Accepted Manuscript

Privacy-friendly platform for healthcare data in cloud based on blockchain environment

Abdullah Al Omar, Md Zakirul Alam Bhuiyan, Anirban Basu,
Shinsaku Kiyomoto, Mohammad Shahriar Rahman

PII: S0167-739X(18)31420-1
DOI: https://doi.org/10.1016/j.future.2018.12.044
Reference: FUTURE 4665

To appear in: *Future Generation Computer Systems*

Received date : 20 June 2018
Revised date : 25 August 2018
Accepted date : 17 December 2018

Please cite this article as: A. Al Omar, M.Z.A. Bhuiyan, A. Basu et al., Privacy-friendly platform for healthcare data in cloud based on blockchain environment, *Future Generation Computer Systems* (2019), https://doi.org/10.1016/j.future.2018.12.044

Privacy-friendly Platform for Healthcare Data in Cloud Based on Blockchain Environment

Abdullah Al Omar[a], Md Zakirul Alam Bhuiyan[b,*], Anirban Basu[c,**], Shinsaku Kiyomoto[d], Mohammad Shahriar Rahman[e,*]

[a]Department of Computer Science and Engineering, University of Asia Pacific, Dhaka, Bangladesh
[b]Department of Computer and Information Sciences, Fordham University, New York, NY 10458, USA
[c]University of Sussex, UK
[d]Information Security Laboratory, KDDI Research, Saitama, Japan
[e]Department of Computer Science and Engineering, University of Liberal Arts Bangladesh, Dhaka, Bangladesh

Abstract

Data in cloud has always been a point of attraction for the cyber attackers. Nowadays healthcare data in cloud has become their new interest. Attacks on these healthcare data can result in annihilating consequences for the healthcare organizations. Decentralization of these cloud data can minimize the effect of attacks. Storing and running computation on sensitive private healthcare data in cloud are possible by decentralization which is enabled by peer to peer (P2P) network. By leveraging the decentralized or distributed property, blockchain technology ensures the accountability and integrity. Different solutions have been proposed to control the effect of attacks using decentralized approach but these solutions

☆A preliminary version of this paper appears in The 10th International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage, SpaCCS Workshops 2017. This is the full version.

*Corresponding authors

**Anirban Basu currently works for Hitachi R&D. The views, opinions and/or findings contained in this article are those of the author(s) and should not be interpreted as an official Hitachi position, policy or decision, unless so designated by other documentation.

Email addresses: omar.cs@uap-bd.edu (Abdullah Al Omar), mbhuiyan3@fordham.edu (Md Zakirul Alam Bhuiyan), a.basu@sussex.ac.uk (Anirban Basu), kiyomoto@kddi-research.jp (Shinsaku Kiyomoto), shahriar.rahman@ulab.edu.bd (Mohammad Shahriar Rahman)

somehow failed to ensure overall privacy of patient centric systems. In this paper, we present a patient centric healthcare data management system using blockchain technology as storage which helps to attain privacy. Cryptographic functions are used to encrypt patients data and to ensure pseudonymity. We analyze the data processing procedures and also the cost effectiveness of the smart contracts used in our system.

Keywords: Blockchain, Decentralization, Healthcare data in cloud, Pseudonymity, Privacy, Security, Smart contract

---

1. Introduction

A lot of work is going on healthcare and information technology in an amalgamated manner and these works are bringing a lot of changes in healthcare discipline. These changes are affecting patients' treatment process hence requiring careful data processing. For treatment, healthcare is completely dependent on data which arises some concerns over data security and privacy. Authorization or private access to the personal data of individual patient refers to the term Privacy, which means only authenticated parties will be able to access the private data. Keeping these personal data safe from the eavesdroppers or intruders refers to the term Security, which means system will be able to protect users' private data from outsiders. Authenticated parties of healthcare data preservation process will get

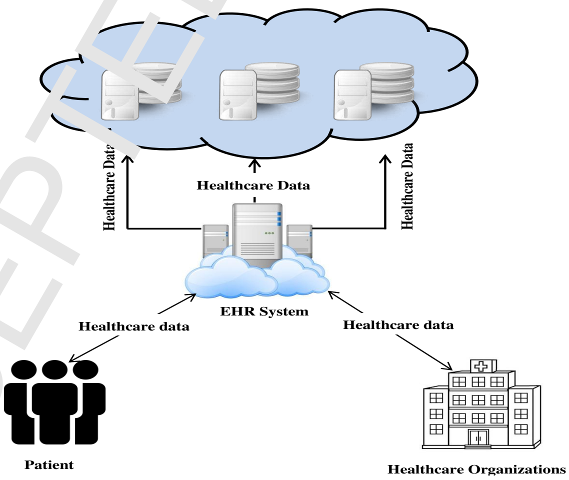


Figure 1: Entities of EHR system and it's Data flow

the access to store data into cloud and retrieve from it. Interaction between the system and the patient requires a secured channel. Different authentication protocol [20, 19, 24] have been proposed to preserve the privacy and security. Lack of security may result in devastating consequences like data loss and data theft. A lot of intruders are searching for an insecured channel and trying to access valuable healthcare data in the cloud network. In most of the cases, data loss in healthcare causes detrimental consequences to the patients and healthcare organizations. Due to recent attacks on healthcare data in cloud systems, different countries like USA [8] and UK [12] have experienced critical data loss. Personal data of patients' were kept without encryption in the cloud which allowed the attackers to steal the sensitive private data. Let's assume a scenario where patients keep their data in any Electronic Health Record (EHR) system [35, 7, 14, 13, 38, 5] for preservation and also for further access. Figure 1 depicts a generalized formation of EHR systems. In the figure patients and healthcare organizations take part in the process as both data sender and data receiver. EHR system is the manager of the whole process that maintains the data flow of the system. Top most entity is the cloud where data is kept. Patients share their personal data with the doctors and healthcare organizations with the help of these EHR systems. Suppose, a patient keeps her data in the cloud system [7] which uses blockchain as a data storage platform. System will store the data on blockchain when the patient shares her data with the system. Accountability of data is system-centric in case of the instance [7], whereby the system will provide data storage service even when data is shared with the doctors or healthcare organizations. Consequently, the system is responsible for data loss.

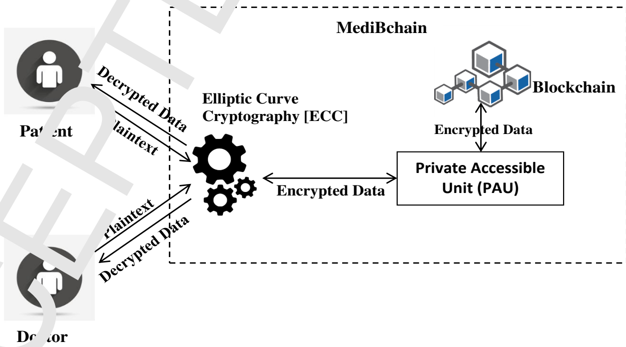


Figure 2: An application of MediBchain

Figure 2[1] depicts the design of our platform in which aforementioned problems have been addressed by storing the encrypted healthcare data in the cloud system. As a result, if our system somehow loses the control over blockchain, patients will be accountable for their data as they will control the encryption keys solely. Data sharing in our system is also being controlled by the patients. Vulnerabilities related to data preservation have been addressed in our system by using cryptographic functions along with blockchain technology. However, our system will store the encrypted personal data ensuring overall privacy of the data such that even if system gets attacked by the attacker the stolen data will make no sense to them. To get the plaintext of those encrypted personal data, attackers will require the keys. There is no identifier for these datasets, only encryption keys will be used to identify such encrypted and pseudonymous[2] data.

1.1. Our Contribution

Our platform ensures that the private healthcare data in cloud is controlled by only patient herself. The main idea of this work is to keep the sensitive healthcare data on the blockchain to attain accountability, integrity and security. Patients will have the overall control over the blocks in which their data will be stored. Present healthcare systems lack in pseudonymity as those only store the data in cloud, but our platform ensures the pseudonymity of patients. We achieve pseudonymity by using cryptographic functions. MediBchain will regain the interest of patients on EHR systems and will retain accountability, integrity, pseudonymity, security and privacy which are being lost with the increasing computational power of emerging technologies in EHR systems[3]. Analysis of these attributes is discussed in section 3. Our contributions are as follows:

1. **Security and privacy guarantee:** The proposed platform guarantees accountability, pseudonymity, authenticity and integrity along with data privacy.
2. **Analysis:** Rigorous analysis on security, privacy, accountability, pseudonymity and integrity shows how our platform achieves the above mentioned properties.

[1] Private Accessible Unit (PAU) is the intermediary unit between blockchain and data sender or receiver.

[2] Pseudonymity refers to the fact of using disguised identity.

[3] Analysis of security terminologies are given in section 5.

3. **Evaluation:** We have implemented smart contract and shown different analogies of costs (e.g., transaction cost, execution cost). Then we have evaluated a java implementation of input and output generation algorithm using Elliptic Curve Cryptography (ECC) for our system. Experimental results will help to compare several aspects of EHR system and will help to decide whether accept our platform or not.

Organization of the paper: The remainder of the paper is organized as follows: Section 2 describes the related work.In section 3 we discuss the preliminaries. In section 4, we describe our platform. In Section 5, we evaluate the platform and analyze it formally. We give some concluding remarks in Section 6.

2. Related Work

Some national level frameworks based on cloud for electronic medical system have been proposed in [14, 13, 25]. Patra et al. [25] proposed a model which is cloud-based and deals with patients' private data. This model ensures cost effectiveness, and this system was designed for rural areas where cost plays an immense role. Medical professionals and policy makers could serve the patients remotely through a cloud-based model which stores all the imperative data in a single cloud. Patients were encouraged to share their data in the cloud so that they could get the medical service from the professionals remotely. Disease diagnosis and control could be made by this remote treatment. Data collection and data delivery are the key points in symptom analysis. Rolim et al. [27] proposed a framework where the system processes data in the steps of data collection and data delivery. In this model, sensors play the role of collector which collects the data and sends directly to the system to store and work with this data further. These data would be accessed by the medical professionals and sensors were proposed to be attached with the medical equipment in this system. Yin et al. [37] introduced cloud based patient centric system. This model includes three layers: data collection layer, data management layer and data service layer. [21] described a blockchain based access control manager for heath data to enhance the interoperability of this system. Off blockchain mechanism with the involvement of public blockchain was proposed as an access control manager of healthcare data.

Controllability and Traceability are two key topics of privacy preserving systems. Xiao et al. [35] proposed a model which is based on blockchain to help patients to own, control and share their personal data easily and securely with privacy preservation. This application based model also deals with Secure Multi-party Computing (MPC) and Indicator-Centric Schema (ICS). Simic et al. [29] showed a case study where the study concludes with the illustration of significant benefits of IoT and blockchain in a combined manner. In their work IoT devices were proposed to be used as collectors of private health data of the patients', and real time data of patient could be saved in blockchain. Scalability of the blockchain in case of Big data has also been tested in their study. Ekblaw et al. [7] proposed a prototype named 'MedRec' which uses blockchain as a backbone and tried to find the security solutions for EHR systems. They tried to give their prototype-integrity, authenticity, auditability and data sharing through blockchain. Elements of their system are: Registrar Contract (RC), Patient-Provider Relationship Contract (PPR), Summary Contract (SC), where RC maps the identification strings of the participants to their Ethereum addresses, PRR issues contracts between two nodes in the system when one node stores and manages medical records for the other, SC locates the participants medical record history. Jun et al. [3] proposed a web-based architecture where they showed a secured accessing multiple patient repository system. They concentrated mainly on lifetime repository of health data, which consists of client application (CA), central access-control (CAC), local access-control (LAC) and Hospital information system. Linn et al. [21] described a blockchain based access control manager for health data to enhance the interoperability of this system

The backbone of our work is blockchain. Blockchain technology is popular for its application in Bitcoin cryptocurrency [26], which is a public ledger to hold and maintain the transactional data and integrity [31]. One of the reasons for using blockchain technology in cryptocurrency is its decentralized digital ledger property, which was presented by Nakamoto [22] in his Bitcoin cryptocurrency framework. Blockchain's data structure has been modeled by blocks which is linearly sequenced. Each block contains the cryptographic hashes corresponding to the previous and current block to ensure continuity and immutability of the chain. Chaining mechanism ensures integrity of this secured data structure.

2.1. Blockchain.

Figure 3 exhibits the structure of blocks in the blockchain network. In the figure each block is connected to its previous block by the hash of previous block. Blocks store the time-stamp of being mined in the network. Mining takes place in the network by solving mathematically complex problems. Miners compete each

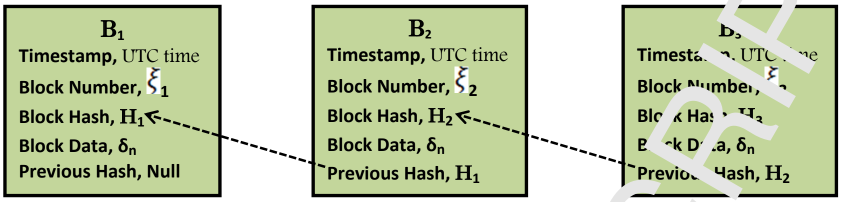


Figure 3: Structure of Blocks in blockchain

other to mine the block so that they could earn some cryptocurrency. In our platform miners will get Ether from Ethereum Network for mining, and our platforms Ethereum account will be charged against it. Simple Ether transfer functionality will be used to transfer the Ether from our account. Each block contains corresponding block number and data that has been given to store in the blockchain which has been denoted as δ_n.

Blockchain-secured transaction-based technology [1] gives the users a better security. Bitcoin as well as blockchain has not been failed since these were introduced [6]. The network is shared and information is stored throughout the whole network, thus increasing the reliability of this technology. All the information is treated in a redundant way in blockchain [28]. Blockchain is distributed but it remains all the same for it's nodes ensuring the integrity [4, 34]. Centralized database can be corrupted and needs a third party to maintain it. To change the history of the blockchain any individual has to control at least 51% of the chain and it will cost a lot to challenge the immutability of blockchain. This immutable architecture [2, 30, 32] is a blessing in archival science too. Identities in the blockchain are covered by pseudonyms by which privacy for the participants is ensured with a very high degree [15]. Cryptographic authentication of the time blocks with time-stamp allows the entire network to hold the logs for any interaction in the blockchain. Blockchain ensures the verifiability of the users. Other than above discussed characteristics some author explicitly mention the key points like trust enabling notion [1, 27, 11, 33] , Consensus, Transparency, Smart contract etc.

Blockchain gives a distribution oriented service to be used as a storage. All the records that may be stored in the blockchain have to use smart contracts[16, 9]. Smart contracts determine the record of data and conditions in the blockchain. These contracts, as a form of code, give a huge power to the programmers to read and write over the blockchain [9]. As storage, blockchain provides accuracy and reliability to it's users and protects the data from fraud and being tampered or corrupted [18]. Blockchain as storage maintains proper decentralization and true redundancy, total privacy and cost reduction [10]. Decentralized web will be the future of this era[36].

3. Preliminaries

In this Section, we explain each properties (e.g., security, privacy and management) that our protocol achieves. Finally, we introduce the building blocks of our protocol.

3.1. Properties

3.1.1. Security and Privacy

We briefly describe each of the security and privacy properties in the context of our system below.

1. Pseudonymity: No entity will be able to identify any party of our system because users are being identified by a dynamic key. As a result users are keeping their selves pseudonymous [4]. Data will not be identified by just seeing it.

2. Privacy: Only registered parties will be able to interact with the system. Even a registered party will not be able to access the private raw data of other parties.

3. Integrity: Authenticated parties will be able to store private data.

4. Accountability: Each block will be identified by corresponding block-id. Only authenticated parties will get them and will interact with them.

5. Security: Parties will keep their encrypted data in the system which ensures secured environment for them.

3.1.2. Management

- Users need to register once and by providing the ID and PWD [5] they can easily get into the platform.

[4]Pseudonymity and anonymity are two different things. Anonymity refers to the fact of being unknown; in our system users are identified with dynamic keys, hence users are pseudonymous.

[5]ID and PWD are described in Table 1.

- PAU will act as a Trusted Third Party (**TTP**) of our system, as it will be the medium between user and blockchain.
- In the case of Block id sharing, users need to be very careful because untrusted access will make the platform vulnerable for that particular user's data.

3.2. Cryptographic tools

Here, we describe Elliptic Curve Cryptography (ECC) [17] which has been used as the cryptographic tool to provide proper cryptographic functionality to the users. Formal definition of ECC will be given here.

Definition 1 (Elliptic Curve Cryptography) Elliptic Curve Cryptographic scheme use the trapdoor function which means if we compute B from A through trapdoor function then it is mathematically infeasible to regenerate A from B.

$$A \xrightarrow{\texttt{trapdoor}} B$$
$$A \nleftarrow B$$

All the functional properties of ECC are described:
Encryption Scheme:
Choose, Elliptic group $\mathbb{E}_p(a, b)$ and generator point, $\mathbb{G} \in \mathbb{E}_p(a, b)$ such that the smallest value of n for that n$\mathbb{G}$ = 0 is a very large prime number.
Message, $\mathcal{M}$ is encoded in to point $P_M \in \mathbb{E}_p(a, b)$
Both sender and receiver selects a private key, $n_A < n$
compute public key P_A, such that $P_A = n_A\mathbb{G}$

$$\text{Ciphertext point, } P_C = [(\mathcal{K}\mathbb{G}), (P_M + \mathcal{K}P_B)]$$

($\mathcal{K}$ is the random integer and P_B is the public key of receiver here).
Decryption Scheme:

$$\text{Plaintext point, } P_M \longleftarrow (P_M + \mathcal{K}n_B\mathbb{G}) \longleftarrow P_M + \mathcal{K}P_B$$

only receiver knowing private key n_B will retrieve this point, P_M by removing $n_B\mathcal{K}\mathbb{G}$.

4. MediBchain Protocol

In this section we present the architectural as well as the design view of our platform. **Table 1.** describes the notations that are used in the next sections.

Table 1: Terminology table

Notation	Description
ID	ID of the User
PWD	Password of the user
U_D	Encrypted user data
U_{id}	Block id, where user data will be saved
ID_X	ID of the User X
PWD_X	Password of the user X
U_{DX}	User X's Encrypted data
U_{idX}	Block number, where user X's data is saved
Secured channel	Obtained by the authentications process of our system
$\mathcal{T}(\delta_n)$	Transaction of δ_n through **smart contract**
$\mathcal{H}_{\mathcal{M}}$	Set of all identical hashes
Γ	Address of the issuer
ν	Address of the message sender
δ_n	Number of categories in the smart contract
$\{\mathcal{S},\mathcal{R}\}_{\texttt{authenticated}}$	authenticated sender, $\mathcal{S}$ and receiver, $\mathcal{R}$
$\{\mathcal{S},\mathcal{R}\}_{\text{~~authenticated~~}}$	Unauthenticated Partied, $\mathcal{S}$ and $\mathcal{R}$
$\mathbb{B}_i$, ξ_i & $\mathbb{H}_i$	Property of different blocks

4.1. Overview of Our Protocol

Fig. 4. shows the high level view of our platform. The following entities and their roles are described briefly here.

Data sender is the patient, who will send her personal healthcare data to the system. Data sender plays the vital role in case of data preservation. Data that will be sent to the system must be accurate otherwise wrong data will be detrimental for patient because the whole treatment depends on this sensitive data. However, our system will take the encrypted data from the users. Encryption of data will be done in the very beginning of MediBchain's process execution.

Data receiver will request for the data after authenticating itself to the system.

Registration Unit will act as an authenticator. When any party (Sender or Receiver) will come for the first time to take the service of the system; it will store their ID and PWD to be used further. Each party will have to register for once

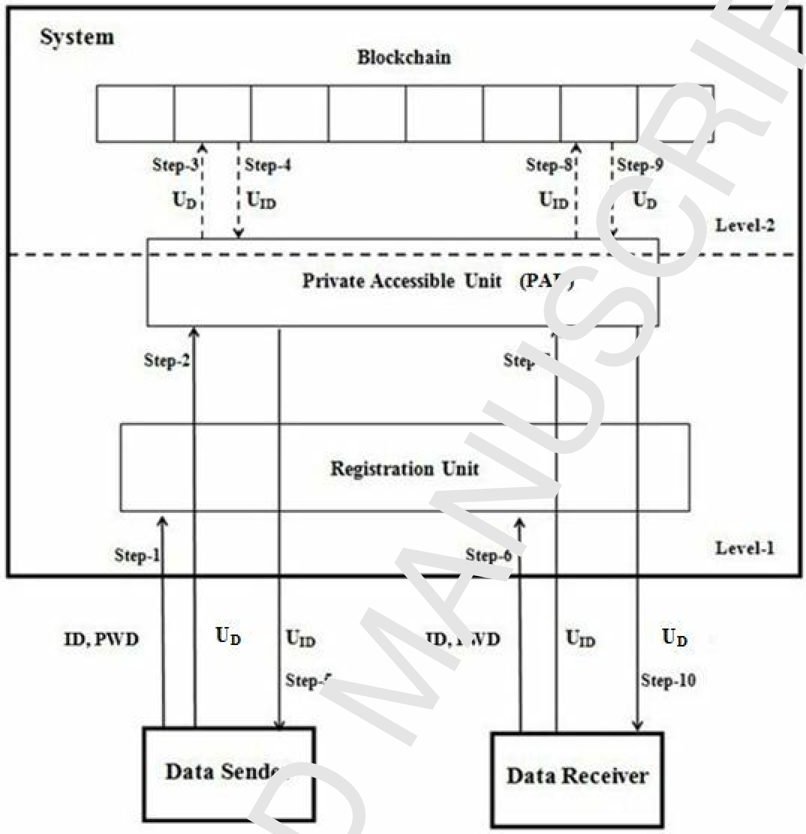


Figure 4: High level view of this system.

and need to preserve the ID and PWD. Further they just have to log in and access through secured channel for transaction of their private data in the cloud.

Private Accessible Unit (PAU) Both the parties of the system will be able to interact with PAU after authentication. It needs a secured channel to interact with PAU because through this unit they will send their private data to the System. It is the intermediary unit for both the levels of our system, through which the element of one level will interact with the other.

blockchain will hold the data of the users. Each transaction in the blockchain will return an identifier. Transaction identifiers will help the users to access the data further.

For better understanding our system is divided into two levels. Level-1 is Graphical User Interface (GUI). User will interact with our system through this level. Elements of level-1 are: Registration Unit and PAU. PAU is the element of both Levels so it will work between level 1 and 2. Level-2 is the backend of our system, which interacts with low level elements of this system through PAU. Element of level-2 is: blockchain. blockchain is being used as a repository of healthcare data in our system. Our platform uses permissioned blockchain which will require authentication to access.

Steps in the system : Steps of our system could be defined from Fig. 4.
Step-1 : Data sender will request with the ID and PWD to have access in the system.
Step-2 : Upon accessing the system in step-2, Data sender will send data to PAU for storing.
Step-3 & 4: Step 3 & 4 will take place in level-2 of our system, where PAU will send U_{ID} to blockchain and it will return U_{ID} for future access to the blockchain and also for finding the exact Block where the data were saved.
Step-5 : In this step PAU will return the U_{ID} to Data sender which was given by blockchain.
Step-6 : From this step rest of the steps are related to Data receiver. As step-1, this step also requires sign in process and after sign in Data receiver can request for the data.
Step-7 : In this step Data receiver will request for the data to Private Accessible Unit along with the U_{ID}. PAU will receive the U_{ID} for further use.
Step-8 & 9 : Step 8 & 9 are same as step 3 & 4 but the data are not same for this steps. In step-8 PAU will request the blockchain along with the U_{ID} and in Step-9 blockchain will return it.
Step-10 : This is the final step where PAU send the private data to the Data receiver.

4.2. Formal Description of Protocol

In this section we will define how Data sender, Data receiver, and our system will work altogether in case of sending and receiving the data. In case of data transmission in our system parties need to go through a step called registration. After confirmation of the Registration Unit that party can access the PAU.

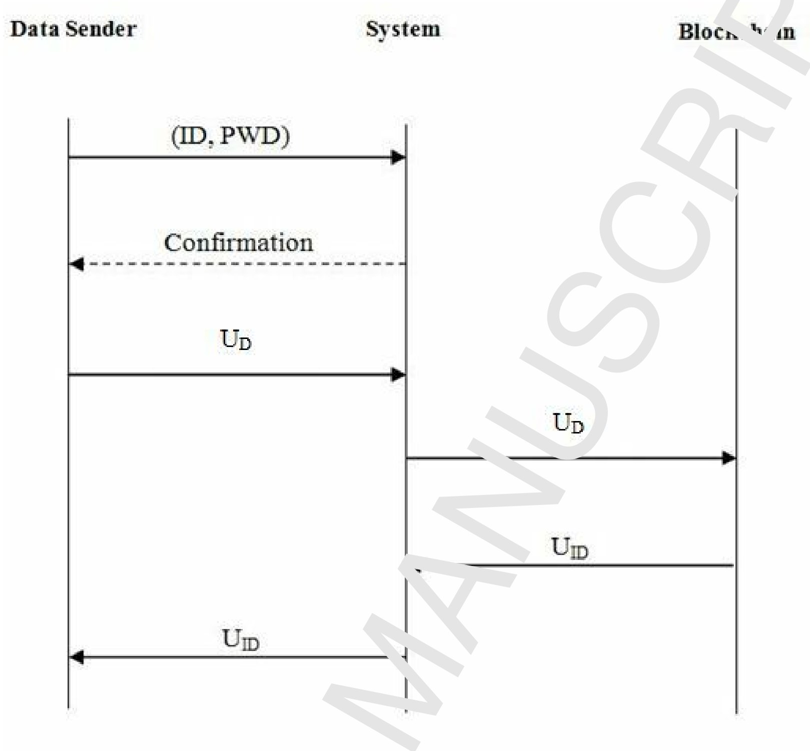


Figure 5: Low level view of Sending Protocol

4.2.1. Protocol between Data sender and System :

Fig. 5. Shows the low level view of sending protocol. A patient will play the role of a data sender in this protocol. Encrypted data will be sent to the system. Generation of ciphertexts solely depend upon a function known as encryption function. Generalized form of this function is *Enc(x,y)*. Below we will see how this function works.

$$Enc(key, Data) = U_D \tag{1}$$

By providing key and the health data to this function data sender will get U_D and will send it to the system. Public key encryption technique (e.g., Elliptic Curve Cryptography (ECC)) will be applied to encrypt the private data.
Suppose X is a Data sender of our system. At first X will request for getting into the system by providing the ID_X and PWD_X. Our system will send confirmation to X if she provides the right ID and PWD. If X could sign in to the system properly and gets the confirmation then she will send her U_{DX} to PAU through a secured

channel. Secured channel will provide the security to the transmission of data . In this stage PAU will interact with blockchain and this interaction with the blockchain will be done by the smart contracts of our system.
In our system smart contracts have been designed in a way such that blockchain will return the number of that block which has been denoted as U_{id}. Each block has a unique id which will work as an id of a specific patient. PAU will get the U_{id} on each transaction of data in the system for X it will be U_{idX}. PAU will send the U_{DX} to the blockchain then smart contract will return the special id U_{idX}, for X. After that PAU will send the U_{idX} to X and end the protocol. X has to store this U_{idX} otherwise next time X will not be able to access her personal private data. Getting the U_{idX} is the confirmation for X that means the data has been kept to the system and then X could log out and end the secured channel transmission with the system.

4.2.2. Protocol between Data Receiver and System:

Receiving in our system will take two layers of authorization. Because after registering or signing into our system parties will have to provide the U_{id} to get their data back through the secured channel. In this phase if they fail to submit the U_{id} then they will not be able to access their data. U_{id} is the key to receive the actual data. Fig. 6. shows a low level view of receiving protocol.
Suppose user X wants to retrieve her data which she sent to the system in sending phase. As like sending phase this phase is also controlled with the authentication or Registration unit where X has to sign in first then will be able to access our system. This sign in requires the ID and PWD of the user which was given in the registration phase. If X provides appropriate ID and PWD only then the system will send confirmation. After getting the confirmation X will be able to interact with the system through a secured channel. In this interaction with the system, X has to provide her U_{idX}. After getting the U_{idX} system will interact with blockchain. This interaction will take place in level-2 of our system. Only PAU can interact with blockchain, here the smart contracts of our system will be the medium.

Smart contract will send the U_{idX} to blockchain for retrieving the data of X from it. 256 bit hash of the corresponding block number will be checked in the smart contract, when the hash will be matched with any block then it will continue the process to retrieve the data. Otherwise this exception will be handled through the smart contracts.
Suppose the hash of any block is,

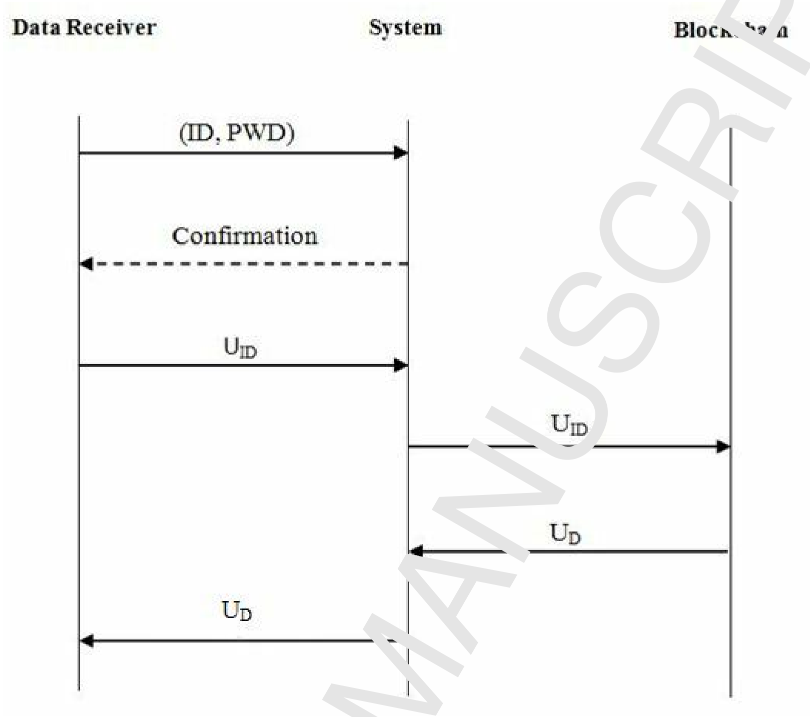


Figure 6: Low level view of Receiving Protocol

```
0xe3b1c14298fc1c149afbf4c8196fb92427ae41e4649b934ca495991b785
```

Only if the hash of U_{DX}'s corresponding block is same then X will be able to get her data. In our system blockchain will return the U_{DX} to PAU and it will be redirected to X later. After this data retrieval session will have it's end.
X will get her U_{DX} which has to be decrypted to get the actual raw data to decrypt the data user need to use *Dec(x,y)* function.

$$Dec(key, U_D) = plaintext \tag{2}$$

X will use equation 2 with key and U_{DX} to retrieve the raw data.

4.2.3. Storage of our system :

Our system will store the ID and PWD for authentication and response purpose. Our system solely will manage these private data in the cloud without depending on any other trusted third party (TTP) apart from the PAU. Each time when user will store the data she will get a new block to write so the block-id will change by time. ID and PWD is dependent on party but U_{id} is dynamic with each data storing process.

4.3. Programmatic view of MediBchain

Smart contract of our system has been presented in this paper through some algorithms. These algorithms have been designed to be converted in any blockchain based language (e.g., Solidity, Golang). Contracts of our system are written in Solidity language and all the results of this paper are also based on Solidity based environment. Algorithms 1-3 will be appropriate for any environment designed for blockchain environment.

Algorithm-1 describes how our system will check the issuers verifiability. All the hash of our system is denoted by $\mathcal{H}_{\mathcal{M}}$ and all the valid $\mathcal{H}_{\mathrm{i}}$ must be a part of $\mathcal{H}_{\mathcal{M}}$. Here, *i* refers to particular number of $\mathcal{H}_{\mathrm{i}}$.

$$\mathcal{H}_{\mathcal{M}} \longleftarrow \{\mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3, ..., \mathcal{H}_{\mathrm{i}}\}$$

$\mathcal{H}_{\mathcal{M}}$ is the set of all identical hashes of our system that will be provided in the time of account creation in the blockchain network. Γ & ν are part of $\mathcal{H}_{\mathcal{M}}$ and play significant role in transaction. Two different notations have been used to reduce the complexity of Algorithm-1, issuer of contract has been denoted with Γ and data uploader/downloader has been denoted with ν. Here, issuer is the address who runs the contract and message sender is the address who sends the message. If both of them are not same then Algorithm-1 will return false.

Algorithm 1: Checking of Issuer and Sender

```
Result: Verified Issuer
Γ, ν;
                         ▷ address of the issuer and message sender respectively
while Γ && ν ∈ H_M do
    if Γ ≠ ν then
        return;
    else
        _;
                                  ▷ It will proceed the code to next algorithm
    end
end
```

This algorithm is important for security and accountability of data transaction.

It will work in between the time of smart contract execution and the data transaction (e.g, upload, download) between MediBchain and blockchain. Eavesdroppers could take a chance of data manipulation in the meantime. All the accounts of this system will be the part of $\mathcal{H}_{\mathcal{M}}$ and also the initiator of contract and data uploader/downloader will be same. Execution of rest of the contract will be dependent on the similarity of Γ & ν. Algorithm-2 will be initiated after algorithm-1, in which δ_{n} represents the number of categories to be held by the structure of data in our contracts.

Algorithm 2: Transaction of Data

Result: Data Upload
struct Data $\longleftarrow \sum_1^n \delta_{\mathrm{n}}$
Data[] data;
bool$\longleftarrow$0;
while n **do**
    ▷ getting input from message sender,ν
    if ν returns string **then**
        data $\longleftarrow \sum_1^n \delta_{\mathrm{n}}$;
        bool $\longleftarrow$ 1;
        return bool;
    else
        return bool;
    end
end

Algorithm-2 will be executed after fulfilling the conditions below.
Iff,

$$\Gamma \,\&\&\, \nu \in \mathcal{H}_{\mathcal{M}} \text{ and,}$$
$$\Gamma \text{ (issuer)} = \nu \text{ (message sender)}$$

Here, Γ is the address who runs the contract and ν is the address who sends the message. If both of them are not same then this algorithm will return false. Users' (patients') data will be having different categories to be inputted. Different categories mean that the healthcare data come in different types, suppose user wants to save Blood sugars data and also Blood pressures data these two are different. By category we refer to this scenario that the user can store different diagnostic results in a block. Hence, we have designed two different contracts. In Algorithm-2 each structure will hold maximum four different types of healthcare data to be stored in the block if we change data part as follows,

$$data \longleftarrow \sum_{n=1}^{4} \delta_{\mathrm{n}}$$

We have another smart contract which takes maximum eight different types of health data to be stored in the block. For that we need to change data part again. So above data part will be changed as follows-

$$data \longleftarrow \sum_{n=1}^{8} \delta_{\mathrm{n}}$$

We have shown some computational analysis in subsection- 3.3 using the variation of data storing capabilities of different smart contracts.

Line-1 is showing that the structure of smart contract can take n number of individual data from a particular patient at a time. In the loop data will be assigned to its corresponding structure in line- 7 and then the data will be written in the block in the same contract. A particular structure will be written in a particular block. As mentioned earlier each block of blockchain holds different id which is not same as $\mathcal{H}_{\mathcal{M}}$. $\mathcal{H}_{\mathcal{M}}$ represents the account id of blockchain network whereas hash ids has been denoted with $\mathbb{H}_i$. A *bool* variable has been returned from Algorithm-2 as a flag for Algorithm-3. In Algorithm-3, ξ_i represents the block number and $\mathbb{H}_i$ represents the hash of particular block.

Algorithm 3: Block-id Generation

Result: Block-id

ξ_i, $\mathbb{H}_i$;
▷ Will hold the $\mathbb{H}_i \longleftarrow$ *Hash* of Block, $\xi_i \longleftarrow$ block number
while bool **do**
▷ Returned value from Algorithm 2
  if bool $\longleftarrow$ 1 **then**
    $\xi_i \longleftarrow$ *blockNumber()*;
    $\mathbb{H}_i$ = block.blockhash(ξ_i);
    return $\mathbb{H}_i$
  else
    return Null;
  end
end

Algorithm-3 will return hash id $\mathbb{H}_i$ if all the requirements will be fulfilled by the contract. It will take a variable named *bool* by which this algorithm will define whether to return block-id, $\mathbb{H}_i$ or not. Functions `block.Number()` and `block.blockhash()` are the syntex of Solidity language, where `block.Number()` will return the corresponding block number ξ_i and `block.blockhash()` will return $\mathbb{H}_i$.

$$\mathbb{B}_i \ni \xi_i, \mathbb{H}_i$$

ξ_i and $\mathbb{H}_i$ are the properties of each block, $\mathbb{B}_i$ by which our system will work

$$\mathbb{H}_i \longleftarrow \texttt{block.blockHash}(\xi_i)$$

Programmatically each $\mathbb{H}_i$ will be generated from it's corresponding ξ_i. As instance, if `block.blockhash()` gets ξ_1 as a parameter it will return $\mathbb{H}_1$ or if it gets ξ_{20} the function will return $\mathbb{H}_{20}$. So the relation can be written as,

$$\{\mathbb{H}_1, \mathbb{H}_2, \mathbb{H}_3, \mathbb{H}_4, ..., \mathbb{H}_i\} \equiv \{\xi_1, \xi_2, \xi_3, \xi_4, ..., \xi_i\}$$

5. Protocol Analysis & Evaluation

5.1. Security Analysis

- **Pseudonymity**: Data Sender, $\mathcal{S}$ and Receiver, $\mathcal{R}$ will not be identified by any party during transaction.

  - Pseudonymity of $\mathcal{S}$: After authentication $\mathcal{S}$ will upload the encrypted private data, U_D. Any other party will not be able to identify $\mathcal{S}$ by looking her U_D because of it's identificationless attribute.
  - Pseudonymity of $\mathcal{R}$: $\mathbb{H}_i$ will be used to trace particular $\mathbb{B}_i$ of the blockchain which holds the private data of $\mathcal{S}$. During transaction $\mathcal{T}$ party will hold the $\mathbb{H}_i$ to have her U_D back from the system, these $\mathbb{H}_i$s are as sensitive as the private data for receiver. $\mathbb{H}_i$ will be held by only our party which ensures the pseudonymity of Data Receiver because no one will be able to detect $\mathcal{S}$ during $\mathcal{T}$ or even after $\mathcal{T}$ because of encrypted property of U_D.
    Suppose α\{ID,PWD\} is the function for authentication,

$$\alpha\{\text{ID,PWD}\} \longrightarrow \{\mathcal{S},\mathcal{R}\}_{\texttt{authenticated}}$$

- **Privacy**: Registration Unit and U_D ensures the privacy of the $\{\mathcal{S},\mathcal{R}\}_{\texttt{authenticated}}$ and data respectively.

  - Privacy from system: Parties, $\{\mathcal{S},\mathcal{R}\}_{\texttt{authenticated}}$ of our system have privacy as pseudonymity of users is maintained. α\{ID,PWD\} will ensure the access in the system. This controlled access of $\{\mathcal{S},\mathcal{R}\}_{\texttt{authenticated}}$ provide privacy to the users of our system. Therefore, U_D of $\{\mathcal{S},\mathcal{R}\}_{\texttt{authenticated}}$ can not be compromised any way.

- Privacy from other parties: $\mathcal{S}$ will have her dedicated $\mathbb{B}_i$ in the blockchain to store U_D. So, if any $\{\mathcal{S},\mathcal{R}\}_{\texttt{authenticated}}$ of our system tries to access any other party's data it will not be able to access the particular block as each party will have their dedicated $\mathbb{H}_i$.

Clearly, the former analysis guarantees a very strong privacy of parties because only $\{\mathcal{S},\mathcal{R}\}_{\texttt{authenticated}}$ will be able to access as well as retrieve data from that particular $\mathbb{B}_i$.

◦ **Integrity**:

- Access request data integrity: Each time $\mathcal{S}$ or $\mathcal{R}$ tries to access the system, she needs to authenticate herself primarily. This access request needs to be done by both the dynamic entities- $\mathcal{S}$ and $\mathcal{R}$ of system. These access requests will require correct ID and PWD, which will be generated by party itself and will be holding by the database of system. So without $\mathcal{S}$ or $\mathcal{R}$ and system these authentication data will not be known by anyone. By which system guarantees the access request data integrity.
- User data Integrity: Use of *Enc(x,y)* function ensures the data integrity as the data in the blockchain will make no sense to any other person except the data owner. After retrieving the data from the system $\{\mathcal{S},\mathcal{R}\}_{\texttt{authenticated}}$ need to decrypt the U_D with *Dec(x,y)* function. In order to break this integrity level attacker needs to break the security of underlying encryption scheme, ECC.

All the data that are related to our healthcare data management system guarantees integrity.

◦ **Accountability**:

- Transactional $\mathbb{B}_i$: When any party will come to save it's data to the system a unique number or nonce, $\mathbb{H}_i$ will be returned which leverages the accountability of our system. Only party itself will be holding this

nonce which makes the party accountable for it's U_D because without valid information about
$\mathbb{B}_i \ni \xi_i, \mathbb{H}_i$ party will not be able to access her private data from blockchain.

- PAU as bridge: Interaction of $\{\mathcal{S},\mathcal{R}\}_{\texttt{authenticated}}$ with the system is controlled. This controlled path refers to the secured channel which will be created by the party itself through $\alpha\{ID, PWD\}$. Through this channel $\{\mathcal{S},\mathcal{R}\}_{\texttt{authenticated}}$ will interact with PAU which is a bridge between the system and blockchain. Secured channel makes the bridge accountable for secured $\mathcal{T}$ with blockchain.

◦ **Security**: Each $\mathbb{B}_i$ will be dedicated to $\{\mathcal{S},\mathcal{R}\}_{\texttt{authenticated}}$ and their $\mathbb{H}_i$ is secured as integrity is guaranteed in our platform. As a result, these $\mathbb{B}_i$ will not be accessed by any $\{\mathcal{S},\mathcal{R}\}_{\texttt{authenticated}}$. If attacker somehow manages to intrude into the blockchain network, patients' sensitive data will make no sense because of encrypted attribute of data. Accessing the raw data of patient will need the keys and $Dec(x,y)$ will return the raw data to parties. So, the data security is guaranteed in our platform.

The equation for **Transaction**,

$$\mathcal{T}(\delta_{\text{n}}) \longleftarrow \left\{\left\{\forall \mathcal{H}_{\mathcal{M}} : \Gamma \in \mathcal{H}_{\mathcal{M}}, \nu \in \mathcal{H}_{\mathcal{M}}, \Gamma = \nu\right\} \text{ and } \left\{\forall \alpha_{\{\mathcal{S},\mathcal{R}\}_{\texttt{authenticated}}} \{ID, PWD\}\right\}\right\}$$

After analyzing each of the properties we can conclude with saying that no platform secures blockchain based pseudonymous healthcare data other than our platform-'MediBchain', in the best of our knowledge.

5.2. Computation & Evaluation

We setup an environment to evaluate our protocol by writing programs using Solidity 0.4.11 and JAVA 1.8 with a computer Intel(R) Core(TM) i5, CPU-3.30 GHz, 8 GB of RAM, Win 8, 64-bit OS. We deployed Elliptic Curve Cryptography (ECC) for generating and retrieving the input and output respectively.

5.3. Data sharing:

We test the computation time to generate the cipher texts. Each encryption is an isolated process. Fig. 7. presents the data encryption time versus string size of healthcare data. We take several inputs to see how the rate of growth of

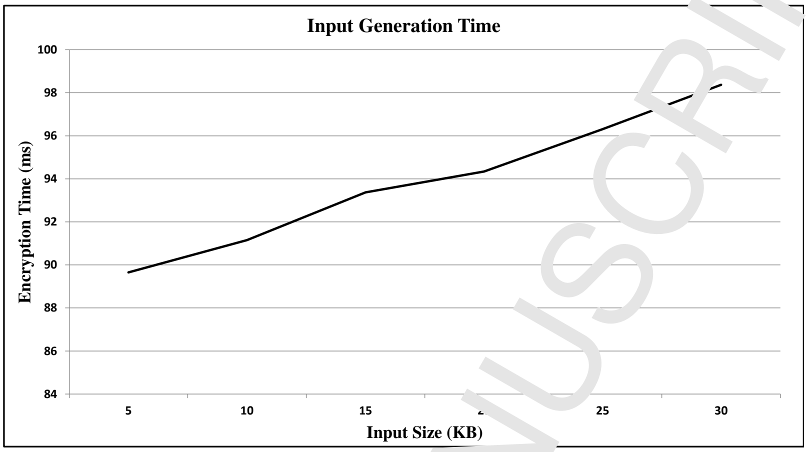


Figure 7: Computation time in generating input

time for encryption changes with variable input size. We take 5 to 30 kilobytes of data to analyze the encryption time of different data size. From the resultant graph we can observe the rate of growth of curve is nearly linear which means the encryption time increases with increase of data size. Data sharing phase of our system is variable and independent process, variable means that input size could vary for different users and independent means the encryption of different users' data are not dependent on each other.

5.3.1. Data manipulation with smart contract:

The issues that have been mentioned in the manuscript could be solved with other technologies but through blockchain environment we get the proper distributive attribute which lacks in others. Blockchain gives us the option to use it as distributed ledger which makes the technology a viable option. Ethereum environment has been used to analyze the effectiveness of this new idea of EHR system over windows operating system. Ethereum is the most effective platform to run Dapps (Distributed Apps) using solidity language that is the reason why Ethereum platform has been used to access blockchain.

Before getting access of a block in the blockchain network data will be accessed by our smart contract. Use of smart contract will cost some gas which is known as the cryptofuel of Ethereum Virtual Machine (EVM). To run any dapp (distributed application) on the Ethereum environment the executed application will need to

Figure 8: Computation cost of transaction and execution of smart contract

have some transactions in the network, in return of transaction the environment costs the executor some gas. Initiator or executor of transaction will get the gas in exchange of Ether in Ethereum environment. We evaluate two smart contracts, one with 4 inputs category other with 8 inputs category. In context of programming language which is number of variables to take input from party. Subsections 5.3.2 and 5.3.3 will depict the analogy of different terms of smart contract with 4 inputs category and 5.3.4 and 5.3.5 will depict the analogy between two different smart contracts with variable inputs, where . We tried to show some analogy based on the transaction and execution cost of our smart contract.

5.3.2. Transaction Cost vs Execution Cost:

Fig. 8. depicts the analogy between transaction and execution cost of smart contract. To have an accurate analyzing result we run the smart contract with different input sizes that varies from 5 to 100 characters of string. Curves in Fig. 8. shows the cost is increasing with the input size. But the rate of growth of these two curves is same between the intervals and linear too.

5.3.3. Block-id generation costs:

One of the key terms to be ensured while writing smart contracts was block-id generation. Block-id generation will cost for execution and transaction. We analyze the block-id generation cost with different string length, but interestingly it

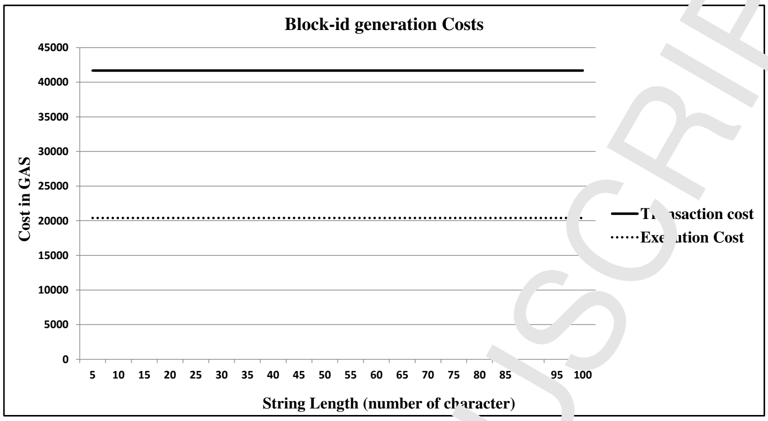


Figure 9: Computation cost of transaction and execution of smart contract

costs same for all the inputs. Fig. 9. shows the curves of execution and transaction cost of block-id generation. It is clear that each parameter is almost constant with the increase of the size of string. Transaction and execution cost is same for growing input size.

5.3.4. Transaction cost of variable inputs:

Parties of our system may have to upload a vast amount of data in different categories. Smart contract may have to be redesigned so that we analyze the cost to see how our platform reacts with an increasing amount of category to store it in blockchain. Before this subsection we were talking only about smart contract having 4 categories to take as input, but for having an effective analogy we will give 8 categories as input to see how the behavior changes of our platform. Fig. 10. shows us the analogy between two smart contracts in which one will take 4 inputs and other will take 8 inputs. In Fig. 10. we can see that smart contract having 8 categories of input will cost higher, but the rate of growth of curves are similar and the cost will increase with string size.

5.3.5. Execution cost of variable inputs:

Fig. 11. presents the execution cost of smart contract with variable input. As explained above smart contracts may vary in different scenario, so that we present the execution costs' analogy in Fig. 11. The rate of growth of curves is similar but smart contract with 8 inputs will cost more gas while execution with increasing string lengths.

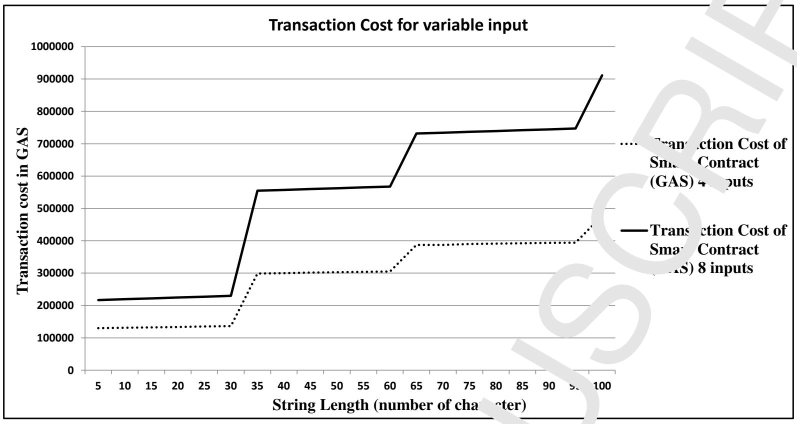


Figure 10: Transaction cost of smart contract with variable input

5.4. Output generation:

To get the plaintext or actual private healthcare data of patient the data from blockchain need to be decrypted. As like encryption, decryption or output generation process is also isolated. All the output generation for the parties is independent from each other. To analyze the output retrieval time we take different sets of string 5 to 30 kilobytes of data at a single input to get an actual idea of output retrieval time for the patients. In Fig. 12. curve shows that the rate of growth of time is related with the input size as the time is increasing for decryption with input size. The curve is nearly linear. Time is in millisecond in the graph that is computed with Java during decryption. Elliptic Curve Cryptography (ECC) is used to generate the plaintext.

5.4.1. Input generation vs Output retrieval:

Generation of input and output is independent from each other. Encryption will take place in the time of giving input and decryption will take place in the time of output. Fig. 13. depicts that two processes take very different amount of time while processing. With the string length both the time increase but encryption needs more time than decryption. For encryption it takes 80 to 90 milliseconds where decryption needs less than 10 milliseconds.

6. Conclusion

The paper presented privacy preserving platform for healthcare data in cloud. We have defined a set of security and privacy requirements for healthcare data

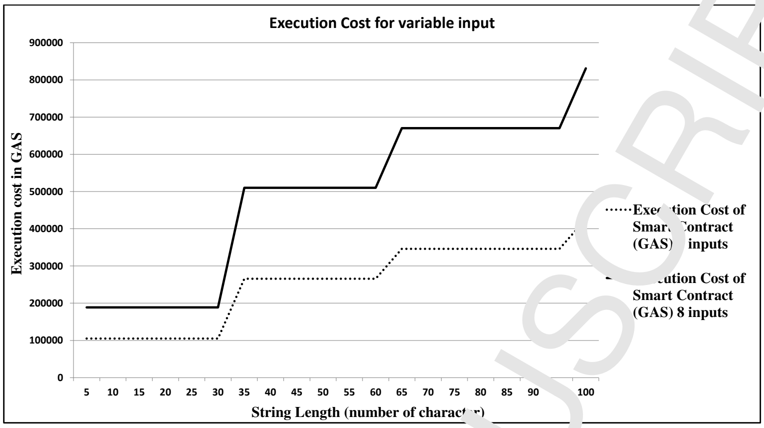


Figure 11: Execution cost of smart contract with variable input

management systems and argued why such attributes are necessary for a healthcare data management system in cloud. Our analysis shows that our platform satisfies all such requirements. Experimental performance evaluation shows that this platform runs well in blockchain environment. In the future we will try to explore the interoperability between different entities (e.g., diagnostic center, hospital, doctors, patients) of healthcare process, and another direction would be to address the issue of handling key-theft/loss mechanisms or key distribution techniques.

References

[1] Roman Beck, Jacob Stenum Czepluch, Nikolaj Lollike, and Simon Malone. Blockchain-the gateway to trust-free cryptographic transactions. In ECIS, page Research Paper153, 2016.

[2] R Böhme, N Christin, and B Edelman. Bitcoin: Economics, technology, and governance. Economic Perspectives, 2015.

[3] Jun Choe and Sun K Yoo. Web-based secure access from multiple patient repositories. International journal of medical informatics, 77(4):242–248, 2008.

[4] Jordi Cucurull and Jordi Puiggalí. Distributed immutabilization of secure

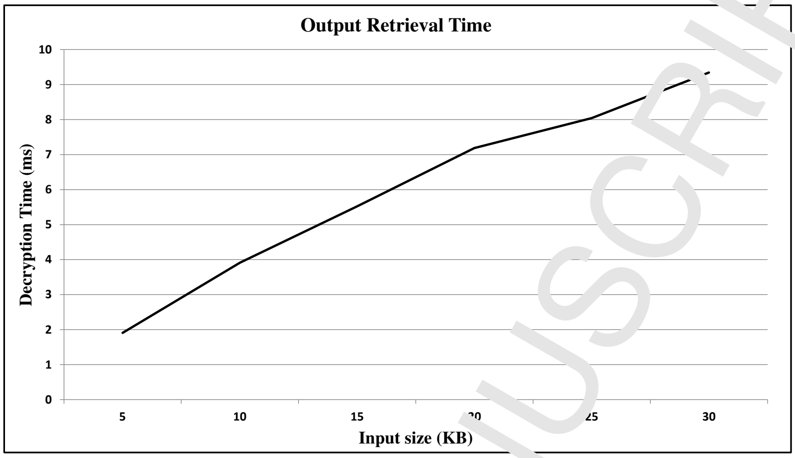


Figure 12: Computation time in generating actual output

logs. In International Workshop on Security and Trust Management, pages 122–137. Springer, 2016.

[5] Gaby G Dagher, Jordan Mohler, Matea Milojkovic, and Praneeth Babu Marella. Ancile: Privacy-preserving framework for access control and interoperability of electronic health records using blockchain technology. Sustainable Cities and Society, 39:283–297, 2018.

[6] Rocky Dariua. 4 Features of Blockchain Technology, 2016.

[7] Ariel Ekblaw, Asaph Azaria, John D Halamka, and Andrew Lippman. A case study for blockchain in healthcare:medrec prototype for electronic health records and medical research data. In Proceedings of IEEE Open & Big Data Conference, volume 13, page 13, 2016.

[8] FoxNewsHealth. 'Ransomware' cyberattack cripples hospitals across England. Associated Press, 5 2017.

[9] Rachel Frank. ISO/TC 307 - Blockchain and distributed ledger technologies.

[10] Joaquin Garcia-Alfaro. Data privacy management, autonomous spontaneous security, and security assurance. Springer, 2014.

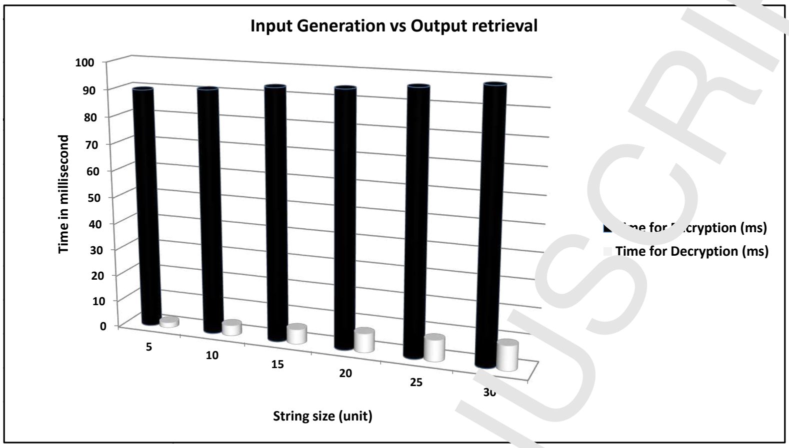


Figure 13: Input generation vs output retrieval time of system

[11] David S Gerstl. Leveraging bitcoin blockchain technology to modernize security perfection under the uniform commercial code. In International Conference of Software Business, pages 109–123. Springer, 2016.

[12] April Glaser. U.S. hospitals have been hit by the global ransomware attack - Recode, 2017.

[13] Omniyah Gul, Mahmoud Al Qutayri, Chan Yeob Yeun, and Quang Hieu Vu. Framework of a national level electronic health record system. In Cloud Computing Technologies, Applications and Management (ICCCTAM), 2012 International Conference on, pages 60–65. IEEE, 2012.

[14] E Hendrick, B Schooley, and C Gao. CloudHealth: developing a reliable cloud platform for healthcare applications. IEEE 10th Consumer Communications and Networking Conference (CCNC), 2013.

[15] Richard Hull, Vishal S Batra, Yi-Min Chen, Alin Deutsch, Fenno F Terry Heath III, and Victor Vianu. Towards a shared ledger business collaboration language based on data-aware processes. In International Conference on Service Oriented Computing, pages 18–36. Springer, 2016.

[16] Mike Jacobs. A Proposed Blockchain Reference Architecture, 2016.

[17] Neal Koblitz. Elliptic curve cryptosystems. Mathematics of computation, 48(177):203–209, 1987.

[18] Victoria Louise Lemieux. Trusting records: is blockchain technology the answer? Records Management Journal, 26(2):110–139, 2016.

[19] Xiong Li, Maged Hamada Ibrahim, Saru Kumari, Arun Kumar Sangaiah, Vidushi Gupta, and Kim-Kwang Raymond Choo. Anonymous mutual authentication and key agreement scheme for wearable sensors in wireless body area networks. Computer Networks, 129:429–443, 2017.

[20] Xiong Li, Jianwei Niu, Md Zakirul Alam Bhuiyan, Fan Wu, Marimuthu Karuppiah, and Saru Kumari. A robust ecc based provable secure authentication protocol with privacy protection for industrial internet of things. IEEE Transactions on Industrial Informatics, 2017.

[21] Laure A Linn and Martha B Koo. Blockchain for health data and its potential use in health it and health care related research. In ONC/NIST Use of Blockchain for Healthcare and Research Workshop. Gaithersburg, Maryland, United States: ONC/NIST, 2016.

[22] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.

[23] Svein Ølnes. Beyond bitcoin enabling smart government using blockchain technology. In International Conference on Electronic Government and the Information Systems Perspective, pages 253–264. Springer, 2016.

[24] Abdullah Al Omar, Mohammad Shahriar Rahman, Anirban Basu, and Shinsaku Kiyomoto. MediBchain: A blockchain based privacy preserving platform for healthcare data. In Security, Privacy, and Anonymity in Computation, Communication, and Storage, pages 534–543. Springer International Publishing, 2017.

[25] Manas Ranjan Patra, Rama Krushna Das, and Rabi Prasad Padhy. Crhis: cloud based rural healthcare information system. In Proceedings of the 6th International Conference on Theory and Practice of Electronic Governance, pages 402–405. ACM, 2012.

[26] Siraj Raval. Decentralized Applications: Harnessing Bitcoin's Blockchain Technology. " O'Reilly Media, Inc.", 2016.

[27] Carlos Oberdan Rolim, Fernando Luiz Koch, Carlos Becker Westphall, Jorge Werner, Armando Fracalossi, and Giovanni Schmitt Salvador. A cloud computing solution for patient's data collection in health care institutions. In eHealth, Telemedicine, and Social Medicine, 2010. ETELEMED'10. Second International Conference on, pages 95–99. IEEE, 2010.

[28] Mike Sharples and John Domingue. The blockchain and kudos: A distributed system for educational record, reputation and reward. In European Conference on Technology Enhanced Learning, pages 490–496. Springer, 2016.

[29] Miloš Simić, Goran Sladić, and Branko Milosavljević. A case study iot and blockchain powered healthcare.

[30] J Sun, J Yan, and KZK Zhang. Blockchain-based sharing services: What blockchain technology can contribute to smart cities. Financial Innovation, 2016.

[31] Melanie Swan. Blockchain: Blueprint for a new economy. " O'Reilly Media, Inc.", 2015.

[32] Ingo Weber, Xiwei Xu, Régis Riveret, Guido Governatori, Alexander Ponomarev, and Jan Mendling. Untrusted business process monitoring and execution using blockchain. In International Conference on Business Process Management, pages 329–347. Springer, 2016.

[33] Duane Wilson and Giuseppe Ateniese. From pretty good to great: Enhancing pgp using bitcoin and the blockchain. In International Conference on Network and System Security, pages 368–375. Springer, 2015.

[34] JJ Xu. Are blockchains immune to all malicious attacks? Financial Innovation, 2016.

[35] Xiao Yue, Huiju Wang, Dawei Jin, Mingqiang Li, and Wei Jiang. Healthcare data gateways: found healthcare intelligence on blockchain with novel privacy risk control. Journal of medical systems, 40(10):218, 2016.

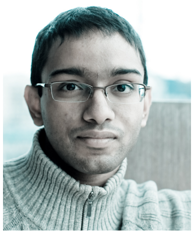
[36] Zach Herbert. Why blockchains are the future of cloud storage. Sia Blog, 2017.

[37] Yin Zhang, Meikang Qiu, Chun-Wei Tsai, Mohammad Mehedi Hassan, and Atif Alamri. Health-cps: Healthcare cyber-physical system assisted by cloud and big data. IEEE Systems Journal, 11(1):88–95, 2017.

[38] Peng Zhanga, Jules Whitea, Douglas C Schmidta, Gunther Lenzb, and S Trent Rosenbloomc. Fhirchain: Applying blockchain to securely and scalably share clinical data. The Journal of Network and Computer Applications, 2018.

Abdullah Al Omar received his B.Sc degree from Department of Computer Science and Engineering, University of Asia Pacific in 2016. Currently he is working as a Lecturer at the Department of Computer Science and Engineering, University of Asia Pacific. His research interests include Applied Cryptography, Protocol Construction, Privacy-preserving and secured platform design and blockchain.

Md Zakirul Alam Bhuiyan received the PhD degree and the M. Eng degree from Central South University, China, in 2013 and 2009 respectively, and the BSc degree from International Islamic University Chittagong, Bangladesh, in 2005, all in Computer Science and Technology. He is currently an assistant professor of the Department of Computer and Information Sciences at the Fordham University. Earlier, he worked as an assistant professor at the Temple University and a post-doctoral fellow at the Central South University, China, a research assistant at the Hong Kong PolyU, and a software engineer in industries. His research focuses on dependable cyber physical systems, WSN applications, big data, cloud computing, and cyber security. He served as a lead guest editor of IEEE TBD, ACM TCPS, Information Sciences, and so on. He also served as general chair, program chair, workshop chair, publicity chair, TPC member, and reviewer of international journals/conferences. He is a member of IEEE and a member of ACM.



Dr. Anirban Basu is a Researcher at Hitachi R&D in Japan, and a Visiting Research Fellow at the University of Sussex. He holds a Ph.D. in Computer Science (2010) and a Bachelor of Engineering (Hons.) in Computer Systems Engineering (2004) from the University of Sussex. His research focuses on a user-centric view of privacy; and computational trust as an information security paradigm in an increasingly knowledge-based connected world. His work has generated over 70 refereed publications and about 20 co-authored Japanese patent applications. He is particularly active within the IFIPTM computational trust management community.

Shinsaku Kiyomoto received his B.E. in engineering sciences and his M.E. in Materials Science from Tsukuba University, Japan, in 1998 and 2000, respectively. He joined KDD (now KDDI) and has been engaged in research on stream ciphers, cryptographic protocols, and mobile security. He is currently a senior researcher at the Information Security Laboratory of KDDI R&D Laboratories (now KDDI Research, Inc). He was a visiting researcher of the Information Security Group, Royal Holloway University of London from 2008 to 2009. He received his doctorate in engineering from Kyushu University in 2006. He received the IEICE Young Engineer Award and IEICE Achievement Award in 2004 and 2016 respectively. He is a member of JPS and IEICE.

Mohammad Shahriar Rahman is currently an associate professor at the University of Liberal Arts Bangladesh. Earlier, he worked as a research engineer at the Information Security group of KDDI Research, Japan. He received his Ph.D. and M.S. degrees in information science from Japan Advanced Institute of Science and Technology (JAIST), in 2012 and 2009 respectively, and B.Sc. in computer science and engineering from University of Dhaka, Bangladesh, in 2006. His research interests include secure protocol construction, privacy-preserving computation and security modeling. He is a member of International Association for Cryptologic Research (IACR). Dr. Rahman has co-authored 40+ research papers and submitted 8 co-authored Japanese patent applications.

Highlights:

1. We proposed a user centric EHR systems for Healthcare, which gives the total controlling power of the data to the users.
2. In generic EHR platform, it becomes easier target of intruders to intrude the system than totally breaking the security of the system. We solved this problem by implementing permissioned Blockchain along with the cryptographic functions.
3. We explored the archival use of Blockchain in our platform by storing the data of users in the blocks of the permissioned Blockchain.
4. Controlling the pseudonymity of the users is a big challenge. We solved the pseudonymity issue by applying cryptographic function. We used Elliptic Curve Cryptography (ECC) to make the data safe from other party in this distributed ledger system.