custodian:** qualified for allowing permissions-based access to different users.

Formally, the proposed system in $Tier$-2 consist of n patients/data owners $\mathcal{O}_i$ $(i \in 1, \ldots, n)$ and h hospitals $\mathcal{H}_i$ $(i \in 1, \ldots, h)$. $\mathcal{O}$ visits $\mathcal{H}$. By executing protocol $\mathcal{F}_{Tier\text{-}2}$, the $\mathcal{O}_i$ get verified by $\mathcal{H}$ and also sends the session key $(\mathcal{SEK}_{\mathcal{O}_i})$ to the $\mathcal{H}$ in order to share the previous dataset D_i from data custodian C_i. It is essential to mention that the session key only enables the view option. The $\mathcal{O}$ share their data to the $\mathcal{H}$ by executing protocol $\mathcal{F}_{Tier\text{-}2}$.

4.1.3. Tier-3

Entire $Tier$-3 mainly focuses on secure data sharing between a $\mathcal{O}$ and $\mathcal{A}$ (i.e., Researchers, Pharmaceutical companies, etc.). $\mathcal{A}$ wants to train its ML model with the dataset D of $\mathcal{O}$. This model consists of two smart contracts (smart contract of $\mathcal{O}$ and smart contract of $\mathcal{A}$). After authentication smart contract of $\mathcal{O}$ will take the encrypted data from the data custodian, decrypt it, pre-process it, apply local differential

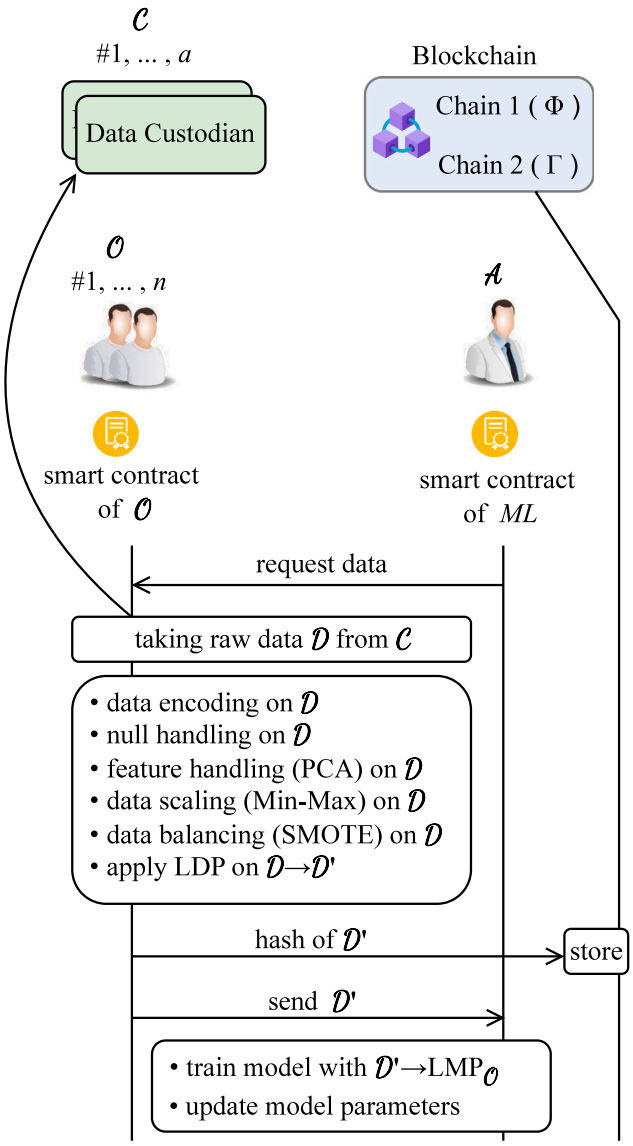


Fig. 4. System model for the data-driven ecosystem of $Tier$-3.

privacy, and send them to the smart contract of $\mathcal{A}$, which is holding the ML model. Fig. 4 shows the system model of this tier. Entities associated with this tier are:

- IoT **Devices:** IoT devices are accountable for sensing and transferring important IoT data by wired and wireless networks.
- **Data Owners/Patient / Data Provider** $\mathcal{O}$: $\mathcal{O}$ gathers all pieces of IoT data from the IoT devices.
- **Data Analyst:** $\mathcal{A}$ wants to train training an ML model upon the dataset gathered from multiple $\mathcal{O}$.

Generally, the proposed system in $Tier$-3 consist of n patients/data owners $\mathcal{O}_i$ $(i \in 1, \ldots, n)$ and an untrusted $\mathcal{A}$. Each $\mathcal{O}_i$ holds a data set D_i at data custodian C_i that contains sensitive information. This work considers horizontal data sharing, that is the n datasets $\{D_i\}_{i=i}^{n}$ share the same feature space but different in samples. The $\mathcal{A}$ sequentially gathers the n encrypted data, and trains a ML model upon the dataset $D := (D_1 \cup .... \cup D_n)$, where, $|D| = \sum_{i=1}^{n} |D_i|$. After executing privacy-preserving training protocols of $\mathcal{F}_{Tier\text{-}3}$, the $\mathcal{A}$ obtains the desired model.

4.2. Threat model

Except for $\mathcal{SP}$ no participants in the proposed model do not trust one another and all participants are considered as curious-but-honest antagonists (Moradi et al., 2021). Again, $\mathcal{AD}$ is a third-party foe that can intercept, modify and forge messages from the public channel. However, it fell to deduce confidential information from the private channel. The threat models are as follows.

- The $\mathcal{C}$ is honest in following the pre-defined protocol of the system. Therefore, the data of $\mathcal{O}$ are secured from alternation, but the $\mathcal{C}$ may try to learn the original data of $\mathcal{O}$.
- The $\mathcal{H}$ and the $\mathcal{O}$ are honest in following the system's pre-defined protocols. Different types of adversary attacks may occur here, such as: Over a public channel $\mathcal{AD}$ can modify/eavesdrop/delete the dispatched messages. On the other hand, $\mathcal{O}$ or $\mathcal{H}$ may deny their past action.
- The $\mathcal{A}$ follows the defined protocol but is also inquisitive to infer more information from intermediate data. $\mathcal{A}$ and various $\mathcal{O}$ can conspire together to infer the crucial information of other $\mathcal{O}$. Furthermore, $\mathcal{O}$ is also interested in $\mathcal{A}$'s model parameter.
- $\mathcal{AD}$ can infer communicating messages by initializing a passive attack and can initiate an active attack in order to accomplish any unauthorized operation. It can forge messages at the key agreement stage and gain the entity's session key.

4.3. Security goals

All security goals are described in this section. The entire privacy-preserving framework satisfies the following requirements.

- **Protocol** $\mathcal{F}_{Tier\text{-}1}$: The $\mathcal{C}_i$ failed to infer any sensitive information of $\mathcal{D}_i$. A $\mathcal{O}_i$ failed to infer any sensitive information of other $\mathcal{O}_i$ form the data custodian $\mathcal{C}_i$.
- **Protocol** $\mathcal{F}_{Tier\text{-}2}$: The integrity, confidentiality, and authenticity of the message sent by the $\mathcal{O}_i$ to The $\mathcal{H}_i$ cannot be thwarted by any adversary. The $\mathcal{H}_i$ cannot alter any sensitive information in the data set $\mathcal{D}_i$.
- **Protocol** $\mathcal{F}_{Tier\text{-}3}$: The $\mathcal{A}$ failed to infer any sensitive information of $\mathcal{D}$. A $\mathcal{O}$ failed to infer the parameters of the model. A $\mathcal{O}$ failed to infer any sensitive information of other O'.
- **Common protocol for all**: No entity can breach others' privacy and security. $\mathcal{AD}$ failed to forge and retrieve any information. Again, it is also unsuccessful in impersonating, forwarding secrecy, and replay attacks.

5. Model construction

The section describes the construction details of the proposed privacy-preserving framework.

5.1. System setup

The setup of the system is described here. $\mathcal{E}(\mathcal{F}_p)$ and $\mathcal{F}_p$ are elliptic curve and finite field are selected by $\mathcal{SP}$, respectively. A prime p decides $\mathcal{F}_p$. $\mathcal{P}$, q, and $\mathcal{SK}_{\mathcal{SP}}$ are generators, order on the curve, and master or secret key selected by $\mathcal{SP}$, respectively. The public key $\mathcal{PK}_{\mathcal{SP}} = (\mathcal{SK}_{\mathcal{SP}}\mathcal{P})$ is published by $\mathcal{SP}$. It also publishes p, q, $\mathcal{P}$, $h_i(.)(i = 1,2,3)$ where $h_i : \{0,1\}^* \rightarrow \mathcal{Z}_q^*, i = 1,2$. $h_3 : \{0,1\}^* \rightarrow \{0,1\}^n$. Integers modulo q's multiplicative group is $\mathcal{Z}_q^*$.

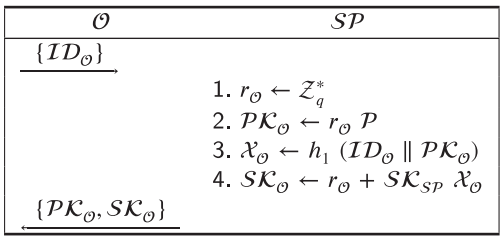


Fig. 5. $\mathcal{O}$'s registration process through protocol Φ.

5.2. Registration

The registration procedure (protocol Φ) of all participants ($\mathcal{O}$, $\mathcal{C}$, $\mathcal{H}$, and $\mathcal{A}$) with $\mathcal{SP}$ is described in this section. The process of registering $\mathcal{O}$ though $\mathcal{SP}$ is described here. The rest of the participants heed the exact methods. Identity $\mathcal{ID}_{\mathcal{O}}$ is submits by $\mathcal{O}$ to $\mathcal{SP}$. The $\mathcal{SP}$ generates a nonce $r_{\mathcal{O}} \in \mathcal{Z}_q^*$, and works out $\mathcal{PK}_{\mathcal{O}} = r_{\mathcal{O}}\ \mathcal{P}$, $\mathcal{X}_{\mathcal{O}} = h_1\ (\mathcal{ID}_{\mathcal{O}} \parallel \mathcal{PK}_{\mathcal{O}})$, and $\mathcal{SK}_{\mathcal{O}} = r_{\mathcal{O}} + \mathcal{SK}_{\mathcal{SP}}\ \mathcal{X}_{\mathcal{O}}$. Then, the $\mathcal{SP}$ sends $\{\mathcal{PK}_{\mathcal{O}}, \mathcal{SK}_{\mathcal{O}}\}$ to $\mathcal{O}$ secretly. $\mathcal{O}$'s registration process is showed in Fig. 5.

5.2.1. Data sharing via chain 1

$\mathcal{SP}$ uses the $\mathcal{PK}$ of other participants and generates the hash. The digital signature ($\mathcal{DS}$) is developed by encrypting them with the $\mathcal{SK}_{\mathcal{SP}}$. $\mathcal{SP}$ joins $\mathcal{PK}$s' of $\mathcal{O}$, $\mathcal{C}$, $\mathcal{H}$, and $\mathcal{A}$. Publicly inferable data are $\mathcal{DS}$ and $\mathcal{SN}_{\mathcal{SP}}$. The smart contract is called and concated data are committed in the blockchain by the $\mathcal{SP}$. The smart contracts' operational process for registration is shown in Algorithm 1. Here, for key generation and data register functions $gen()$ and $reg()$ are used respectively, on chain 1. The entire process is explained beneath:

$\mathcal{DS}_{\mathcal{O}}$ is generated by $\mathcal{SP}$ utilize Eq. (3) and then $(\mathcal{PK}_{\mathcal{O}} \| \mathcal{DS}_{\mathcal{O}} \| \mathcal{SN}_{\mathcal{SP}})$.

$$\mathcal{DS}_{\mathcal{O}} = \mathcal{EN}(h(\mathcal{PK}_{\mathcal{O}}), \mathcal{SK}_{\mathcal{SP}}) \tag{3}$$

Available public information from chain 1 are Entities' public key; Entities' verifiable digital signatures; Sign of the service provider.

Algorithm 1: Smart contract's working process for registration

```
O's Input: request req, ID_O.
SP's Input: r_O, P, SK_SP SN_SP.
O's Output: PK_O, SK_O.
if req == 1 then
    {DS, PK, SK}_O ← gen(r_O, P, SK_SP, ID_O);
    reg(DS_O, PK_O, SN_SP);
end
```

5.2.2. Analysis of security protocol Φ

The security analysis of registration (Protocol Φ) is described as follows.

Proposition 1. *Protocol Φ in Fig. 5 is protected from $\mathcal{AD}$.*

Proof. In Protocol Φ: $\mathcal{O}$, and $\mathcal{SP}$, two entities are involved. The actions and processes of other entities ($\mathcal{C}$, $\mathcal{H}$, and $\mathcal{A}$) are the same. Consequently, one design that is protected signifies others are protected. The analysis is done based on the design of Fig. 5. In order to assess the potential inference of private information among the involved entities, it is essential to examine the perspectives of each party. This analysis will help determine whether any party has the ability to derive sensitive information from others. The view of each $\mathcal{O}$ is

$$view_{\mathcal{O}}^{\Phi} = \left(\{\mathcal{ID}, \mathcal{PK}, \mathcal{SK},\}_O, \mathcal{P}, p, q, h_{1,2,3}(.)\right) \tag{4}$$

$\mathcal{C}$
1. $\mathcal{PK}_{\mathcal{SP}} \leftarrow recognizes\ \mathcal{SN}_{\mathcal{SP}}$
2. $h(\mathcal{PK}_{\mathcal{O}})_{\mathcal{SP}} \leftarrow (\mathcal{DE}(\mathcal{DS}_{\mathcal{O}}), \mathcal{PK}_{\mathcal{SP}})$
3. $\bar{\mathcal{H}} \leftarrow h(\mathcal{PK}_{\mathcal{O}})^{\mathcal{C}}$
4. $\bar{\mathcal{H}}\ ? = h(\mathcal{PK}_{\mathcal{O}})^{\mathcal{SP}}$

Fig. 6. Verification of $\mathcal{PK}$ and corresponding $\mathcal{SP}$ through protocol Γ.

Only entity involved here is $\mathcal{O}$ and $\mathcal{SP}$. Therefore, no participants can infer any private data of another participant. When $\mathcal{AD}$ is a curious but honest foe, it fails to predict and learn any confidential data. On the other hand, when $\mathcal{AD}$ is an adversary and not among the participants of the system also fails to infer any data as the system is under private Blockchain network. Therefore, $\mathcal{AD}$ fails to accept authority over the network. The only trusted entity is $\mathcal{SP}$. Lastly, discuss the privacy and security concerns connected to the public register of chain 1. Therefore, $\mathcal{AD}$'s view of blockchain ledger is:

$$view^{\Phi}_{\mathcal{AD}} = (\mathcal{PK}_{\mathcal{O}}, \mathcal{PK}_{\mathcal{SP}}, \mathcal{DS}_{\mathcal{O}}, \mathcal{SN}_{\mathcal{SP}}) \tag{5}$$

Now, $\mathcal{PK}_{\mathcal{O}}, \mathcal{PK}_{\mathcal{SP}}, \mathcal{DS}_{\mathcal{O}}$ and $\mathcal{SN}_{\mathcal{SP}}$ has no security concerns as they are just addresses. Thus, protocol Φ is protected from dishonest and curious-but-honest foe for Fig. 5. ■

5.3. Authentication

This section describes the authentication process and protocols Γ (authentication process). Each entity maintains protocol Γ at the time of interaction. The authentication process of $\mathcal{O}$ with $\mathcal{C}$ is described here and the rest of the participants follow the same protocol.

5.3.1. Verification of $\mathcal{PK}$ & corresponding $\mathcal{SP}$

This section describes the verification of participants' $\mathcal{PK}$, where any $\mathcal{PK}$ entities can determine the affiliated $\mathcal{SP}$. Consider a design where a $\mathcal{C}$ tries to affirm the $\mathcal{PK}$ and identify its corresponding $\mathcal{SP}$ of an $\mathcal{O}$. The entire scheme is illustrated in Fig. 6. $\mathcal{C}$ retrieves $\mathcal{O}$'s $\mathcal{PK}_{\mathcal{O}}$ along with $\mathcal{DS}_{\mathcal{O}}$ and $\mathcal{SN}_{\mathcal{SP}}$ from chain 1. It infers $\mathcal{PK}_{\mathcal{SP}}$ by $\mathcal{SN}_{\mathcal{SP}}$. It decrypts $\mathcal{DS}_{\mathcal{O}}$ with $\mathcal{PK}_{\mathcal{SP}}$ and gets $h(\mathcal{PK}_{\mathcal{O}})_{\mathcal{SP}}$, which is generated by $\mathcal{SP}$. It generates $h(\mathcal{PK}_{\mathcal{O}})^{\mathcal{C}}$ as $\bar{\mathcal{H}}$. It compares $\bar{\mathcal{H}}$ and $h(\mathcal{PK}_{\mathcal{O}})_{\mathcal{SP}}$, $\mathcal{SP}$ verify the $\mathcal{PK}_{\mathcal{O}}$, if matches. Other entities utilize this process to verify the $\mathcal{PK}$ of another participant in a similar way.

5.3.2. Authentication between $\mathcal{O}$&$\mathcal{C}$

This section describes the authentication process, which is shown in Fig. 7. $\mathcal{C}$ sends its identity to $\mathcal{O}$ using asymmetric encryption.

1. **Phase 1:** $\mathcal{O}$ chooses a nonce $u \in \mathcal{Z}^*_q$, $\mathcal{B}_1 = u\mathcal{P}$, $\mathcal{B}_2 = h_2(u\mathcal{PK}_{\mathcal{C}} + uh_1(\mathcal{ID}_{\mathcal{C}} \parallel \mathcal{PK}_{\mathcal{C}})\mathcal{PK}_{\mathcal{SP}}) \oplus \mathcal{ID}_{\mathcal{O}}$, $\mathcal{B}_3 = h_2(\mathcal{B}_1 \| \mathcal{PK}_{\mathcal{O}} \| \mathcal{PK}_{\mathcal{C}} \| \mathcal{ID}_{\mathcal{O}} \| \mathcal{ID}_{\mathcal{D}})$. Then the message $\mathcal{MA}_1 = \{\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3, \mathcal{PK}_{\mathcal{O}}\}$ is sent to $\mathcal{C}$.
2. **Phase 2:** $\mathcal{C}$ calculates $\mathcal{ID}_{\mathcal{O}} = \mathcal{B}_2 \oplus h_2(\mathcal{SK}_{\mathcal{C}}\mathcal{B}_1)$ and checks $\mathcal{B}_3 ? = h_2(\mathcal{B}_1 \| \mathcal{PK}_{\mathcal{O}} \| \mathcal{PK}_{\mathcal{C}} \| \mathcal{ID}_{\mathcal{O}} \| \mathcal{ID}_{\mathcal{C}})$. If true, $\mathcal{C}$ continues to select $v \in \mathcal{Z}^*_q$ and calculates $\mathcal{B}_4 = v\mathcal{P}$, $\mathcal{SEK}_{\mathcal{C}} = h_3(\mathcal{B}_1 \| \mathcal{B}_4 \| v\mathcal{B}_1)$, $\mathcal{B}_5 = h_2(v\mathcal{PK}_{\mathcal{O}} + vh_1(\mathcal{ID}_{\mathcal{O}} \parallel \mathcal{PK}_{\mathcal{O}})\mathcal{PK}_{\mathcal{SP}}) \oplus \mathcal{ID}_{\mathcal{C}}$, and $\mathcal{B}_6 = h_2(\mathcal{ID}_{\mathcal{O}} \| \mathcal{ID}_{\mathcal{C}} \| \mathcal{B}_1 \| \mathcal{B}_4 \| \mathcal{SEK}_{\mathcal{C}})$. Then the message $\mathcal{MA}_2 = \{\mathcal{B}_4, \mathcal{B}_5, \mathcal{B}_6\}$ is sent to $\mathcal{O}$.
3. **Phase 3:** $\mathcal{O}$ calculates $\mathcal{SEK}_{\mathcal{O}} = h_3(\mathcal{B}_1 \| \mathcal{B}_4 \| u\mathcal{B}_4)$, $\mathcal{ID}_{\mathcal{C}} = h_2(\mathcal{SK}_{\mathcal{O}}\mathcal{B}_4) \oplus \mathcal{B}_5$, and checks $\mathcal{ID}_{\mathcal{C}}$ and $\mathcal{B}_6 ? = h_2(\mathcal{ID}_{\mathcal{O}} \| \mathcal{ID}_{\mathcal{C}} \| \mathcal{B}_1 \| \mathcal{B}_4 \| \mathcal{SEK}_{\mathcal{O}})$. If true, then the two entities $\mathcal{O}$ and $\mathcal{C}$ are authenticated on both sides.

5.3.3. Data sharing via chain 2

Every entity's $\mathcal{PK}$ is verified by other participants at the time of the authentication stage through protocol Γ. In Fig. 7, $\mathcal{MA}_1$ hash is generated by $\mathcal{O}$ and calls the smart contract to commit it in the blockchain along with its own $\mathcal{PK}_{\mathcal{O}}$. Again, $\mathcal{C}$ generates the hash of the $\mathcal{MA}_2$, $[[\mathcal{ID}_{\mathcal{C}}]]_{\mathcal{PK}_{\mathcal{O}}}$ and register it in the shared ledger by contacting the smart contract along with own $\mathcal{PK}_{\mathcal{C}}$. Smart contract's authentication process is shown in Algorithm 2, where $reg()$ and $auth()$ functions represent to register and authentication for writing data into chain 2. The methodology is explained as follows:

- $\mathcal{O}$ generates $(\mathcal{PK}_{\mathcal{O}} \parallel \bar{\mathcal{H}}_{\mathcal{O}})$ using (6)

$$\bar{\mathcal{H}}_{\mathcal{O}} = h(\mathcal{MA}_1) \tag{6}$$

- $\mathcal{C}$ generates $(\mathcal{PK}_{\mathcal{C}} \parallel \bar{\mathcal{H}}_{\mathcal{C}})$ using (7)

$$\bar{\mathcal{H}}_{\mathcal{C}} = h(\mathcal{MA}_2 \parallel [[\mathcal{ID}_{\mathcal{C}}]]_{\mathcal{PK}_{\mathcal{O}}}) \tag{7}$$

All entities follow the same protocol for authentication. Publicly available information from chain 2 are the hash of entities public key and shared messages.

Algorithm 2: Working process of smart contract for authentication

```
O's Input: PK_O, SK_O, u, P, ID_O.
C's Input: ID_C, PK_C, SK_C, v, P.
if PK_O is in chain 1 then
    if auth(u, v, P, {SK, SM, ID}_{O|C}) == 1 then
        reg(PK_O, H̄_O, PK_C, H̄_C);
    end
end
```

5.3.4. Security analysis of protocol Γ

The security analysis of authentication (Protocol Γ) is described as follows.

Proposition 2. *Adversaries $\mathcal{AD}$ cannot infer any data from Fig. 7 (Protocol Γ).*

Proof. Entities $\mathcal{A}$, $\mathcal{O}, \mathcal{C}$, and $\mathcal{H}$ are involved in Protocol Γ. All participants follow the same protocol. Consequently, a scenario is protected which means all other scenarios are protected. Fig. 7 is considered in order to describe this scenario.

The view of each $\mathcal{O}$ is:

$$view^{\Gamma}_{\mathcal{O}} = (\mathcal{ID}_{\mathcal{C}}, u, \mathcal{P}, \mathcal{PK}_{\mathcal{SP}}, \mathcal{PK}_{\mathcal{C}}, \mathcal{SEK}_{\mathcal{O}}) \tag{8}$$

Here, $\mathcal{O}$ can authenticate $\mathcal{C}$ by checking $\mathcal{B}_6$ and no data is observable to $\mathcal{O}$ from where it can deduce additional confidential information. Besides, the view of individual $\mathcal{C}$ is:

$$view^{\Gamma}_{\mathcal{C}} = (\mathcal{ID}_{\mathcal{O}}, v, \mathcal{P}, \mathcal{PK}_{\mathcal{SP}}, \mathcal{PK}_{\mathcal{O}}, \mathcal{SEK}_{\mathcal{C}}) \tag{9}$$

$\mathcal{C}$ can authenticate $\mathcal{O}$ by checking $\mathcal{B}_3$ and no other data is visible to $\mathcal{C}$ for analysis. Again, it is important to analyze the view of $\mathcal{AD}$. $\mathcal{AD}$'s view is:

$$view^{\Gamma}_{\mathcal{AD}} = (\mathcal{PK}_{\mathcal{O}}, \mathcal{PK}_{\mathcal{SP}}, \mathcal{PK}_{\mathcal{C}}, \bar{\mathcal{H}}_{\mathcal{O}}, \bar{\mathcal{H}}_{\mathcal{C}}) \tag{10}$$

$\mathcal{AD}$ fails to learn any knowledge from $\mathcal{PK}_{\mathcal{O}}, \mathcal{PK}_{\mathcal{SP}}$, $\mathcal{PK}_{\mathcal{C}}$, $\bar{\mathcal{H}}_{\mathcal{O}}$ and $\bar{\mathcal{H}}_{\mathcal{C}}$ as $\mathcal{PK}$s' are addresses and no backward operation for hash values. $\mathcal{AD}$ is more informative then public data based on the threat model. It is clear that the $\mathcal{ID}_{\{\mathcal{O},\mathcal{C}\}}$ are secured by the hash values $h_2(u\mathcal{PK}_{\mathcal{O}} + uh_1(\mathcal{ID}_{\mathcal{O}} \parallel \mathcal{PK}_{\mathcal{O}})\mathcal{PK}_{\mathcal{SP}})$ and $h_2(v\mathcal{PK}_{\mathcal{C}} + vh_1(\mathcal{ID}_{\mathcal{C}} \parallel \mathcal{PK}_{\mathcal{C}})\mathcal{PK}_{\mathcal{SP}})$, respectively. The outcomes needs $\mathcal{SK}_{\mathcal{SP}}$ or $\mathcal{SK}_{\mathcal{O}}$ and $\mathcal{SK}_{\mathcal{SP}}$ or $\mathcal{SK}_{\mathcal{O}}$ to indirectly or directly forge those hash values. Only the respective owner has their private key. In forwarding secrecy, $\mathcal{AD}$ obtains and breaks all of the secret keys $\mathcal{SK}_{\mathcal{C}}$ and $\mathcal{SK}_{\mathcal{O}}$ from participants. Regardless, $\mathcal{AD}$ cannot deduce the one-time session keys because they are developed depending on the ECDH issue. Preservation of forward secrecy is done as $u, v, \mathcal{P}$ are not precisely computable. $\mathcal{AD}$ attempts to learn any data during the key agreement, it requires $\mathcal{SK}_{\mathcal{SP}}, \mathcal{SK}_{\mathcal{O}}$ or $\mathcal{SK}_{\mathcal{C}}$ for impersonation attack. However, $\mathcal{AD}$ fails to achieve anything, and invasion will fail. Lastly, $\mathcal{AD}$ depends on unknown random digits v and u each time for a reply attack. $\mathcal{AD}$ fails to break the ECDH issue based on $(u\ \mathcal{P}, v^{old}\ \mathcal{P})$ or $(u^{old}\ \mathcal{P}, v\ \mathcal{P})$, in spite of any messages, are being replied. Therefore, protocol Γ is secured from dishonest adversaries and curious-but-honest for Fig. 7. ■

$\mathcal{O}$	$\mathcal{C}$
	1. $[[\mathcal{ID}_C]]_{\mathcal{PK}_\mathcal{O}} \leftarrow \mathcal{EN}(\mathcal{ID}_C, \mathcal{PK}_\mathcal{O})$
$\leftarrow \{[[\mathcal{ID}_C]]_{\mathcal{PK}_\mathcal{O}}\}$	
2. $\mathcal{ID}_C \leftarrow \mathcal{DE}([[\mathcal{ID}_C]]_{\mathcal{PK}_\mathcal{O}}, \mathcal{SK}_\mathcal{O})$	
3. $u \leftarrow \mathcal{Z}_q^*$	
4. $\mathcal{B}_1 \leftarrow u\mathcal{P}$	
5. $\bar{\mathcal{H}}_2 \leftarrow h_2(u\mathcal{PK}_C + uh_1(\mathcal{ID}_C \parallel \mathcal{PK}_C)\mathcal{PK}_{\mathcal{SP}})$	
6. $\mathcal{B}_2 \leftarrow \bar{\mathcal{H}}_2 \oplus \mathcal{ID}_\mathcal{O}$	
7. $\mathcal{B}_3 \leftarrow h_2(\mathcal{B}_1 \parallel \mathcal{PK}_\mathcal{O} \parallel \mathcal{PK}_C \parallel \mathcal{ID}_\mathcal{O} \parallel \mathcal{ID}_C)$	
8. $\mathcal{MA}_1 \leftarrow \{\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3, \mathcal{PK}_\mathcal{O}\}$	
$\{\mathcal{MA}_1\} \rightarrow$	
	9. $\mathcal{ID}_\mathcal{O} \leftarrow \mathcal{B}_2 \oplus h_2(\mathcal{SK}_C\mathcal{B}_1)$
	10. $\bar{\mathcal{H}}_2' \leftarrow h_2(\mathcal{B}_1 \parallel \mathcal{PK}_\mathcal{O} \parallel \mathcal{PK}_C \parallel \mathcal{ID}_\mathcal{O} \parallel \mathcal{ID}_C)$
	11. check $\mathcal{B}_3 ? = \bar{\mathcal{H}}_2'$
	12. $v \leftarrow \mathcal{Z}_q^*$
	13. $\mathcal{B}_4 \leftarrow v\mathcal{P}$
	14. $\mathcal{SEK}_C \leftarrow h_3(\mathcal{B}_1 \parallel \mathcal{B}_4 \parallel v\mathcal{B}_1)$
	15. $\bar{\mathcal{H}}_2'' \leftarrow h_2(v\mathcal{PK}_\mathcal{O} + vh_1(\mathcal{ID}_\mathcal{O} \parallel \mathcal{PK}_\mathcal{O})\mathcal{PK}_{\mathcal{SP}})$
	16. $\mathcal{B}_5 \leftarrow \bar{\mathcal{H}}_2'' \oplus \mathcal{ID}_C$
	17. $\mathcal{B}_6 \leftarrow h_2(\mathcal{ID}_\mathcal{O} \parallel \mathcal{ID}_C \parallel \mathcal{B}_1 \parallel \mathcal{B}_4 \parallel \mathcal{SEK}_C)$
	18. $\mathcal{MA}_2 \leftarrow \{\mathcal{B}_4, \mathcal{B}_5, \mathcal{B}_6\}$
$\leftarrow \{\mathcal{MA}_2\}$	
19. $\mathcal{SEK}_\mathcal{O} \leftarrow h_3(\mathcal{B}_1 \parallel \mathcal{B}_4 \parallel u\mathcal{B}_4)$	
20. $\mathcal{ID}_C \leftarrow h_2(\mathcal{SK}_\mathcal{O}\mathcal{B}_4) \oplus \mathcal{B}_5$	
21. $\bar{\mathcal{H}}_2''' \leftarrow h_2(\mathcal{ID}_\mathcal{O} \parallel \mathcal{ID}_C \parallel \mathcal{B}_1 \parallel \mathcal{B}_4 \parallel \mathcal{SEK}_\mathcal{O})$	
22. check $\mathcal{ID}_C$ and $\mathcal{B}_6 ? = \bar{\mathcal{H}}_2'''$	

Fig. 7. Authentication process of $\mathcal{O}$ and C through protocol Γ.

$\mathcal{O}$	C
1. $[[D]]_{\mathcal{SK}_\mathcal{O}} \leftarrow enc(D, \mathcal{SK}_\mathcal{O})$	
2. $\mathcal{E} \leftarrow$ utilizing Algorithm 3	
$\{[[D]]_{\mathcal{SK}_\mathcal{O}}\} \rightarrow$	
	3. received

Fig. 8. $\mathcal{F}_{Tier\text{-}1}$: Protocol of data sharing (between the data owner and data custodian) and Real-time health monitoring service from the smart contract.

5.4. Tier-1

Now, we describe the data-sharing process from the $\mathcal{O}$ to C. After the completion of authentication following the protocol Γ, $\mathcal{O}$ and C develop a secure channel in between them. Fig. 8 specifies the protocol of $Tier$-1 ($\mathcal{F}_{Tier\text{-}1}$), which needs only one interaction. For simplicity, we are considering a scenario where participants are one $\mathcal{O}$ and a C. $\mathcal{O}$'s smart contract encrypt D, with his $\mathcal{O}$'s own private key $\mathcal{SK}_\mathcal{O}$ and send $[[D]]_{\mathcal{SK}_\mathcal{O}}$ to C. $\mathcal{O}$ also gets real-time health monitoring service from the smart contract. Smart-contract uses algorithm 3 [Algorithm 3 only to show the sample condition. In total, there are 55 conditions.], which is a rule-based method. If any parameter of the body shows an abnormal result, the $\mathcal{E}$ will be 1 else $\mathcal{E}$ will be 0. $\mathcal{O}$ will provide dataset D as input to $\mathcal{O}$'s smart contract. D consists of different feature sets (i.e, body temperature, EEG, EGA, heart-bit rate, etc.). From body temperature to the heart-bit rate all the feature has different threshold values, above which all values are considered abnormal. $\mathcal{E}$ equal to 1 represents abnormal conditions, $\mathcal{O}$ should visit doctors. Algorithm 4 shows the smart contract working process for $Tier$-1.

The security analysis of $\mathcal{F}_{Tier\text{-}1}$ is described as follows.

Proposition 3. *Protocol $\mathcal{F}_{Tier\text{-}1}$ in Fig. 8 is protected from $\mathcal{AD}$.*

Algorithm 3: Rule-based health monitoring system

```
O's Input: D = {[body_temp], [heart_bit], ...}
. O's Output: Emergency E
. Smart_contract initialize E equalto 0;
if [body_temp] ≥ 101 then
    Smart_contract computes (E = E ∪ 1);
end
...
else
    Smart_contract computes (E = E ∪ 0);
end
Smart_contract send E to O;
```

Algorithm 4: Working process of $Tier$-1's smart contract

```
O's Input: SK_O, D.
if D is received then
    encrypt(D,SK_O);
    if check(D) == 1 then
        send(emergency);
    end
end
```

Proof. In Protocol $\mathcal{F}_{Tier\text{-}1}$: $\mathcal{O}$ and C, two entities are involved. The view of C is

$$view_C^{\mathcal{F}_{Tier\text{-}1}} = ([[D]]_{\mathcal{SK}_\mathcal{O}}) \tag{11}$$

Now, secretiveness of $([[D]]_{\mathcal{SK}_\mathcal{O}})$ needs to be discussed, i.e., whether C can predict the private data of $\mathcal{O}$ from the value. The confidentiality

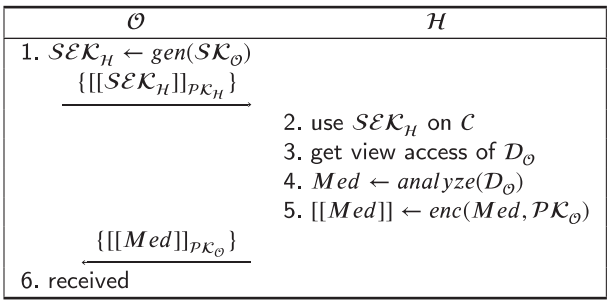


Fig. 9. $\mathcal{F}_{Tier\text{-}2}$: Protocol of data sharing (between the patient and Hospital).

of $([[D]]_{\mathcal{SK}_{\mathcal{O}}})$ is directly equivalent to the used Symmetric-key cryptosystem. So, $\mathcal{C}$ has no chance to get the original data. Except for brute force cracking, there is no more reliable way of determining the true value of $\mathcal{D}$'s. Assume that each $\mathcal{O}$'s has a 2-dimensional limited data set with 100 occurrences. Each dimension is worth 32 bits. [Single-precision floating-point typically takes up 4 bytes (32 bits) of memory space.] Based on this condition, the chance of $\mathcal{C}$'s and $\mathcal{AD}$'s correctly predicting is $\frac{1}{2^{(n\times 6400)}}$. It is a small but achievable possibility. So, under the curious-but-honest paradigm, Protocol $\mathcal{F}_{Tier\text{-}1}$ is secure. ■

5.4.1. Tier-1 Data sharing via chain 2

During the interaction of $Tier$-1, only the hash of the transferred data will be in the blockchain. The procedure is described in detail below:

$\mathcal{O}$'s smart contract generates $\bar{\mathcal{H}}_{Tier\text{-}1}$ using (12)

$$\bar{\mathcal{H}}_{Tier\text{-}1} \leftarrow h([[D]]_{\mathcal{SK}_{\mathcal{O}}}) \tag{12}$$

5.5. Tier-2

In this section, we discuss the data-sharing process between a $\mathcal{O}$ and a $\mathcal{H}$. After the completion of authentication following the protocol Γ, $\mathcal{O}$ and $\mathcal{H}$ verify them self. Fig. 9 specifies the protocol of $Tier$-2 ($\mathcal{F}_{Tier\text{-}2}$). For simplicity, we are considering a scenario where participants are one $\mathcal{O}$, $\mathcal{C}$, and a $\mathcal{H}$. $\mathcal{O}$ visits $\mathcal{H}$ for medication and $\mathcal{H}$ requests $\mathcal{O}$'s data. $\mathcal{O}$ and $\mathcal{H}$ authenticate each other using protocol Γ. After authentication $\mathcal{O}$ generates a $\mathcal{SEK}$ from its own private key $\mathcal{SK}$. The $PBKDF2$ function receives the key of length 255-bit, $\mathcal{SK}_{\mathcal{O}}$, a salt value (random or user-specific), and an iteration count (to slow down the derivation process) as parameters in order to generate the session key $\mathcal{SEK}_{\mathcal{H}}$. $\mathcal{O}$ encrypts the session key with the $\mathcal{PK}_{\mathcal{H}}$ and sends it to $\mathcal{H}$ after assigning the role-based access control mechanism. Here, the smart contract of $\mathcal{O}$ assigns roles to the hospital and defines permissions associated with each role. When a doctor logs in with a session key, verify their role and grant only view permissions if appropriate. $\mathcal{H}$ used the session key $\mathcal{SEK}_{\mathcal{H}}$ to get the view access of $\mathcal{O}$'s sensitive data D from $\mathcal{C}$. $\mathcal{H}$ provides medication report Med based on D and sends it to $\mathcal{O}$'s smart contract in encrypted manner. Algorithm 5 shows the working process of $Tier$-2's smart contract.

The security analysis of $\mathcal{F}_{Tier\text{-}2}$ is described as follows.

Proposition 4. *Protocol* $\mathrm{F}_{Tier\text{-}2}$ *in Fig. 9 is secure in an insecure channel and also protected from* $\mathcal{AD}$.

Proof. In Protocol $\mathrm{F}_{Tier\text{-}2}$: $\mathcal{O}$, $\mathcal{H}$ and $\mathcal{C}$, three entities are associated. The view of $\mathcal{H}$ is

$$view_{\mathcal{H}}^{\mathcal{F}_{Tier\text{-}2}} = (\mathcal{SEK}_{\mathcal{H}}, \mathcal{D}_{\mathcal{O}}, Med) \tag{13}$$

The security analysis of authentication protocol is already proved in Proposition 2. $\mathcal{H}$ is curious but honest adversaries. It needs to see the original $\mathcal{O}_{\mathcal{H}}$ in order to provide medication. This protocol only allows view access to $\mathcal{H}$, So, $\mathcal{H}$ cannot manipulate or change $\mathcal{O}_{\mathcal{H}}$. Now the view of adversary $\mathcal{AD}$ is

$$view_{\mathcal{AD}}^{\mathcal{F}_{Tier\text{-}2}} = ([[Med]]_{\mathcal{PK}_{\mathcal{O}}}) \tag{14}$$

The confidentiality of $([[Med]]_{\mathcal{PK}_{\mathcal{O}}})$ is directly equivalent to the used asymmetric-key cryptosystem. So, $\mathcal{AD}$ has no chance to get the original data. Except for brute force cracking, there is no more accurate way for determining the true value of Med. Assume that each $\mathcal{O}$ contains a two-dimensional, constrained data set with 100 occurrences. Each dimension has a value of 32 bits. [Single-precision floating-point typically takes up 4 bytes (32 bits) of memory space.] The likelihood of $\mathcal{AD}$ predicting correctly based on this condition is $\frac{1}{2^{(n\times 6400)}}$. It is a minuscule chance of success. As a result, protocol $\mathcal{F}_{Tier\text{-}2}$ is safe. ■

Algorithm 5: Working process of $Tier$-2's smart contract

```
1  O's Input: SK_O, PBKDF2( ), key_length kl, salt_value sv,
     iteration_count ic, PK_H.
2  H's Input: request req, medication Med.
3  if req is received then
4  |  SEK_H ← PBKDF2(SK_O, kl, sv, ic);
5  |  embed access control with SEK_H;
6  |  [[SEK_H]]_PK_H ← encrypt(SEK, PK_H);
7  |  send([[SEK_H]]_PK_H) to H;
8  end
9  if med is received then
10 |  send(med) to C;
11 end
```

5.5.1. Tier-2 Data sharing via chain 2

During the interaction of $Tier$-2, only the hash of the transferred data will be in the blockchain. The procedure is described in detail below:

$\mathcal{H}$'s smart contract generates $\bar{\mathcal{H}}_{Tier\text{-}2}$ using (15)

$$\bar{\mathcal{H}}_{Tier\text{-}2} \leftarrow h([[Med]]_{\mathcal{PK}_{\mathcal{O}}}, [[\mathcal{SEK}_{\mathcal{H}}]]_{\mathcal{PK}_{\mathcal{H}}}) \tag{15}$$

5.6. Tier-3

This section illustrates the data-sharing process between $\mathcal{O}$ and $\mathcal{A}$. After the completion of authentication following the protocol Γ, $\mathcal{O}$ and $\mathcal{A}$ verify themself. The aim is to secure the privacy of specific $\mathcal{O}$ and $\mathcal{A}$ while training any ML model across numerous private datasets from different $\mathcal{O}$. Fig. 10 shows the protocol of $Tier$-3, $\mathcal{F}_{Tier\text{-}3}$. For simplicity, this study considers single $\mathcal{A}$ wants to train his/her ML model with the data of n number of $\mathcal{O}$. Each $\mathcal{O}$ and $\mathcal{A}$ has its own smart contracts. The smart contract of $\mathcal{A}$ holds the ML model and the smart contract of $\mathcal{O}$ pre-process the data with data encoding, null handling, feature handling, data scaling, data balancing (Synthetic Minority Oversampling Technique SMOTE), and finally with $\varepsilon{-}LDP$. $\mathcal{A}$ requests data from various $\mathcal{O}$. $\mathcal{O}$'s smart contract takes dataset D from $\mathcal{C}$ and pre-process them. $\mathcal{O}$'s smart contract sets $\varepsilon - LDP$ parameters α [$\alpha \in$ (privacy budget ε, sensitivity δ, ..., etc.)] from $\mathcal{O}$ in order to apply $\varepsilon - LDP$ and generates D'. $\mathcal{A}$'s smart contract receives the differentially private data D' and apply ML algorithms. $\mathcal{A}$ will provide model parameters β [$\beta \in$ (centroid, learning rate, ..., etc.)] as input to its smart contract. After training $\mathcal{A}$ gets updated ML model parameters β' and also gets the classification results. This study uses ML and ensemble learning (EL) algorithms. While confronted with any curious but honest foe, no participant will be able to deduce any sensitive data of other participants from the intermediate outputs of the algorithm. Algorithm 6 shows the working process of $Tier$-3's smart contract.

The security analysis of $\mathcal{F}_{Tier\text{-}3}$ is described as follows.

$\mathcal{O}$	$\mathcal{A}$
1. $\mathcal{D}' \leftarrow \varepsilon\_\mathcal{LDP}(D, \alpha)$	
$\{\mathcal{D}'\}$ →	
	2. received
	3. $\beta' \leftarrow ML\_model(\mathcal{D}', \beta)$

Fig. 10. $F_{Tier\text{-}3}$: Protocol of data sharing between the data owner and data analyst.

Algorithm 6: Working process of $Tier$-3's smart contract

```
1 O's Input: D.
2 A's Input: request req.
3 if req is received then
4 |   D ← preprocess(D);
5 |   D' ← LDP(D);
6 |   send(D) to A;
7 end
```

Proposition 5. *Protocol $\mathcal{F}_{Tier\text{-}3}$ in Fig. 10 is protected from $\mathcal{AD}$.*

Proof. Fig. 10 represents Protocol $\mathcal{F}_{Tier\text{-}3}$, where $\mathcal{O}$ and $\mathcal{A}$ are the participants. All $\mathcal{O}$ act in a similar manner. Therefore, the security satisfaction of one $\mathcal{O}$ is enough to declare all $\mathcal{O}$ is secure. Individual IoT data owner's $\mathcal{O}$ view is

$$view_{\mathcal{O}}^{\mathcal{F}_{Tier\text{-}3}} = (D, \mathcal{D}', \alpha) \tag{16}$$

where $\mathcal{O}$, $\mathcal{O}'$, α are the parameters of $\mathcal{O}$ and they are operated by individual $\mathcal{O}$'s own smart contract. So, none of the $\mathcal{O}$ can inter other $\mathcal{O}$'s value and cannot even get the ML model parameters of $\mathcal{A}$. The view of $\mathcal{A}$ is

$$view_{\mathcal{A}}^{\mathcal{F}_{Tier\text{-}3}} = (\mathcal{D}', \beta, \beta') \tag{17}$$

Now, the $\mathcal{D}'$'s secrecy must be evaluated in order to understand the predictability of $\mathcal{A}$ from $\mathcal{D}'$ on confidential $\mathcal{D}$ of $\mathcal{O}$. Unquestionably, $\mathcal{D}'$ does not represent any solution of unexplored $\mathcal{D}$. $\mathcal{A}$ might attempt to infer undisclosed $\mathcal{D}$ by randomly guessing, the private α. It is not possible for $\mathcal{A}$ to identify the private $\mathcal{D}$ of $\mathcal{O}$ because the α is unknown to $\mathcal{A}$. So, $\mathcal{F}_{Tier\text{-}3}$ is protected from $\mathcal{AD}$. ■

5.6.1. Tier-3 Data sharing via chain 2

During the interaction of $Tier$-3, the transferred data's hash will be stored on the blockchain. The procedure is described in detail below:

$\mathcal{O}$'s smart contract generates $\bar{\mathcal{H}}_{Tier\text{-}3}$ using (18)

$$\bar{\mathcal{H}}_{Tier\text{-}3} \leftarrow h(\mathcal{D}') \tag{18}$$

6. Experiment and result evaluation

6.1. Testbed

All processes are accomplished on MacBook Pro implemented with memory (1600 MHz DDR3 4 GB), Intel processor (2.5 GHz) Core i5. Random Forest implemented in the Google Chrome Browser in Python 3 on Google's Collaboratory. Hyperledger Fabric platform's configuration: Memory 66 GB. Processor: W-2135 CPU (6 Core), 3.70 GHz Intel(R), Xeon(R). Product Name: NVIDIA GeForce RTX 2080 Ti. GPU: Attached GPUs: 4. Blockchain transaction platform's configuration: Processor: 8-Core 2.3 GHz Core i9 Intel. Memory, 16 GB 2667 MHz DDR4. GPU: Graphics 630 1536 MB Intel UHD.

Table 3
Statistics of datasets.

Measures	Datasets		
	DD	HDD	BCWD
Instances	768	303	699
Attributes	9	13	9
Discrete	0	13	0
Numerical	9	0	9

6.2. Dataset

Three medical datasets have been used namely Diabetes Data Set (DD) (Dua and Graff, 2019a), Heart Disease Data Set (HDD) (Dua and Graff, 2019b), and Breast Cancer Wisconsin Data Set (BCWD) (Dua and Graff, 2019c). Statistics of the datasets are represented in Table 3. Fig. 11 shows the feature correlation and principal component analysis (PCA) of the employed datasets. Based on Fig. 11, it is clear that BCWD dataset's features are positively correlated and most of the features have significance in PCA analysis. On the other hand, DD and HDD datasets do not have any correlation with their feature sets and their features have less significance in PCA analysis. The training set is 80% and the testing set is 20%.

6.3. Score evaluation metric

This section describes score evaluation metrics based on which performance of the proposed system will be measured. Kappa Statistic measures the agreement's degree between two sets of categorized data. It differs in intervals of 0 to 1. The higher the value of Kappa means the stronger the agreement. Again, error rates are also calculated mainly for $Tier$-1. For error measurement, four parameters are considered, they are: (1) RMSE Root mean squared error $RMSE = \sqrt{\frac{\sum_{j=1}^{m}(a_j - \hat{a}_j)^2}{m}}$, (2) MAE Mean absolute error $MAE = \frac{\sum_{j=1}^{m}|\hat{a}_j - \bar{a}|}{m}$ and (3) RRSE Root relative squared error $RRSE = \sqrt{\frac{\sum_{j=1}^{m}(a_i - \hat{a}_j)^2}{\sum_{j=1}^{m}(a_i - \bar{a})^2}}$. Here, $\bar{a}$, $\hat{a}$, a, m stands for mean value, actual value, predicted value, and the number of the test sets, respectively.

Again mainly for evaluating ML and EL algorithms in $Tier$-3, this study uses the three most popular metrics, i.e., $Accuracy(a) = \frac{t_p + t_n}{t_p + t_n + f_p + f_n}$, $Precision(p) = \frac{t_p}{t_p + f_p}$, $Recall(r) = \frac{t_p}{t_p + f_n}$, $f1 - score(f) = 2 \times \frac{Precision \times Recall}{Precision + Recall}$ and Area under the ROC Curve AUC score. Here, the positive or relevant classes has represented as t_p. These classes are precisely labeled. The negative or irrelevant classes that are labeled correctly are represented as f_p. f_n and t_n represent the number of relevant but mislabeled and the number mislabeled but irrelevant, respectively in the test result.

6.4. Scalability evaluation metric

Scalability is measured based on the time of execution, mean latency, and mean throughput, where (ts_1: Transaction deployment time and ts_2: Transaction confirmation time in the blockchain). Here, $Execution\ Time = \sum_{p=1}^{q}(ts_2 - ts_1)$ and $Latency = ts_2 - ts_1$. Average latency can be calculated from latency: $Average\ Latency = \frac{\sum_{p=1}^{q}(ts_2 - ts_1)}{q}$. The throughput is also calculated from average latency, $Throughput = \frac{1}{Average\ Latency}$ and $Average\ Throughput = \frac{Throughput}{n}$.

6.5. Result evaluation

This section evaluates the measured scores and scalability of the proposed system. Here, units s and tps stand for second and transaction per second, respectively. Fig. 12 illustrates the scalability analysis of the framework. Fig. 12.(a) shows the total execution time analysis

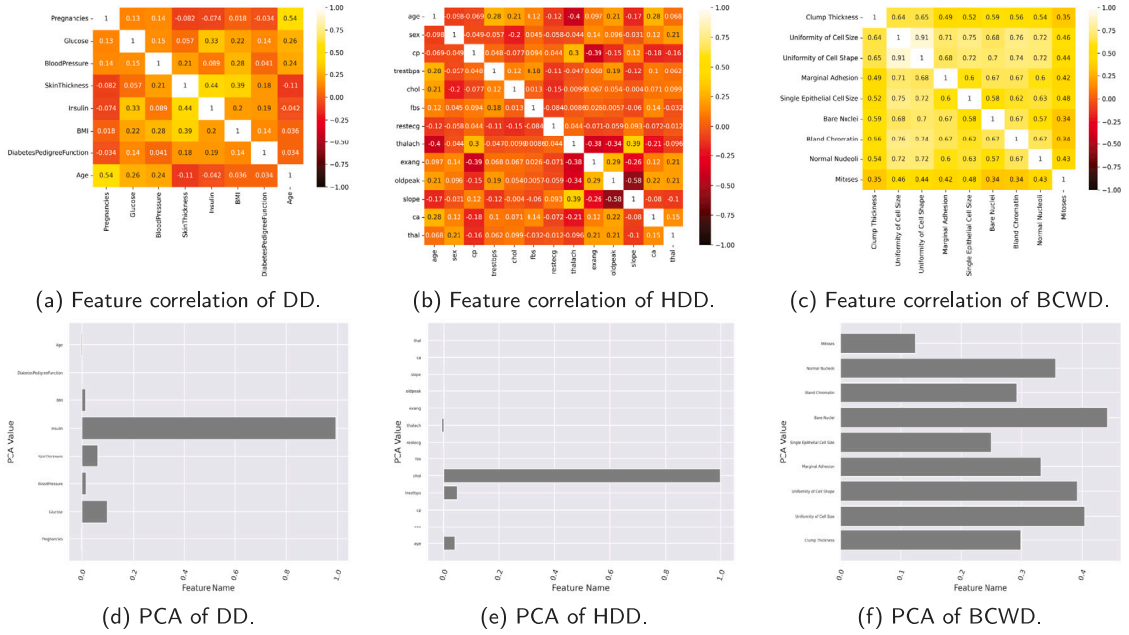


Fig. 11. Feature correlation and PCA of DD, HDD, and BCWD dataset.

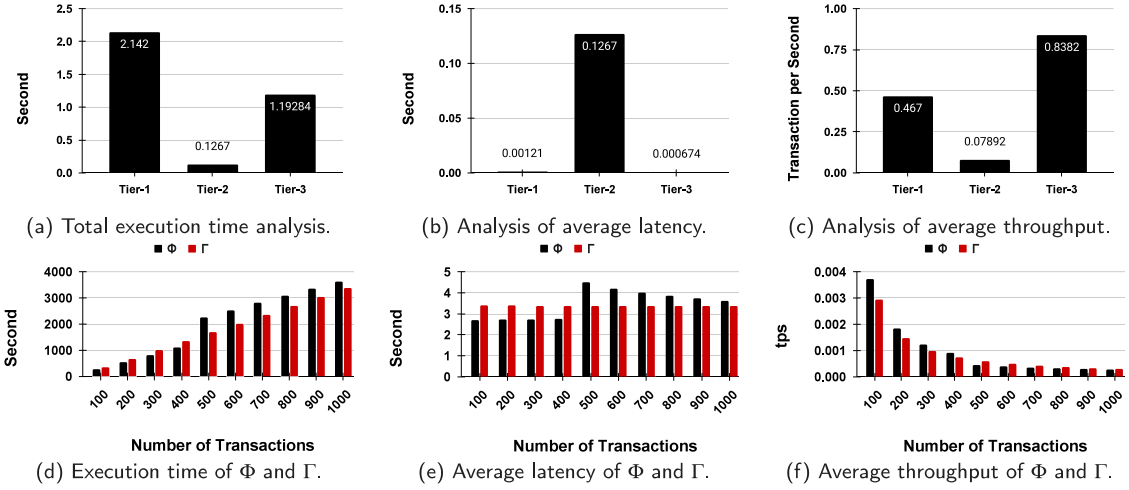


Fig. 12. Scalability analysis of proposed framework.

of each tier in the proposed framework. *Tier*-1 and *Tier*-3 are much higher than *Tier*-2 in the case of total execution time. Fig. 12.(b) and 12.(c) shows the detailed analysis of the average latency and average throughput of each tier in the framework, respectively. Fig. 12.(d) shows the execution time of protocols Φ and Γ in Hyperledger for the transactions number. The X-axis and the Y-axis illustrate transaction counts, which range from 1 to 1000, and a particular batch's transaction time, respectively. Based on the performance analysis, this execution time of Hyperledger is practical. The average latency and throughput for the two protocols are assessed and noted in Fig. 12.(e) and (f).

6.5.1. *Tier-1*

This section analyzes the performance of *Tier*-1. This study uses the same datasets across all tiers of the framework for that reason datasets are prepared separately to handle individual tiers. *Tier*-1 uses The Advanced Encryption Standard (AES-128) symmetric key encryption. The key size is 128 bits. Measures are evaluated based on kappa score, error rate, and scalability. Table 4 shows the performance analysis of real-time health monitoring in *Tier*-1. The proposed rule-based model achieved 0.98, 0.90, and 0.95 kappa scores in BCWD, HDD, and DD data sets respectively. Here, the kappa score represents the similarity between the actual emergency alert and the emergency alert shown by the proposed system. So, it is clear that the proposed method achieves a reliable score on three different datasets in order to be used in real life. On the other hand, Table 5 shows the scalability analysis of *Tier*-1. Encryption times are 0.025s, 0.738s, and 0.029s for BCWD, HDD, and DD datasets, respectively. Time for health monitoring is 0.2s, 0.9s, and 0.25s for BCWD, HDD, and DD datasets, respectively. Table 5 also shows that each iteration in *Tier*-1 executes in milliseconds, which is an acceptable performance. Table 6 shows the formal performance analysis of *Tier*-1 with previous works. Here, the communication costs are compared as 160-bit output is considered for hash function SHA-1 and humming weight, random numbers, and identities. Further, the size of a timestamp is 32 bits.

6.5.2. *Tier-2*

Tier-2 considers a scenario, where 100 patients want to share their data with the hospital and get medication. Based on this setup the total

Table 4
Score analysis of $Tier$-1.

Dataset	Measures			
	Kappa score	RMSE	MAE	RRSE
BCWD	0.98	0.45	3.5	0.08
HDD	0.90	0.18	10.7	0.25
DD	0.95	0.36	5.43	0.16

Table 5
Scalability analysis of $Tier$-1.

Measures	Dataset		
	BCWD	HDD	DD
Total execution time	0.225 s 2.142 s	1.638 s	0.279 s
Encryption time	0.025 s	0.738 s	0.029 s
Health monitoring time	0.2 s	0.9 s	0.25 s
Each iteration time	0.0003 s	0.005 s	0.0004 s
Average latency	0.00121 s		
Average throughput	0.467 tps		

Table 6
Formal analysis of $Tier$-1.

Study	Ct	Co-co	Ex-me
Miyachi and Mackey (2021)	✗	✗	✗
Deepa and Pandiaraja (2021)	$\mathcal{T}_{Ex}+2\mathcal{T}_{Pa}+\mathcal{T}_{En}$	✗	4
Ghayvat et al. (2021)	$5\mathcal{T}_{Pa}+3(\mathcal{T}_{Ex}+\mathcal{T}_{En}+\mathcal{T}_{Ha}+\mathcal{T}_{Ot})$	2050	6
$Tier$-1	$3\mathcal{T}_{Sm}+\mathcal{T}_{Dm}+\mathcal{T}_{Mm}$ $6\mathcal{T}_{Ha}+2\mathcal{T}_{En}$	1568	4

Here, Computation time (Ct) in millisecond (ms); Comparison of communication cost (Co-co) in bits; the number of the exchanged message (Ex-me); ✗: "Evaluation is not available"; $\mathcal{T}_{Ex}$—time required for exponential; $\mathcal{T}_{Pa}$—time required for paring; $\mathcal{T}_{Ha}$—time required for hashing; $\mathcal{T}_{En}$—time required for encryption; $\mathcal{T}_{Sm}$—time required for scalar multiplication; $\mathcal{T}_{Dm}$—time required for double scalar multiplication; $\mathcal{T}_{Mm}$—time required for modular multiplication; $\mathcal{T}_{Ot}$—time required for other operations.

Table 7
Scalability analysis of $Tier$-2.

Measures	Value for single $\mathcal{O}$	Average value for 100 $\mathcal{O}$
Execution time	0.1023 s	0.1267 s
Time of $\mathcal{O}$	0.098 s	0.12 s
Time for doctors feedback	X s	X s
Time of $\mathcal{H}$	$X+0.0043$ s	$X+0.0067$ s
Average latency	–	0.1267 s
Average throughput	–	0.07892 tps
Key size	255 bits	

scalability performance is measured and noted in Tables 7 and 8. $Tier$-2 uses Rivest, Shamir, and Adleman algorithm (RSA). The key size is 255 bits due to limited resources. Execution time for a single $\mathcal{O}$ and average execution time for 100 $\mathcal{O}$ are measured and shown in Table 7 in order to illustrate the efficiency of the proposed system. In Table 7 signature generation and signature verification consume 0.098 s and 0.0043 s but the execution time is 0.1023 s. Most of the time consumes the doctor's feedback and in real life, it may vary for that reason this study considers that doctors take Xs for providing diagnosis reports to $\mathcal{O}$. The average latency is 0.1267 s, and the average throughput is 0.07892t ps. Table 8 shows the formal analysis of $Tier$-2 with previous works. The following values are determined in the same setup as Table 6.

6.5.3. Tier-3

This section describes the performance analysis of $Tier$-3. Tables 9–11 shows the three cross-validation score analysis on DD, HDD, and BCWD datasets of the top three models based on accuracy, precision, recall, f1-score, and AUC score. Three medical data sets are employed with various ML and EL models to evaluate the performance of the

Table 8
Formal analysis of $Tier$-2.

Study	Ct	Co-co	Ex-me
Wu et al. (2017)	$8\mathcal{T}_{En}+20\mathcal{T}_{Ha}$	3968	6
Merabet et al. (2020)	$5\mathcal{T}_{Sm}+\mathcal{T}_{Ad}$	1728	12
Khan et al. (2020)	$4\mathcal{T}_{Ha}+\mathcal{T}_{En}+\mathcal{T}_{Dn}$	1440	5
$Tier$-2	$3\mathcal{T}_{Sm}+\mathcal{T}_{Dm}+\mathcal{T}_{Mm}$ $7\mathcal{T}_{Ha}+2\mathcal{T}_{En}$	1568	4

Here, $\mathcal{T}_{Dn}$—time required for decryption; $\mathcal{T}_{Sm}$—time required for scalar multiplication; $\mathcal{T}_{Ad}$—time required for point addition.

Table 9
Summary of performance of top three models on DD dataset with three cross validation.

Method	Measures				
	a	p	r	f	AUC
Random Forest	0.77	0.78	0.76	0.77	0.85
k-nn	0.76	0.74	0.82	0.78	0.76
Ada Boosting	0.76	0.75	0.77	0.76	0.82

Table 10
Summary of performance of top three models on HDD dataset with three cross validation.

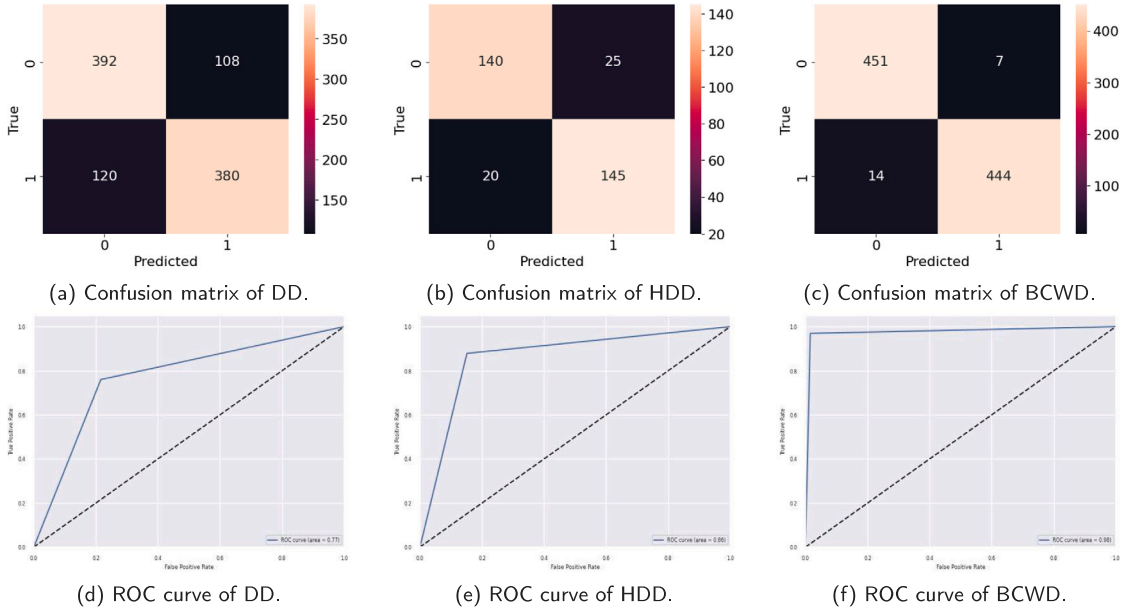
Method	Measures				
	a	p	r	f	AUC
Random Forest	0.86	0.86	0.88	0.86	0.93
k-nn	0.84	0.85	0.83	0.84	0.84
SVM (poly)	0.84	0.83	0.87	0.84	0.92

Table 11
Summary of performance of top three models on BCWD dataset with three cross validation.

Method	Measures				
	a	p	r	f	AUC
Random Forest	0.98	0.98	0.97	0.98	0.99
SVM (rbf)	0.97	0.98	0.96	0.97	0.99
SVM (linear)	0.97	0.98	0.97	0.97	1.00

proposed system. From the analysis, it is clear that Random Forest outperforms all other employed methods on all three datasets. The confusion matrix and receiver operating characteristic curve (ROC) curve of the proposed Random Forest is shown in Fig. 13. Now, it is important to check the performance of the proposed method with state-of-the-art models. Table 12 shows the comparison of the proposed method with the state-of-the-art models based on accuracy a, cross-validation cv, and data balancing db. Again, Table 13 shows the accuracy of the LDP-based analysis for the Random Forest algorithm. This study choice various values for privacy budget ε, where, $\varepsilon \in \{0, 1, 3\}$. It also shows that the increase of privacy budget ε is directly proportional to the scores but inversely proportional to the privacy. Moreover, Random Forests outperform all other methods in all datasets. The accuracy of Random forest is 97%, 83%, and 74% at $\varepsilon = 3$, where other methods are around 96% 81% and 65% at $\varepsilon = 3$ on BCWD, HDD, and DD corpus respectively. The suggested scheme confirms the robustness of both discrete and numerical attributes datasets and also protects privacy. Table 13 shows the fluctuation in accuracy scores between $\varepsilon - LDP$ data and original data.

On the other hand, Table 14 shows the analysis of time consumption for $Tier$-3. The total execution time is 1.19284 s, which is consist of a Machine learning algorithm (Random Forest) and three datasets (BCWD, HDD, and DD). HDD has only 303 discrete items but highest in data pre-processing time. The execution time of $\varepsilon - LDP$ is proportional to the size of the dataset. Moreover, the total execution time for each dataset consumes less than half of a second for each dataset with ML algorithms, which is an acceptable performance for real life.



(a) Confusion matrix of DD. (b) Confusion matrix of HDD. (c) Confusion matrix of BCWD.

(d) ROC curve of DD. (e) ROC curve of HDD. (f) ROC curve of BCWD.

Fig. 13. Confusion matrix and ROC curve of proposed Random Forest on DD, HDD, and BCWD dataset.

Table 12
Performance analysis of Random Forest with the state-of-the-art methods based on Accuracy. Here, n/a means not applicable.

Method	a	cv	db	Datasets	This study a
Shen et al. (2019)	0.97	n/a	n/a	BCWD	0.98
	0.81	n/a	n/a	HDD	0.86
Zhang et al. (2022)	0.96	n/a	n/a	BCWD	0.98
Akter et al. (2022)	0.65	n/a	n/a	DD	0.77
	0.82	n/a	n/a	HDD	0.86
	0.97	n/a	n/a	BCWD	0.98
Zhu et al. (2021)	0.96	n/a	n/a	BCWD	0.98
Jia et al. (2020)	0.93	n/a	n/a	BCWD	0.98
Le Nguyen et al. (2020)	0.93	n/a	n/a	BCWD	0.98
	0.80	n/a	n/a	HDD	0.86
Zhao et al. (2023)	0.97	n/a	n/a	BCWD	0.98
	0.72	n/a	n/a	DD	0.77

Table 13
Accuracy summary of the proposed LDP based Random Forest.

Measures	Dataset	Standard	LDP-Privacy Budget		
			$\epsilon = 0$	$\epsilon = 1$	$\epsilon = 3$
Accuracy	BCWD	0.98	0.94	0.95	0.97
	HDD	0.86	0.80	0.81	0.83
	DD	0.77	0.71	0.72	0.74

Table 14
Scalability analysis of $Tier$-3.

Measures	Dataset		
	BCWD	HDD	DD
Total execution time	0.394111 s 1.19284 s	0.35218 s	0.44655 s
Time for data pre-processing	0.00011 s	0.002 s	0.00015 s
Time for applying $\epsilon - LDP$	0.005 s	0.00018 s	0.0054 s
Time for $\mathcal{O}$	0.00511 s	0.00218 s	0.00555 s
Time for ML (Random Forest)	0.389 s	0.350 s	0.441 s
Time for $\mathcal{A}$	0.389 s	0.35 s	0.441 s
Average latency	0.000674 s		
Average throughput	0.8382 tps		

7. Conclusion

This study introduces a novel three-tier distributed privacy-preserving architecture for health care. Each of the three-tier addresses three different issues. $Tire-1$ introduces a trusted data repository $\mathcal{C}$, and also provides a real-time health monitoring system depending on a rule-based algorithm. The symmetric cryptosystem was used to encrypt data of $\mathcal{O}$ for ensuring privacy and store it in $\mathcal{C}$. The proposed rule-based model in $Tier$-1 achieved 0.98, 0.99 0.95 kappa scores in BCWD, HDD, and DD, respectively, which means that the proposed method is highly accurate in terms of prediction. $Tier$-2 focuses on data sharing between $\mathcal{O}$ and $\mathcal{H}$, where asymmetric encryption and the digital signature were used to complete the user authentication process. $\mathcal{O}$ generates a session key and implements access control in order to permit $\mathcal{H}$ to have view access to the data from the $\mathcal{C}$. This process helps to ensure data integrity and availability. In $Tier$-2 the execution time, average latency, and average throughput are 0.01267 s, 0.01267 s, and 0.07892t ps, respectively. $Tier$-3 employs a method based on $\epsilon - LDP$, blockchain, and smart contract for secure data sharing between the $\mathcal{O}$ and $\mathcal{A}$ in order to train the ML algorithm. Here blockchain ensures the authenticity of the transactions and $\epsilon - LDP$ ensures confidentiality. In $Tier$-3 Random Forest outperforms all other methods, where accuracy was 97%, 83%, and 74% at $\epsilon = 3$ on BCWD, HDD, and DD datasets, respectively, and time consumption for each data set is less than half a second. Moreover, the proposed architecture outperform all the previous state-of-the-art and showed satisfying performance. On the other hand, the proposed method ensures the privacy, security, and authenticity of all the entities. Firstly, all the interactions throughout the tiers are recorded in the blockchain, so authenticity has been assured. Now, in $Tier$-1 all the data are encrypted and stored in the custodian, which means that the data confidentiality is preserved. Again, in $Tier$-2 the session key only allows view access to the hospital. So, data security and authenticity have been confirmed. Finally, in $Tier$-3 the data owner used LDP, which guarantees data privacy from all kinds of attacks such as middleman attacks, passive attacks, and so on. The effectiveness and security of the proposed framework are demonstrated in this study. Note that most of the tasks are managed by smart contract and all intermediate data are recorded into the blockchain.

A significant limitation of this study is that participants are unable to verify the identity of other entities or the issuer of their identity. To address this, future research aims to develop a healthcare framework based on a decentralized identity model. This framework would allow all participants to verify each other's identities. Additionally, future work involves integrating federated learning and deep learning techniques to analyze real-time healthcare data and ensure security and privacy.

CRediT authorship contribution statement

Rakib Ul Haque: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Resources, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **A.S.M. Touhidul Hasan:** Conceptualization, Investigation, Methodology, Project administration, Supervision, Validation, Writing – review & editing. **Apubra Daria:** Software. **Abdur Rasool:** Writing – original draft. **Hui Chen:** Project administration. **Qingshan Jiang:** Project administration, Supervision, Validation. **Yuqing Zhang:** Funding acquisition, Project administration, Supervision, Validation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

Akter, S., Reza, F., Ahmed, M., 2022. Convergence of blockchain, k-medoids and homomorphic encryption for privacy-preserving biomedical data classification. Internet Things Cyber-Phys. Syst.

Anderson, C., Baskerville, R., Kaul, M., 2023. Managing compliance with privacy regulations through translation guardrails: A health information exchange case study. Inf. Organ. 33 (1), 100455.

Ayvaz, S., Alpay, K., 2021. Predictive maintenance system for production lines in manufacturing: A machine learning approach using IoT data in real-time. Expert Syst. Appl. 173, 114598.

Deepa, N., Pandiaraja, P., 2021. E health care data privacy preserving efficient file retrieval from the cloud service provider using attribute based file encryption. J. Ambient Intell. Humaniz. Comput. 12 (5), 4877–4887.

Dua, D., Graff, C., 2019a. UCI Machine Learning Repository. University of California, School of Information and Computer Science, Irvine, CA, https://archive.ics.uci.edu/ml/datasets/diabetes. (Accesed 12 December 2022).

Dua, D., Graff, C., 2019b. UCI Machine Learning Repository. University of California, School of Information and Computer Science, Irvine, CA, https://archive.ics.uci.edu/ml/datasets/heart+disease. (Accesed 12 December 2022).

Dua, D., Graff, C., 2019c. UCI Machine Learning Repository. University of California, School of Information and Computer Science, Irvine, CA, https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin(Diagnostic). (Accesed 12 December 2022).

Erlingsson, Ú., Pihur, V., Korolova, A., 2014. Rappor: Randomized aggregatable privacy-preserving ordinal response. In: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security. pp. 1054–1067.

Ghayvat, H., Pandya, S.N., Bhattacharya, P., Zuhair, M., Rashid, M., Hakak, S., Dev, K., 2021. CP-BDHCA: Blockchain-based confidentiality-privacy preserving big data scheme for healthcare clouds and applications. IEEE J. Biomed. Health Inf.

Gorkhali, A., Li, L., Shrestha, A., 2020. Blockchain: A literature review. J. Manag. Anal. 7 (3), 321–343.

Hasan, H.R., Salah, K., Jayaraman, R., Arshad, J., Yaqoob, I., Omar, M., Ellahham, S., 2020. Blockchain-based solution for COVID-19 digital medical passports & immunity certificates. IEEE Access 8.

He, D., Kumar, N., Chen, J., Lee, C.C., Chilamkurti, N., Yeo, S.S., 2015. Robust anonymous authentication protocol for health-care applications using wireless medical sensor networks. Multimedia Syst. 21 (1), 49–60.

Issa, W., Moustafa, N., Turnbull, B., Sohrabi, N., Tari, Z., 2023. Blockchain-based federated learning for securing internet of things: A comprehensive survey. ACM Comput. Surv. 55 (9), 1–43.

Jayabalan, J., Jeyanthi, N., 2022. Scalable blockchain model using off-chain $IPFS$ storage for healthcare data security and privacy. J. Parallel Distrib. Comput.

Jia, N., Fu, S., Xu, M., 2020. Privacy-preserving blockchain-based nonlinear SVM classifier training for social networks. Secur. Commun. Netw. 2020, 1–10.

Khan, M.A., Quasim, M.T., Alghamdi, N.S., Khan, M.Y., 2020. A secure framework for authentication and encryption using improved ECC for IoT-based medical sensor data. IEEE Access 8, 52018–52027.

Laghari, A.A., Wu, K., Laghari, R.A., Ali, M., Khan, A.A., 2021. A review and state of art of Internet of Things (IoT). Arch. Comput. Methods Eng. 1–19.

Le Nguyen, B., Lydia, E.L., Elhoseny, M., Pustokhina, I., Pustokhin, D.A., Selim, M.M., et al., 2020. Privacy preserving blockchain technique to achieve secure and reliable sharing of IoT data. Comput. Mater. Contin. 65 (1), 87–107.

Liu, L., Su, J., Zhao, B., Wang, Q., Chen, J., Luo, Y., 2020. Towards an efficient privacy-preserving decision tree evaluation service in the Internet of Things. Symmetry 12 (1), 103.

Merabet, F., Cherif, A., Belkadi, M., Blazy, O., Conchon, E., Sauveron, D., 2020. New efficient M2C and M2M mutual authentication protocols for IoT- based healthcare applications. Peer-to-Peer Netw. Appl. 13 (2), 439–474.

Miyachi, K., Mackey, T.K., 2021. hOCBS: A privacy-preserving blockchain framework for healthcare data leveraging an on-chain and off-chain system design. Inf. Process. Manage. 58 (3), 102535.

Moradi, A., Venkategowda, N.K., Talebi, S.P., Werner, S., 2021. Distributed Kalman filtering with privacy against honest-but-curious adversaries. In: 2021 55th Asilomar Conference on Signals, Systems, and Computers. IEEE, pp. 790–794.

Nakamoto, S., 2008. Bitcoin: A peer-to-peer electronic cash system. Available online: https://bitcoin.org/bitcoin.pdf. (Accessed 19 2018).

Pasdar, A., Lee, Y.C., Dong, Z., 2023. Connect API with blockchain: A survey on blockchain oracle implementation. ACM Comput. Surv. 55 (10), 1–39.

Paul, M., Maglaras, L., Ferrag, M.A., AlMomani, I., 2023. Digitization of healthcare sector: A study on privacy and security concerns. ICT Express.

Rahman, M.A., Hossain, M.S., 2021. An internet-of-medical-things-enabled edge computing framework for tackling COVID-19. IEEE Internet Things J. 8 (21), 15847–15854.

Roy, C., Mukherjee, A., Chaki, N., 2022. Merkle DAG-based distributed data model for content-addressed trust-less verifiable data. In: 2022 7th International Conference on Computer Science and Engineering. UBMK, IEEE, pp. 462–467.

Shen, M., Tang, X., Zhu, L., Du, X., Guizani, M., 2019. Privacy-preserving support vector machine training over blockchain-based encrypted IoT data in smart cities. IEEE Internet Things J. 6 (5), 7702–7712.

Tang, X., Zhu, L., Shen, M., Peng, J., Kang, J., Niyato, D., Abd El-Latif, A.A., 2022. Secure and trusted collaborative learning based on blockchain for artificial intelligence of things. IEEE Wirel. Commun. 29 (3), 14–22.

Wu, F., Xu, L., Kumari, S., Li, X., 2017. An improved and anonymous two-factor authentication protocol for health-care applications with wireless medical sensor networks. Multimedia Syst. 23 (2), 195–205.

Wu, F., Xu, L., Li, X., Kumari, S., Karuppiah, M., Obaidat, M.S., 2018. A lightweight and provably secure key agreement system for a smart grid with elliptic curve cryptography. IEEE Syst. J. 13 (3), 2830–2838.

Yu, R., Oguti, A.M., Ochora, D.R., Li, S., 2022. Towards a privacy-preserving smart contract-based data aggregation and quality-driven incentive mechanism for mobile crowdsensing. J. Netw. Comput. Appl. 207, 103483.

Zhang, P., Huang, T., Sun, X., Zhao, W., Liu, H., Lai, S., Liu, J.K., 2022. Privacy-preserving and outsourced multi-party k-means clustering based on multi-key fully homomorphic encryption. IEEE Trans. Dependable Secure Comput.

Zhao, J., Zhu, H., Wang, F., Lu, R., Wang, E., Li, L., Li, H., 2023. VFLR: An efficient and privacy-preserving vertical federated framework for logistic regression. IEEE Trans. Cloud Comput.

Zhu, L., Tang, X., Shen, M., Gao, F., Zhang, J., Du, X., 2021. Privacy-preserving machine learning training in IoT aggregation scenarios. IEEE Internet Things J. 8 (15), 12106–12118.

Rakib Ul Haque is currently working (remotely) as a research fellow at the Department of Digital Humanities, University of Religions and Denominations, Pardisan, Qom, Iran. He received his Master's degree in Computer Science and Technology from the University of Chinese Academy of Sciences, Beijing, China. He also worked as a lecturer at the Santo Marium University of Creative Technology, Dhaka, Bangladesh. Further, he also worked as a reviewer in several journals and conferences. His research interests include information security, deep learning, and self sovereign identity.

A.S.M. Touhidul Hasan is currently an Assistant Teaching Professor at the Division of Computing, Analytics and Mathematics, School of Science and Engineering, University of Missouri-Kansas City, USA. He received a Ph.D. degree in Computer Applied Technology from the Shenzhen Institutes of Technology, University of Chinese Academy of Science, China, in 2018. In 2014, he was a Doctoral Fellow at the Shenzhen Institutes of Advanced Technology, University of Chinese Academy of Sciences, China. His research interests include privacy preserving data publishing, information security, blockchain applications, and IoT privacy and security.

Apubra Daria is a blockchain Engineer at Technohaven Company LTD and a Research Engineer at the Institute of Automation Research and Engineering. His research interests include privacy preserving data publishing, information security, blockchain applications, AI, Data Science, and IoT privacy and security.

Abdur Rasool is currently a Ph.D. student at Shenzhen Institutes of Advanced Technology, University of Chinese Academy of Sciences, Shenzhen, China. He received the master's degree in computer science and technology with Donghua University, Shanghai, China. He received the B.S. degree in software engineering from Government College University, Faisalabad, Pakistan, in 2015. His primary research interests include DNA data storage, machine learning, data mining, sentiment analysis, natural language processing, and social media analysis.

Hui Chen is currently a lecturer in the Shenzhen Polytechnic. She received the Ph.D. degree in the Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences in 2022. She received the M.S. degree in Information and Communication Engineering from Nanjing University of Posts and Telecommunications in 2014. Her research interests include data security, machine learning, data mining, and pattern recognition.

Qingshan Jiang is currently a professor with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China. He received his Ph.D. degree in mathematics from the Chiba Institute of Technology, Japan, in 1996, and the Ph.D. degree in computer science from the University of Sherbrooke, Canada, in 2002. In 1999, he was a Postdoctoral Fellow with The Fields Institute for Research in Mathematical Sciences, University of Toronto, Canada. His research interests include data mining, information security, pattern recognition, massive data analysis, and database technology.

Yuqing Zhang received the Ph.D. degree in cryptography from Xidian University, Xi'an, China. He is a Professor and the Director of National Computer Network Intrusion Protection Center, University of Chinese Academy of Sciences, Beijing, China. He has authored or coauthored over 100 research papers in international journals and conferences, such as the ACM CCS, the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, and the IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING. His research has been sponsored by NSFC, Huawei, Qihu360, and Google. His current research interests include network and system security and applied cryptography.