HIDEchain: A User-Centric Secure Edge Computing Architecture for Healthcare IoT Devices

Rabeya Bosri*, Abdur Razzak Uzzal†, Abdullah Al Omar‡, Md Zakirul Alam Bhuiyan§, and Mohammad Shahriar Rahman¶

*†‡ Department of Computer Science and Engineering, University of Asia Pacific, Dhaka, Bangladesh
§ Department of Computer and Information Sciences, Fordham University, New York, NY 10458, USA
¶ Department of Computer Science and Engineering, University of Liberal Arts Bangladesh, Dhaka, Bangladesh
*rabeyabosri.cse@gmail.com, †abdurrazzakuzzal.aru@gmail.com, ‡omar.cs@uap-bd.edu,
§bhuiyan3@fordham.edu, ¶shahriar.rahman@ulab.edu.bd

***Abstract*—Cloud-based IoT architectures are expecting a fast data transfer rate within a limited bandwidth. But achieving a fast data transfer rate is becoming more difficult gradually, with the massive amount of data produced by the IoT devices. The concept of edge computing gives a solution to this problem and provides fast interaction between data centers and IoT devices. With the help of edge computing, the data transfer rate has been increased drastically within a limited bandwidth. As a result, edge computing is being used in different IoT based industry including the IoT based healthcare industry. On the other hand, the distributed nature of the edge computing paradigm introduces security and privacy concerns (e.g., data theft). The IoT based healthcare solutions are getting popular among the users. Therefore, users' highly sensitive data is at high risk of theft before eventually reaching the cloud. Nowadays, these issues are being solved by introducing blockchain in different architectures. Hence, this paper is utilizing blockchain technology for providing a user-centric secure edge computing architecture for healthcare IoT devices. We are proposing to integrate blockchain to store the users' data transfer records and ensuring data security by managing the number of blockchain transactions. We are providing user-centricity by allowing users to monitor their data transactions.**

***Index Terms*—Healthcare IoT devices, Blockchain, Edge Computing, network bandwidth , Data security and privacy**

I. Introduction

Edge computing is a distributed paradigm of computing that extends the cloud to the edge of the network. The concept of edge computing leverages when cloud could not support the IoT-based architecture with acceptable data transfer rate and quick response. IoT-based architecture requires real-time data analysis within a limited bandwidth [1]. For many cloud computing architectures (e.g, healthcare IoT based architectures), providing real time data analysis becomes difficult due to the increasing amount of data. The increasing development of healthcare IoT devices produces a vast amount of health data [2]. To fulfill users' requirement (e.g., real-time response), IoT-based architectures are moved from cloud computing to the edge computing paradigm. The edge computing paradigm improves the response time by allowing some applications to work on the edge server.

Corresponding Authos. § ¶

If the sensitive data collected by the IoT devices could be easily intercepted by the adversaries, it poses a serious threat for most of the edge computing based IoT applications [3]. Security and privacy issues in edge computing has become a concerning issue for healthcare IoT-based architectures. Because the user health data is useful in different research and it is also a valuable resource for commercial application producers [4]. The distributed structure of edge computing causes a significant privacy and security issue. In edge computing, data are processed at the edge that causes the leakage of privacy-sensitive data of end-users [5].

Recent IoT-based smart healthcare architectures have been designed by utilizing blockchain to solve the security and privacy issues [6] [7] [8]. However, it is not clear how users' sensitive data are being protected against the adversaries. There is no mechanism for the users to keep track of, how the system is using their data for which purposes [7] [9]. As user data is controlled by the service provider, hence there is always a chance of user data leakage to the unauthorized party by the service provider itself.The motivation behind this work is to utilize the blockchain to build a mechanism for the user to monitor how their data is being used by the system.

Our Contribution: In this paper, a user-centric edge computing architecture has been proposed which guarantees a secure edge computation paradigm for healthcare IoT devices. The main idea of our work is to keep user data transaction records on the blockchain. By utilizing the stored hashes a user could observe how the architecture is using her data. Blockchain stores the data transactions ensuring the data integrity. Thus, the service providers will not be able to share user data with any unauthorized party. User data remains anonymous in the edge node which provides user anonymity in the architecture. Only the registered IoT devices can transfer data with the proposed architecture. Hence, device authorization and authentication is also guaranteed. A rigorous security analysis has been added in Section VI to justify our security claims.

Paper Organization: The remainder of the paper is organized

as follows: Related work is described in Section II, preliminaries is discussed in Section III. Section IV explains our methodology. Section V describes the protocol construction. The protocol analysis of our architecture has been described in Section VI. Section VII describes the experimental evaluation and concluding remarks is included in Section VIII.

II. Related Work

With the rising health consciousness, more IoT devices are being used for healthcare. As a result, edge computing is becoming common with the growing development of IoT devices for the healthcare sector. Recent edge computing architectures pose obstacles to data security and privacy [10], [11]. Many experiments have been carried out to improve the security, efficiency, and usability of edge computing and IoT platforms [12].

Rahmani *et al*. introduced a well-planned IoT based healthcare system using the concept of fog computing and smart e-health [7]. They investigated the high-level services that can be provided to the sensors through smart gateways at the edge of the network. They have demonstrated the smart e-health gateway system. An intermediary processing layer is formed to demonstrate the fog computing concept. In the smart e-health gateways operating system is used (e.g., Linux's IPtable) to provide data security.

Hamid *et al*. proposed an agreement protocol that generates a key among the participants and communicates securely [13]. Data are accessed and stored by implementing a decoy technique. The original medical big data is kept secured in the cloud and decoy medical big data is kept in the fog. Original medical big data can only be accessed after authentication.

Omar *et al*. introduced MediBchain, a patient-centric data management system. They used blockchain as a storage to achieve privacy and protect patient data from cyber attacks [14]. They proposed that all medical data should be stored in blockchain. With the approval of the authorized patients and doctors, the stored data can be accessed. They have also made improvements to the MediBchain by making it more efficient and patient-centric in [15].

Pahl *et al*. discussed to implement three kinds of blockchain to store the data in [16]. Pham *et al*. proposed the implementation of blockchain-based smart contracts on Ethereum to store the computed data in the cloud. They also suggested that the sensor data can be filtered before storing the data into the blockchain to reduce the size of blockchain [8]. Although the data were filtered before storing, the growing IoT devices generate enormous amount of data. It is quite challenging to store these data in the blockchain. The interaction between blockchain and the IoT devices will increase the latency, which decreases the benefit of using IoT devices in the healthcare platforms.

III. Preliminaries

A. Notations

Table I describes the notations used in this paper.

TABLE I: Terminology table

Notation	Description
$\mathcal{D}_{\mathrm{i}}$	Healthcare IoT Devices
$\mathcal{N}_{\mathrm{i}}$	Edge Node
$\mathcal{F}$	Data Filter
$\mathcal{S}$	Data Server
$\mathcal{CU}$	Computational Unit
$\mathcal{C}$	Cloud
$\mathcal{BC}$	blockchain
$\mathcal{U}$	User
$\mathcal{HM}$	Health Monitoring Unit
$\mathcal{MD}_{\mathrm{i}}$	MAC Address of a device
δ	Inputted Data
δ_{req}	Data Transfer Request
$\delta_{req^{\text{successful}}}$	Acceptance of The Data Transfer Request
δ^{valid}	Valid data
δ^{invalid}	Garbage value
$\delta_{req^{\text{unsuccessful}}}$	Rejection of The Data Transfer Request
$\mathcal{R}_{\mathrm{i}}$	Result Generated By the $\mathcal{CU}$
$\mathcal{H}$	Hash value generated by the Cloud
$\mathcal{H}'$	Hash value coming from a device
$\mathcal{H}^{\text{stored}}$	A set of hash value stored by the node
$\mathcal{K}$	Private key
P	Prime number

B. Properties

- Authorization: Through registration, edge nodes will get the hash values of the registered devices.
- Authentication: Edge node gives the data sending permission to the device.
- Anonymity: Users are unknown to the edge nodes.
- Privacy: Users personal data are protected from the unauthorized healthcare monitoring units.
- Integrity: The consistency of data transaction records will be kept unchanged in the architecture.
- Verifiability: A user can verify and check on his data transaction records.
- Security: The access of user data is limited to the authorized healthcare units.

C. Assumption

The data transaction between the IoT device and the edge node is secured by using the cryptographic tools. Using separate networks to isolate devices also helps with establishing secure and private communication. As a result, the data transaction remains confidential. In this proposed architecture, edge node acts as a trusted entity and cloud uses SHA-256 to calculate the hash.

D. Cryptographic Tools

Private key: The private key is kept secret by the generator and never transmitted over a network. Suppose, q is a prime number.

Select q such that: $2^{159} \leq q \leq 2^{160}$

Select a random number $\mathcal{K}$ such that: $1 \leq \mathcal{K} \leq q-1$

Here, $\mathcal{K}$ is the primary key.

Hash Function: A hash function (h(x)) is a computationally efficient function mapping binary strings (x) of arbitrary

length to binary strings of some fixed length, called hash-values. A function is given and input x, hence, h(x) is easy to compute. For our proposed architecture,

$$x = \mathcal{MD}_i \parallel \mathcal{K}$$
$$\mathcal{H} = h(x)$$

$\mathcal{MD}_i$ is the address of a device and $\mathcal{H}$ is the required hash value.

IV. Our proposed architecture

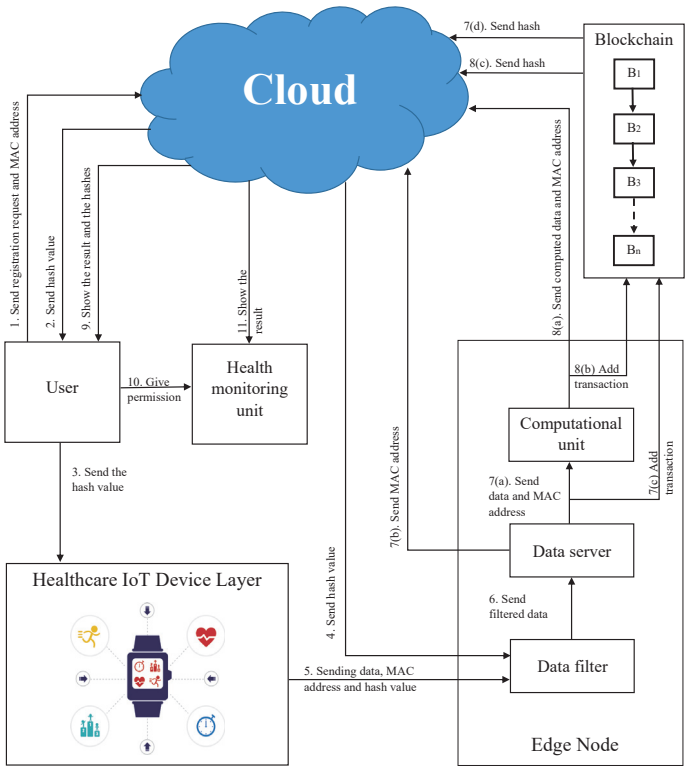

Fig. 1: Architecture of HIDEchain

Fig. 1 shows the architecture of HIDEchain. Blockchain is used to record all the data transactions of this architecture. To ensure that the information is not used for any other purpose than computing, the users in this architecture can track their personal data transactions by auditing the hashes.

A. Entities of Our Architecture:

a) ***Healthcare IoT Devices*** *($\mathcal{D}$):* A set of devices $\mathcal{D}$, which can be used to monitor health conditions. Set of devices could be expressed as follows:

$$\mathcal{D} \subset [\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3, ..., \mathcal{D}_i]$$

These devices make a connection with the proposed architecture by the users.

b) ***Edge Node*** *($\mathcal{N}$):* A set of nodes which are connected with our system. It could be expressed as follows:

$$\mathcal{N} \subset [\mathcal{N}_1, \mathcal{N}_2, \mathcal{N}_3 ..., \mathcal{N}_i]$$

The edge node of the proposed architecture gathers health data from the registered devices and works as a data storage near to the end-devices. $\mathcal{N}_i$ collects data from the IoT devices and store the data for further actions (i.e., real-time data analysis, real-time response, computation). In this system, the edge node performs some computational analysis on data. After computation, the results are sent to the cloud data centers in a periodic order.

In this proposed architecture, $\mathcal{N}_i$ consists of three entities, i. Data Filter ii. Data Server iii. Computational Unit

i. Data Filter ($\mathcal{F}$): Data filtering is the task of reducing the noise and error from a collection of data. In this architecture, $\mathcal{F}$ removes the unwanted data coming from the $\mathcal{D}_i$. As an instance, if a pulse sensor generates some data which is not in the range to the pulse of a human body. Then $\mathcal{F}$ will not grant permission to these kind of data to be stored in the data server and sends a notification to the D_i that it is generating the erroneous data.

ii. Data Server ($\mathcal{S}$): $\mathcal{S}$ works as a storage of data that come after the filtering phase. $\mathcal{S}$ stores the raw data and sends the data to the Computational Unit for further computation.

iii. Computational Unit ($\mathcal{CU}$): To reduce the latency and to provide a real-time response, computation take place in the $\mathcal{N}_i$. In this architecture, the $\mathcal{CU}$ runs computation on the raw data and sends the result in the Cloud.

c) ***Cloud****($\mathcal{C}$):* The cloud, $\mathcal{C}$ contains a data storage and is connected with users, edge node and monitoring unit. $\mathcal{C}$ stores the user's information, hashes coming from the blockchain, and result from the computational unit. It also makes a connection with the Health Monitoring Unit of this architecture.

d) ***Blockchain*** *($\mathcal{BC}$):* Blockchain is an entity of this system. We are using permissioned $\mathcal{BC}$ as only the permitted $\mathcal{N}$ can send data transactions records. The configuration of permissioned $\mathcal{BC}$ controls the data transactions and defines how a $\mathcal{U}$ can access and contribute to the $\mathcal{BC}$. User job is the task of transaction verification. A block contains multiple transactions [1] before adding a new block into the chain a verification request will be sent to the $\mathcal{U}$. When the $\mathcal{U}$ verifies the transactions only then the block will be added into the chain. User access defines how a $\mathcal{U}$ monitors his data transaction. $\mathcal{BC}$ sends all the hashes into the $\mathcal{C}$. In $\mathcal{C}$, there will be 2 hashes against a $\mathcal{R}_i$. By using these hashes, a $\mathcal{U}$ can monitor his data transactions.

e) ***User*** *($\mathcal{U}$):* $\mathcal{U}$ is an individual who has been successfully registered in our proposed architecture with her IoT devices. $\mathcal{U}$ can keep track of how her data is being used in this architecture by using hash generated from $\mathcal{BC}$ transaction in all steps. It also monitors the computed data from $\mathcal{C}$. If $\mathcal{U}$ allows a Healthcare Monitoring Unit only then monitor $\mathcal{U}$ health condition. $\mathcal{U}$ can view the computed result and the health condition result based on the computed data (e.g., if the computed data from a pulse sensor pulse rate around 70, then it will show only the result as, normal or good heart condition.)

f) ***Health monitoring Unit*** *($\mathcal{HM}$):* $\mathcal{HM}$ has doctors or health professionals who can monitor the computed data from

[1] Adding multiple transactions in a block is possible in Ethereum Network.

$\mathcal{C}$ only if $\mathcal{U}$ allows $\mathcal{HM}$ to check her health condition. $\mathcal{HM}$ also offers the health advice to the $\mathcal{U}$.

B. Architecture Analysis

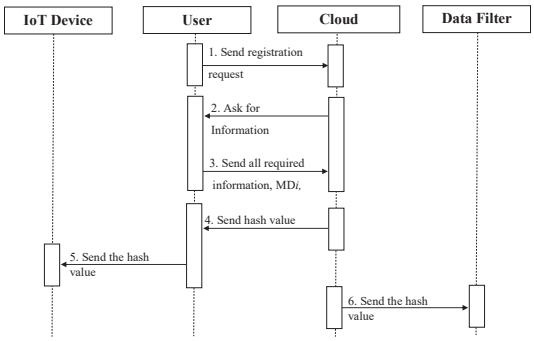

Fig. 2: Registration process of the user and her device

The working procedure of the proposed architecture is divided into four parts; i. IoT layer ii. Edge node, $\mathcal{N}$ iii. Cloud, $\mathcal{C}$ and iv. Blockchain, $\mathcal{BC}$. Data filtering process and the computation take place on the edge node. $\mathcal{BC}$ stores data transaction records and send the hashes to the $\mathcal{C}$. $\mathcal{C}$ stores the computed result, hashes and user information. $\mathcal{HM}$ may connect to this architecture via the $\mathcal{C}$.

1) ***User registration and device authorization****:* Fig. 2 demonstrates a simplified overview of the $\mathcal{U}$ registration process. Before sending data into the $\mathcal{N}$, all $\mathcal{D}_\text{i}$ needs to be registered. The $\mathcal{D}_\text{i}$'s registration is done by the $\mathcal{U}$ during registration process. The $\mathcal{U}$ sends the required information including the MAC address ($\mathcal{MD}_\text{i}$) of the $\mathcal{D}_\text{i}$ to the $\mathcal{C}$. The $\mathcal{C}$ generates a private key ($\mathcal{K}$) to calculate the hash value ($\mathcal{H}$) of the $\mathcal{D}_\text{i}$. $\mathcal{H}$ is calculated by using the SHA-256 algorithm. $\mathcal{C}$ transmits the $\mathcal{H}$ to the $\mathcal{N}$ and generates a device-specific account. $\mathcal{U}$ may register multiple $\mathcal{D}_\text{i}$s in the architecture. After registering a $\mathcal{D}_\text{i}$, the cloud will create an account concerning the $\mathcal{MD}_\text{i}$ and all the data coming from this device will be stored in this account after the computation. The account concerning on the $\mathcal{MD}_\text{i}$ is called the device-specific account.

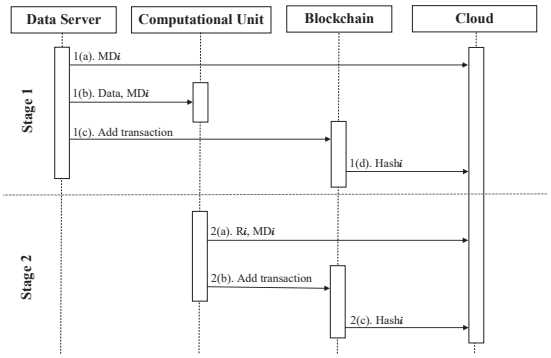

Fig. 3: Blockchain transaction in the architecture

2) ***Data filtering and device authentication****:* Data filtering takes place at the network edge. The $\mathcal{F}$ begins the process of receiving and filtering the data after the authentication process. The $\mathcal{F}$ gets information from various $\mathcal{D}_\text{i}$ registered with our proposed architecture. $\mathcal{F}$ does the authentication by checking whether the $\mathcal{H}'$ is matched with one of the stored $\mathcal{H}$ only then the $\mathcal{F}$ starts receiving data from the $\mathcal{D}_\text{i}$.

$$\mathcal{H}' \in \mathcal{H}^{\text{stored}}$$

The $\mathcal{F}$ removes unwanted data. As an instance, if any $\mathcal{D}_\text{i}$ that can measure the human heartbeats may raise some difficulties and could generate some value that is not relevant to the heart rate of humans (e.g., 500 bpm). In this case, as $\mathcal{D}_\text{i}$ generates irrelevant values, the $\mathcal{F}$ will not receive the data and it will send a notification to the $\mathcal{D}_\text{i}$.

3) ***Data transaction between*** $\mathcal{S}$ ***and*** $\mathcal{CU}$*:* $\mathcal{S}$ stores the filtered data that are coming from the $\mathcal{F}$. In this architecture, only $\mathcal{S}$ stores the data and works as a data provider to the $\mathcal{CU}$. Stage 1 of Fig. 3 shows the simplified overview of data transaction between $\mathcal{S}$ and the $\mathcal{CU}$. Whenever $\mathcal{S}$ sends the data to $\mathcal{CU}$, in the meantime $\mathcal{S}$ also sends the $\mathcal{MD}_\text{i}$ to the $\mathcal{C}$ and $\mathcal{C}$ starts waiting for receiving a hash from $\mathcal{BC}$ transaction. The data transfer between $\mathcal{S}$ and $\mathcal{CU}$ is recorded on the $\mathcal{BC}$ as a transaction, and $\mathcal{BC}$ sends the hash to the $\mathcal{C}$. Whenever the $\mathcal{C}$ receives the hash a data transaction from $\mathcal{S}$ to $\mathcal{CU}$ is completed. $\mathcal{C}$ stores the hash in the $\mathcal{U}$'s account.

4) ***Data transaction between*** $\mathcal{CU}$ ***and*** $\mathcal{C}$ *:* Stage 2 of Fig. 3 shows the data transaction between the $\mathcal{CU}$ and the $\mathcal{C}$. The $\mathcal{S}$ provides the raw data to the $\mathcal{CU}$ to calculate the result. After the computation, the $\mathcal{CU}$ sends the result to the $\mathcal{C}$. Whenever $\mathcal{CU}$ sends a data to the $\mathcal{C}$, it also sends $\mathcal{MD}_\text{i}$ to the $\mathcal{C}$. After receiving the $\mathcal{MD}_\text{i}$, $\mathcal{C}$ waits for receiving a hash from the $\mathcal{BC}$. Data transaction from the $\mathcal{CU}$ to $\mathcal{C}$ is recorded on the $\mathcal{BC}$ as a transaction and $\mathcal{BC}$ sends the hash to the $\mathcal{C}$. Whenever $\mathcal{C}$ receives the result, $\mathcal{MD}_\text{i}$, and hash then the data transaction from $\mathcal{CU}$ to $\mathcal{C}$ completes.

5) ***Data storing at the*** $\mathcal{C}$*:* $\mathcal{C}$ stores the $\mathcal{U}$ information during the registration process. Whenever $\mathcal{C}$ receives the hashes and the $\mathcal{MD}_\text{i}$ from the $\mathcal{N}_\text{i}$, it stores all information in the $\mathcal{U}$'s account. $\mathcal{U}$ account is identified by the $\mathcal{MD}_\text{i}$ as $\mathcal{C}$ has created a device-specific account.

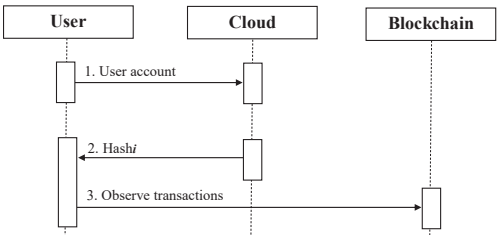

Fig. 4: User's transaction observation or audit process

6) ***Data access monitoring by the user****:* One of the important functionalities of the architecture is the process through which a user can track the transaction of her data. Fig. 4 shows how a $\mathcal{U}$ can monitor her data transaction. Every transaction of data has a record on the chain. A new hash is generated while generating a new block on the chain, which is sent it to the $\mathcal{C}$. The hash regarding the user account is

stored in the $\mathcal{C}$. A user merely monitors the data transactions using these hashes.

Suppose, $\mathcal{U}$ wants to verify whether her data is computed or not. If there is no hash generated by the $\mathcal{BC}$, the data is still on the $\mathcal{S}$. That means data is not sent to $\mathcal{CU}$ yet. If there is only one hash, that means the data is sent to $\mathcal{CU}$. If there are two hashes then $\mathcal{U}$ will be assured that the data is sent to the $\mathcal{C}$ after $\mathcal{CU}$.

V. Protocol Construction

A. Algorithm for device authentication and data filtering

Algorithm 1: Device authentication and data Filtering Algorithm

Input: δ, MA_{i}, $\mathcal{H}'$, $\mathcal{H}^{\text{stored}}$
Output: ${\delta_{req}}^{\text{successful}}$, δ^{valid}

1: **if** $(\mathcal{H}' \in \mathcal{H}^{stored})$ **then**
2: $\quad \delta_{req} \longleftarrow {\delta_{req}}^{\text{successful}}$
3: $\quad$ **if** $\delta \in [a, b]$ **then**
4: $\quad\quad \delta \longleftarrow \delta^{\text{valid}}$
5: $\quad\quad \mathcal{F} \xrightarrow{\delta, \mathcal{MA}_{\text{i}}} \mathcal{S}$
6: $\quad$ **else**
7: $\quad\quad \delta \longleftarrow \delta^{\text{invalid}}$, Send notification to the device
8: $\quad$ **end if**
9: **else**
10: $\quad \delta_{req} \longleftarrow {\delta_{req}}^{\text{unsuccessful}}$
$\quad \{\delta_{req}$ from unregistered device$\}$
11: **end if**

B. Algorithm for transaction adding on blockchain

Algorithm 2: Transaction adding algorithm on blockchain

Input: δ, $\mathcal{MD}_{\text{i}}$
Output: $\mathcal{R}_{\text{i}}$

1: $\mathcal{F} \xrightarrow{\delta^i, \mathcal{MA}_{\text{i}}} \mathcal{S}$
2: **while** $\beta \neq 0$ **do**
3: $\quad \mathcal{S} \xrightarrow{\mathcal{MA}_{\text{i}}, \delta} \mathcal{CU}$
4: $\quad \mathcal{S} \xrightarrow{\delta^i} \mathcal{CU}$
5: $\quad \mathcal{S} \xrightarrow{transaction} \mathcal{BC}$
6: $\quad \mathcal{BC} \xrightarrow{hash} \mathcal{C}$
7: $\quad \beta = 0$
8: **end while**
9: **if** $\beta = 0$ **then**
10: $\quad \mathcal{CU} \xrightarrow{generate} \mathcal{R}_{\text{i}}$
$\quad \{\mathcal{CU}$ will compute the result, $\mathcal{R}_{\text{i}}\}$
11: $\quad \mathcal{CU} \xrightarrow{\mathcal{MA}_{\text{i}}, \mathcal{R}_{\text{i}}} \mathcal{C}$
12: $\quad \mathcal{CU} \xrightarrow{transaction} \mathcal{BC}$
13: $\quad \mathcal{BC} \xrightarrow{hash} \mathcal{C}$
14: **end if**

VI. Security Analysis

In this section, we have analyzed our protocol in terms of security parameters.

Authorization: The $\mathcal{C}$ performs the device authorization. No $\mathcal{D}_{\text{i}}$ other than the registered $\mathcal{D}_{\text{i}}$ can send δ_{req} to the $\mathcal{N}_{\text{i}}$. Every $\mathcal{D}_{\text{i}}$ needs a $\mathcal{H}$ to send the data to the $\mathcal{N}$. A $\mathcal{D}_{\text{i}}$ will get the $\mathcal{H}$ after the registration. Hence, our suggested architecture guarantees that no unregistered $\mathcal{D}_{\text{i}}$ or $\mathcal{U}$ can be connected to our architecture.

Authentication: The $\mathcal{N}_{\text{i}}$ performs the $\mathcal{D}_{\text{i}}$ authentication. The authentication method begins at the $\mathcal{F}$ by checking whether or not the $\mathcal{H}$ is stored. The $\mathcal{C}$ uses a $\mathcal{K}$ to generate the $\mathcal{H}$ so, there is no chance for the third party to generate the hash. Only the ${\delta_{\text{req}}}^{\text{successful}}$ will happen if the $\mathcal{H}$ coming from the $\mathcal{D}_{\text{i}}$ is stored at the $\mathcal{F}$.

Integrity: Data integrity is maintained as our architecture keeps the data transaction records into the $\mathcal{BC}$. A user will receive two hash values that correspond to one calculated result. If data is shared for computation or any other activities, surely a transaction will be added on $\mathcal{BC}$. Unauthorized party cannot use the data for unauthorized activities and cannot alter the transaction record as $\mathcal{BC}$ is immutable. Immutability refers the ability of $\mathcal{BC}$ ledger to remain unchanged. So, there is no chance of data theft for unauthorized activities.

Anonymity: Our architecture provides anonymity by maintaining the $\mathcal{U}$ unknown to the $\mathcal{N}_{\text{i}}$. $\mathcal{U}$ are anonymous at the $\mathcal{N}_{\text{i}}$ as $\mathcal{U}$ information are stored at the $\mathcal{C}$. $\mathcal{N}_{\text{i}}$ only receive the δ and $\mathcal{MD}_{\text{i}}$, but cannot identify the $\mathcal{U}$ itself.

Privacy: Many registered $\mathcal{D}_{\text{i}}$s can make a connection with the $\mathcal{N}_{\text{i}}$, here comes the issue of $\mathcal{U}$ data privacy. Privacy is maintained as the $\mathcal{U}$ is anonymous to the $\mathcal{N}_{\text{i}}$. The $\mathcal{N}_{\text{i}}$ only receives the data coming from the $\mathcal{D}_{\text{i}}$ but does not have any clue who is the owner of the $\mathcal{D}_{\text{i}}$. Therefore, $\mathcal{N}_{\text{i}}$ cannot identify a particular $\mathcal{U}$ from the $\mathcal{MD}_{\text{i}}$ and δ.

Security: Security is provided by keeping the $\mathcal{U}$ information into the $\mathcal{C}$. $\mathcal{U}$ personal data as well as the computed result are secure as no $\mathcal{HM}$ can view $\mathcal{U}$ information without $\mathcal{U}$ permission.

Verifiability: In the architecture, $\mathcal{U}$ can monitor data transactions by using the hashes from $\mathcal{BC}$ transaction. If more than two hashes are corresponding to one $\mathcal{R}_{\text{i}}$ then the $\mathcal{U}$ can claim that her data has been shared with unauthorized parties. However, it will never happen because $\mathcal{U}$ will verify their data transactions before adding the new block into chain.

VII. Experimental Evaluation

Empirical overview of our architecture is shown through graphs with description and also a comparison with related works has been included here.

A. Experimental Setup

We setup an environment to evaluate our protocol by using a computer computer Intel(R) Core(TM) i5-7200U 2.5GHz, 8GB of RAM, Win10 (64-bit) OS.

In our evaluation we write the programs by using languages:

Solidity, Web3.js, HTML and CSS. Software: browser, sublime text, Remix-Ethereum IDE to write the smart contact in solidity language and EthereumJS TestRPC to construct a simulated Ethereum network locally. Wi-Fi connection is required in the setup.

B. Computation and Evaluation

Fig. 5 shows an overview of cost in terms of Gwei (Gwei is a unit of Ether) to create a new block. Each transaction needs an amount of gas to be successfully completed and the same amount of Gwei is deducted from each account as cost for creating a new block. Ten fake accounts are given in the ethereum test network with 100ETH.

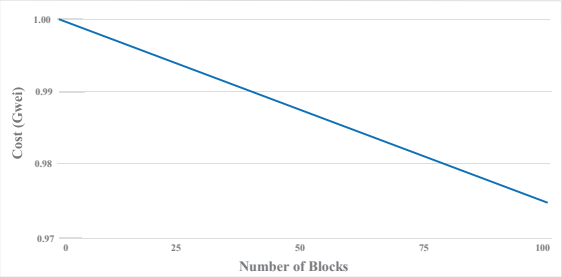

Fig. 5: Cost to create a new block.

C. Comparison between our architecture and the related works

In this subsection, we show the results in the nine certain properties/metrics of the architectures, if a particular architecture has the property in it then we marked it with 'Y' otherwise marked with 'N'. Here, table II compares our architecture to other existing architectures. With the careful analogy of the systems and in account of the important nine properties/metrics of Edge computing based healthcare systems, a conclusion can be drawn that our architecture has greater advantages than the other systems in this table.

TABLE II: Property comparison between proposed architecture and different related systems

Metric	[7]	[6]	[8]	Our architecture
Data filtering before uploading to the cloud	Y	Y	N	Y
Computation on the Edge	Y	N	N	Y
Accountability of the data transactions	N	N	Y	Y
User centric	N	Y	N	Y
Privacy of data owner	N	Y	Y	Y
Anonymity of the user	N	N	N	Y
IoT implementation in the platform	Y	Y	Y	Y
Use of cryptographic functions in the core platform	N	Y	Y	Y
blockchain-Based	N	N	Y	Y

VIII. Conclusion

In this paper, we propose a user-centric and secure edge computing architecture for healthcare IoT devices using blockchain. Blockchain is used to store the record of the data transaction instead of storing the data. The most important aspect of our architecture is the control of users, which allows every user to monitor their data by herself. The user can permit the healthcare professional to monitor their health condition. Our architecture thus provides a secure and user-centric edge computing architecture for healthcare IoT devices. Rigorous security and experimental analysis show that the approach is fit for the study.

References

[1] P. Verma and S. K. Sood, "Fog assisted-iot enabled patient health monitoring in smart homes," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1789–1796, 2018.

[2] M. S. Rahman, A. Al Omar, M. Z. A. Bhuiyan, A. Basu, S. Kiyomoto, and G. Wang, "Accountable cross-border data sharing using blockchain under relaxed trust assumption," *IEEE Transactions on Engineering Management*, 2020.

[3] X. Zhang, P. Huang, L. Guo, and M. Sha, "Incentivizing relay participation for securing iot communication," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 1504–1512.

[4] X. Zheng, R. R. Mukkamala, R. Vatrapu, and J. Ordieres-Mere, "Blockchain-based personal health data sharing system using cloud storage," in *2018 IEEE 20th International Conference on e-Health Networking, Applications and Services (Healthcom)*. IEEE, 2018, pp. 1–6.

[5] W. Yu, F. Liang, X. He, W. G. Hatcher, C. Lu, J. Lin, and X. Yang, "A survey on the edge computing for the internet of things," *IEEE access*, vol. 6, pp. 6900–6919, 2017.

[6] S. Pirbhulal, O. W. Samuel, W. Wu, A. K. Sangaiah, and G. Li, "A joint resource-aware and medical data security framework for wearable healthcare systems," *Future Generation Computer Systems*, vol. 95, pp. 382–391, 2019.

[7] A. M. Rahmani, T. N. Gia, B. Negash, A. Anzanpour, I. Azimi, M. Jiang, and P. Liljeberg, "Exploiting smart e-health gateways at the edge of healthcare internet-of-things: A fog computing approach," *Future Generation Computer Systems*, vol. 78, pp. 641–658, 2018.

[8] H. L. Pham, T. H. Tran, and Y. Nakashima, "A secure remote healthcare system for hospital using blockchain smart contract," in *2018 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2018, pp. 1–6.

[9] V. Ramani, T. Kumar, A. Bracken, M. Liyanage, and M. Ylianttila, "Secure and efficient data accessibility in blockchain based healthcare systems," in *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2018, pp. 206–212.

[10] A. Alrawais, A. Alhothaily, C. Hu, and X. Cheng, "Fog computing for the internet of things: Security and privacy issues," *IEEE Internet Computing*, vol. 21, no. 2, pp. 34–42, 2017.

[11] R. Roman, J. Lopez, and M. Mambo, "Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges," *Future Generation Computer Systems*, vol. 78, pp. 680–698, 2018.

[12] S. Khan, S. Parkinson, and Y. Qin, "Fog computing security: a review of current applications and security solutions," *Journal of Cloud Computing*, vol. 6, no. 1, p. 19, 2017.

[13] H. A. Al Hamid, S. M. M. Rahman, M. S. Hossain, A. Almogren, and A. Alamri, "A security model for preserving the privacy of medical big data in a healthcare cloud using a fog computing facility with pairing-based cryptography," *IEEE Access*, vol. 5, pp. 22 313–22 328, 2017.

[14] A. Al Omar, M. S. Rahman, A. Basu, and S. Kiyomoto, "Medibchain: A blockchain based privacy preserving platform for healthcare data," in *International conference on security, privacy and anonymity in computation, communication and storage*. Springer, 2017, pp. 534–543.

[15] A. Al Omar, M. Z. A. Bhuiyan, A. Basu, S. Kiyomoto, and M. S. Rahman, "Privacy-friendly platform for healthcare data in cloud based on blockchain environment," *Future Generation Computer Systems*, vol. 95, pp. 511–521, 2019.

[16] C. Pahl, N. El Ioini, and S. Helmer, "A decision framework for blockchain platforms for iot and edge computing." in *IoTBDS*, 2018, pp. 105–113.