BlocHIE: a BLOCkchain-based platform for Healthcare Information Exchange

Conference Paper · April 2018

DOI: 10.1109/SMARTCOMP.2018.00073

6 authors, including:

Shan Jiang
The Hong Kong Polytechnic University
14 PUBLICATIONS **202** CITATIONS
SEE PROFILE

Yanni Yang
The Hong Kong Polytechnic University
18 PUBLICATIONS **259** CITATIONS
SEE PROFILE

Mingyu Derek Ma
University of California, Los Angeles
8 PUBLICATIONS **113** CITATIONS
SEE PROFILE

BlocHIE: a BLOCkchain-based platform for Healthcare Information Exchange

Shan Jiang*, Jiannong Cao*, Hanqing Wu*, Yanni Yang*, Mingyu Ma*, Jianfei He†
*The Hong Kong Polytechnic University, Hong Kong, China
†Huawei Technologies Co. Ltd., Shenzhen, China
{cssjiang,csjcao,cshwu,csynyang}@comp.polyu.edu.hk, derek.ma@connect.polyu.hk, jeffrey.he@huawei.com

***Abstract*—Nowadays, a great number of healthcare data are generated every day from both medical institutions and individuals. Healthcare information exchange (HIE) has been proved to benefit the medical industry remarkably. To store and share such large amount of healthcare data is important while challenging. In this paper, we propose BlocHIE, a Blockchain-based platform for healthcare information exchange. First, we analyze the different requirements for sharing healthcare data from different sources. Based on the analysis, we employ two loosely-coupled Blockchains to handle different kinds of healthcare data. Second, we combine off-chain storage and on-chain verification to satisfy the requirements of both privacy and authenticability. Third, we propose two fairness-based packing algorithms to improve the system throughput and the fairness among users jointly. To demonstrate the practicability and effectiveness of BlocHIE, we implement BlocHIE in a minimal-viable-product way and evaluate the proposed packing algorithms extensively.**

I. Introduction

Healthcare has always been important to the society. Illness, accidents, and emergencies do arise every day, and the incurred ailments and diseases are supposed to be diagnosed, treated, and managed. In recent years, healthcare information exchange (HIE) among medical institutions has been proved to benefit the medical industry a lot [1]. First, HIE can enhance the understanding of each individual clinical trial. Second, the researchers can get scientific insights by analyzing a bunch of clinical trials. Third, the healthcare information interoperability between clinical research enterprises strengthens their collaborations.

Besides utilizing the data shared by the medical institutions, daily data collection is also beneficial for personal healthcare. With the development of the Internet of things (IoT) technology [2] [3], numerous personal healthcare data are generated by the IoT devices every day [4]. The doctor can take advantage of these data for precision medicine [5]. That is, the doctor takes the individual variability in environment and lifestyle into consideration when conducting disease treatment or giving prevention advice. There is no doubt that the data from individuals and various medical institutions benefits healthcare. However, it is challenging to store and share such large amount of data.

Early success in HIE arose from the field of cloud computing [6]. The idea to store the huge amount of data remotely rather than locally is simple but effective. The cloud service providers (CSPs) propose various schemes for reliable data storage and efficient data processing. Then the stakeholders choose a specific CSP by balancing various factors such as cost and reliability. It has been a trend to resort to CSPs when there are some data to be stored. The beneficiaries range from patients, medical institutions, and research institutions to big corporations. Therefore, the CSPs have been taking great responsibilities to provide a controlled, cross-domain and flexible HIE platform.

However, the CSPs have been struggling a lot to provide data sharing service [7]. On the one hand, the cloud storage market has been dominated by the largest CSPs such Google, Dropbox, *etc.* They are unwilling to share their data with the small/medium ones and between themselves due to market competition. On the other hand, it is risky if the healthcare data, which is highly private information, is exposed to the malicious users unexpectedly. Fortunately, Blockchain technology, which starts at 2008 [8] and booms at 2014 [9], provides great potential for HIE through its attractive features such as security, privacy, decentralization, and immutability.

Blockchain technology has been successfully applied in many areas. Bitcoin [8], as the first decentralized cryptocurrency, is also the first successful Blockchain application. After the boom of cryptocurrencies, it comes to the era of Blockchain 2.0 with the release of Ethereum [9]. During this time, a lot of Blockchain-based systems are proposed for the purpose of decentralization. The applications range from transportation [10], e-government [11] to education [12].

When Blockchain technology meets HIE, there are only few proposed systems [13][14] and they all suffer from the following two problems. First, they only consider to store and share the electronic medical records (EMRs) and ignore the useful and numerous personal healthcare data (PHD). The requirements to store and share the huge amount of PHD are significantly different from storing and sharing the EMRs, which brings new challenges in the aspect of system throughput and fairness. Second, the existing systems directly store the EMRs in the cloud environment with complicated access control mechanism to prevent undesired data dissemination. However, such system architecture heavily relies on the security of the cloud environment.

To address the issues mentioned above, we propose BlocHIE, a BLOCkchain-based platform for Healthcare Information Exchange. In the system architecture, we use two loosely-coupled Blockchain, namely EMR-Chain and PHD-Chain to store EMRs and PHD separately. For the EMR-Chain,

978-1-5386-4705-9/18/$31.00 ©2018 IEEE
DOI 10.1109/SMARTCOMP.2018.00073

we combine off-chain storage and on-chain verification to take care of both privacy and authenticability, which also removes the dependency on cloud services. For both of the EMR-Chain and PHD-Chain, we propose two fairness-based transaction packing algorithms to enhance the system throughput, and to improve the fairness among the system users.

The contributions of this paper are as follows:

- We analyzed the requirements for storing and sharing EMRs and PHD. Based on the analysis, we propose to use two loosely-coupled Blockchain, namely EMR-Chain and PHD-Chain, as the system architecture. The EMR-Chain stores EMRs from medical institutions while the PHD-Chain serves the data from individuals. The usage of multiple chains satisfies the different requirements of storing and sharing different data.
- We combine off-chain storage and on-chain verification in the EMR-Chain, which fulfills the requirements of privacy and authenticability, at the same time reduces the storage overhead for the EMR-Chain.
- We propose two fairness-based transaction packing algorithms, namely FAIR-FIRST and TP&FAIR, for the EMR-Chain and PHD-Chain respectively. The proposed algorithms can enhance the system throughput and improve the fairness among the users.
- We implement BlocHIE in a minimal-viable-product way. The implementation demonstrates the practicability of BlocHIE. Moreover, we evaluate the packing algorithms of FAIR-FIRST and TP&FAIR in terms of fairness and throughput. The experimental result indicates that FAIR-FIRST enhances fairness significantly and TP&FAIR improves throughput remarkably while guaranteeing an acceptable fairness.

The rest of the paper is organized as follows. In Section II, we describe the preliminaries towards developing BlocHIE. Section III demonstrates the design of BlocHIE. In its subsections, three key novelty ranging from system architecture to underlying algorithm are introduced. The system implementation and evaluation are showcased in Section IV. Finally, Section V concludes the paper.

II. PRELIMINARIES

In this section, we formally describe the preliminaries used in BlocHIE. We first introduce how Blockchain works, the advantages of Blockchain, and how Blockchain benefits BlocHIE in Subsection II-A. Then, we summarize the existing distributed consensus algorithms and how BlocHIE is built upon them in Subsection II-B.

A. Blockchain - Distributed Ledger Technology

A Blockchain is an append-only data structure, to store a continuously growing list of transactions. A Blockchain is replicated and maintained among the members of a network. As a distributed ledger, Blockchain has two key features, *i.e.*, immutability and non-repudiability. The immutability is achieved because it is computationally impossible to modify any committed transaction in the Blockchain. The transactions in a Blockchain are non-repudiable since they are replicated by a large number of entities.

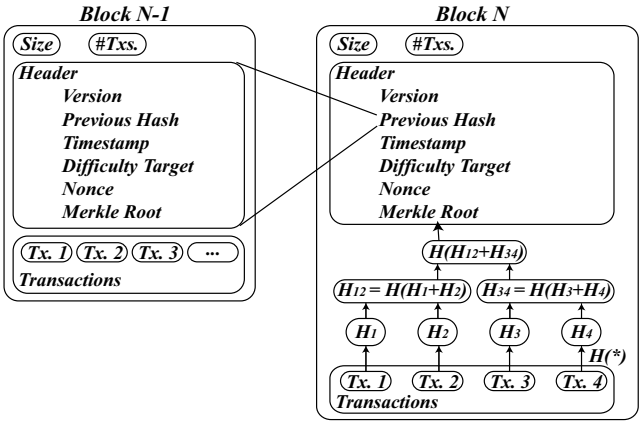

Fig. 1. Structure of a Traditional Blockchain

Traditionally, a Blockchain is a chain of blocks linked and secured using cryptography. As shown in Fig. 1, each block contains four components, namely block size, transaction counter, block header, and transactions. The block size and the transaction counter are the number of bytes of the block and the number of transactions, respectively. The block header contains six fields, namely version, previous block hash, timestamp, difficulty target, nonce, and Merkle root. The version is a version number to track the consensus protocol upgrades, the timestamp is the approximate creation time of the block, while the difficulty level and nonce are used for proof-of-work consensus protocol. The Merkle root refers to the hash of all the hashes of all the transactions. The previous block hash is a reference to the hash of the previous block along the chain. The hash value of a block, which is the primary identifier of a block, is made by hashing the block header twice through the SHA-256 hash function.

In our system BlocHIE, we take advantage of the immutability and non-repudiability for HIE. On the one hand, the feature of immutability is essential to prevent untrustworthy or malicious modification on the healthcare records. On the other hand, the healthcare records, as evidence to showcase the treatment procedure between medical institutions and individuals, should be non-repudiable. In addition, with the feature of non-repudiability, an unnecessary disputation between the medical institutions and individuals can be avoided.

B. Distributed Consensus

A Blockchain is replicated among the members of a network, in which each member holds a replication of the committed transactions and a pool of the submitted but uncommitted transactions. Each member is responsible for packing the transactions from the pool to the blocks to make them committed. In order to make the Blockchain remain functional, the members need to agree on a certain state of the Blockchain. This procedure is accomplished by the underlying distributed consensus algorithm.

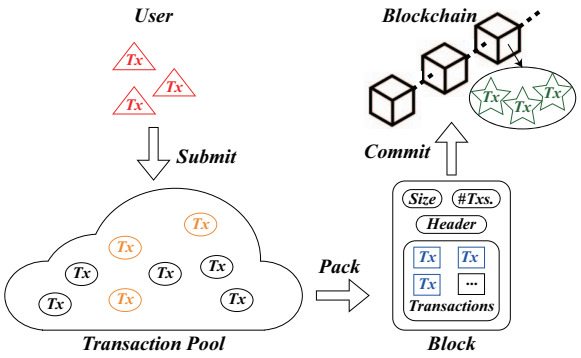

Fig. 2. The procedure for commiting transactions

As shown in Fig. 2, it requires three steps for a transaction to be committed. At first, the user has some *raw* transactions (the red triangles) and wants to publish them. Then, the user submits the raw transactions to the Blockchain network. Each member in the Blockchain network receives the transactions from the user and maintains a transaction pool. The transactions in pool are called *submitted* transactions (the yellow and the black ellipses). At this time, the members are supposed to make consensus on the way to maintain the Blockchain based on the transaction pool. The consensus consists of two steps, namely packing and committing. At the packing stage, each member selects some submitted transactions and puts them into a block. The transactions that are packed into a block but not yet committed are called *packed* transactions (the blue rectangles), and the block containing packed transactions are called uncommitted block. Finally, at the step of committing, the members make efforts to get the uncommitted blocks validated and committed. If a transaction is in a validated block, it is said to be *committed* (the green tars).

As stated above, packing and committing are both required for the distributed consesus. However, traditional Blockchain-based systems, *e.g.*, Bitcoin [8] and Ethereum [9], only focus on the step of committing. Some of the applied committing protocols include proof-of-work (PoW) [15], proof-of-stake (PoS) [16], proof-of-burn [17], *etc.* In BlocHIE, we employs PoW as one of the building blocks, which is introduced in detail as follows. The PoW committing protocol is based on some pre-defined puzzles that are difficult, *i.e.*, costly and time-consuming, to solve but easy to be verified. For example in Bitcoin [8], the miner has to change the nonce (as introduced in Fig. 1) constantly to make the hash value of the block begin with a certain amount of zeros. It is difficult to find such nonce while the validity is easy to be checked once found. The cryptographic hash function used in Bitcoin is twice SHA-256. Other hashing algorithms, including Scrypt (used in Litecoin [18]) CryptoNight [19] (used in Monero [20]), *etc.*, are also employed.

III. BlocHIE System Design

In this section, we outline the proposed platform BlocHIE, a BLOCkchain-based platform for Healthcare Information Exchange. We highlight the three key innovation points in the following subsections.

A. System Architecture: Loosely-coupled EMR-Chain and PHD-Chain

The system architecture of BlocHIE is presented in Fig. 3. BlocHIE is envisioned for storing and sharing healthcare data from medical institutions and individuals. There are mainly three components in BlocHIE. The first component is the Blockchain network. The Blockchain network is responsible for storing and sharing the collected healthcare data. Anyone who is willing to contribute to this platform can join the network. The medical institutions, *e.g.*, hospitals and clinics, act as the second component. When there are new patients in a hospital, their diagnostic records will be submitted to the Blockchain network and shared with other hospitals and clinics. The privacy issue will be discussed in subsection III-B. The third component consists of all the individuals who are willing to store and share their daily healthcare data. In a smart home, numerous healthcare data are generated by the IoT devices, *e.g.*, smart watch, smart thermometer, and smart sphygmomanometer. These devices can automatically submit the generated data to the Blockchain network.

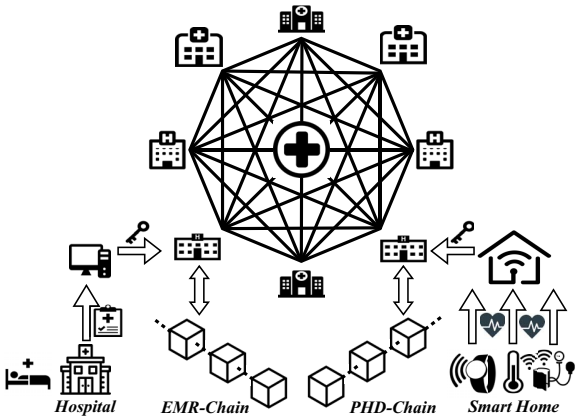

Fig. 3. BlocHIE system architecture

Among the three components, there are two parties, *i.e.*, medical institutions and individuals, who are submitting and sharing healthcare data in BlocHIE. The reason why we separate them is that there are different requirements to share their data. For medical institutions, what they submit are medical diagnostic report, medical examination report, *etc.* These data are incredibly privacy-sensitive. Moreover, there is a high demand to authenticate these data. For example, if a patient receives some treatment in a medical hospital and the medical diagnostic report is published with the signatures from both the hospital and the patient, neither the hospital nor the patient can deny the treatment. When it comes to the data generated by the individuals, the primary concern is the quantity. The amount of healthcare data generated by each person is remarkable. Besides, the individuals compete to publish their data for future healthcare usage. Consequently,

the key requirements to publish and share individuals' data are high throughput and substantial fairness. In the following parts, we abbreviate the data generated by medical institutions and individuals as critical data (EMR) and personal healthcare data (PHD) respectively.

TABLE I
REQUIREMENTS TO PUBLISH AND SHARE HEALTHCARE DATA

Requirement	EMR	PHD
privacy	high	moderate
authenticability	high	no
throughput	moderate	high
latency	moderate	moderate
fairness	moderate	moderate

The requirements to publish and share EMRs and PHD are summarized in Tab. I. As we can see from the summarization, their requirements are significantly different. Hence, we propose to store and share EMR and PHD with two loosely-coupled Blockchains, namely *EMR-Chain* and *PHD-Chain*. Suppose that a person visits a hospital and some medical diagnostic records are generated. If both the patient and the hospital agree to publish the data, the data will be published to the EMR-Chain with their signatures. Suppose that some daily healthcare data are generated in a smart home, the data will be published on PHD-Chain with the signature of the owner.

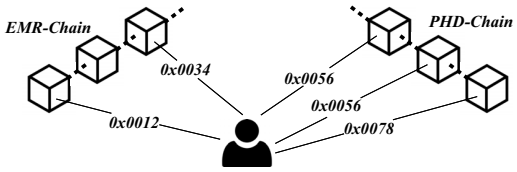

Fig. 4. An individual can have multiple identities on EMR-Chain and PHD-Chain

As shown in Fig. 4, EMR-Chain and PHD-Chain are coupled because a single person can publish data on both of the chains. However, they are only loosely coupled since the identities of the same person can be different. An individual knows the set of identities he/she owns. Indeed, the identity on EMR-Chain can be interpreted as the unique record identifier, while the identity on PHD-Chain can be treated as the unique device identifier. When there is requirement to query the healthcare data, the person can use the set of identities to fetch data on both of the chains.

To conclude, we propose to use loosely-coupled EMR-Chain and PHD-Chain to store and share EMR and PHD respectively. The proposed system architecture can satisfy different requirements of storing EMR and PHD concerning privacy, authenticability, throughput, latency, and fairness.

B. Combining Off-chain Storage and On-chain Verification

In subsection III-A, we propose to use EMR-Chain to store and share EMRs. As summarized in Tab. I, the key requirements of EMR are privacy and authenticability. However, in existing Blockchain-based system, these two properties cannot be guaranteed at the same time. Specifically, the whole data is stored in existing Blockchain-based systems, which arouses great privacy concern. To preserve privacy and authenticability simultaneously, we propose to combine off-chain storage and on-chain verification.

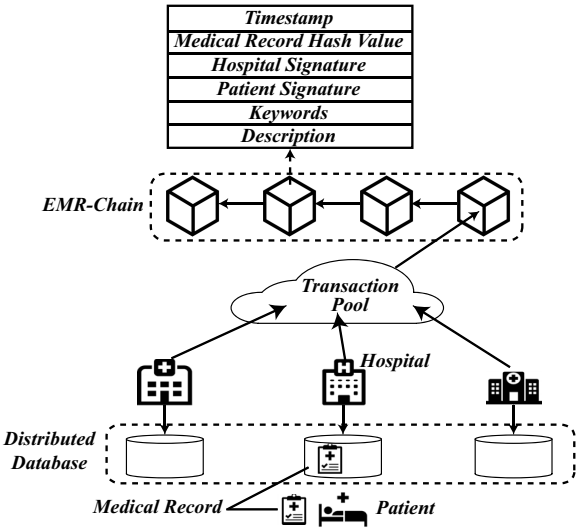

Fig. 5. The mechanism and structure of EMR-Chain

The process of publishing a piece of EMR is shown in Fig. 5. When a medical record of a patient is generated at a hospital, three copies of the medical record will be generated. The first copy is stored in the database of the hospital, the second copy is sent to the patient, and the third copy is submitted to the Blockchain network. The first two copies are identical and contain the full information of the EMR while there is a vast difference between the third copy and the first two. The full structure of the third copy is shown in Fig. 5. It contains the timestamp, the hash value of the medical record, the hospital signature, the patient signature, a set of keywords, and extra description. The hash value of the medical record is generated using some digest algorithm such as MD5.

Indeed, the third copy serves as a proof-of-existence copy rather than a full copy. The advantages of such structure are as follows. First, the detailed medical record is not publicly accessible, which preserves the privacy of the patient. Second, EMR-Chain reduces the throughput requirement significantly. The original medical records are large files of several megabytes. If they are stored, it requires a very high throughput of the system. Moreover, a single block can even only contain a single record. In EMR-Chain, the hash value, whose size is several kilobytes, is stored instead, which reduces the throughput requirement. Third, authenticability is preserved in EMR-Chain. The patient and the hospital can compare their records in hand with the hash value along the EMR-Chain to authenticate the medical record. It can prevent the repudiation of the hospital and the patient. Finally, EMR-Chain enhances the searchability. The keywords published along with the medical record can be used for information

retrieval. For example, if the data of a certain kind of disease is desired, the disease name be used for searching.

To conclude, the design concept of EMR-Chain is to combine off-chain storage and on-chain verification. On the one hand, the off-chain storage is achieved by storing in the distributed databases of the hospitals. On the other hand, the on-chain verification is achieved by including the hash value of each medical record in the transaction.

C. Fairness-based Transaction Packing Algorithm

In subsection III-A, we propose to use PHD-Chain to store and share data from individuals. As summarized in Tab. I, the key requirement of PHD is throughput. However, existing Blockchain-based system cannot satisfy the throughput requirement of sharing PHD. To this end, we propose two fairness-based transaction packing algorithms. The proposed algorithms can bring about not only high throughput but low latency and moderate fairness as well.

To introduce the algorithm, we firstly define some terminologies, *i.e.*, *response time*, *waiting time*, and *fairness*. Jain *et. al.* introduced Jain's fairness index to evaluate the fairness in allocation of a resource to a set of users/devices [21]. Suppose there are n users sharing a network service and x_i to be the throughput for the i-th user, then the Jain's fairness index is defined as $\mathcal{J}(x_1, x_2, \cdots, x_n) = (\sum_{i=1}^{n} x_i)^2/(n \cdot \sum_{i=1}^{n} x_i^2)$. In a Blockchain-based system, the fairness is defined in a similar way.

Definition 1. *Suppose a transaction x_i is submitted at time s_i and committed at time e_i, then the response time t_i of x_i is defined as:*

$$t_i = e_i - s_i \tag{1}$$

Definition 2. *Suppose there are n committed transactions $x_1, x_2, \cdots, x_n$ with response time $t_1, t_2, \cdots, t_n$. The fairness of the system is defined as:*

$$\mathcal{J}(x_1, x_2, \cdots, x_n) = \frac{(\sum_{i=1}^{n} t_i)^2}{n \cdot \sum_{i=1}^{n} t_i^2} \tag{2}$$

Definition 3. *Suppose a submitted or packed transaction x_i is submitted at time s_i, then the waiting time w_i of x_i at time t_c is defined as:*

$$w_i = t_c - s_i \tag{3}$$

We assume that the submitting times of the transactions are distinct. It is reasonable since there must be a slight time difference between submitting two transactions. Even if two transactions are submitted at the the same time, the symmetry can be broken by comparing the transaction content *e.g.*, assume that the transaction with larger hash value is submitted slightly later. According to Eq. 3, the waiting times of transactions in pool are distinct as well.

In a Blockchain-based system, the number of transactions inside a block should be bounded. A block will be of huge size if too many transactions are included. When a huge block is synchronized in the Blockchain network, the network congestion will be very high. There are a lot of research on setting the optimal block size [22][23]. In our system, we set the maximum number of transactions inside a block to be an adjustable parameter m.

When the Blockchain network wants to get some transactions committed, each node in the Blockchain network is supposed to follow the procedure illustrated in Fig. 2. That is, the nodes are supposed to select some transactions from the transaction pool first. Different transactions in the transaction pool can have different waiting times. It is intuitive to pack as many transactions as possible and to pack those transactions that have the longest waiting times. On the one hand, it can increase the throughput to pack as many transactions as possible. On the other hand, it can enhance the fairness to pack those transactions with the longest waiting times. To this end, all the nodes will pack the transactions with top-m waiting times. However, it is indeed a waste of computing resources for all the nodes to work on the same subset of transactions. As a result, it is a problem to coordinate the nodes to pack transactions in the Blockchain network to take both fairness and throughput into consideration.

As above, we give the intuition to pack the transactions with top-m waiting times. Here, we formally prove that this strategy achieves the maximum fairness.

Theorem 4. *Given a setting of n transactions $x_1, x_2, \cdots, x_n$ in pool with waiting time $w_1, w_2, \cdots, w_n$ and m transactions are supposed to be packed, the strategy to pack the transactions with top-m waiting times achieves the maximum fairness.*

Proof. Assume without loss of generality that the transactions $x_1, x_2, \cdots, x_n$ are sorted with decreasing waiting times, *i.e.*, $w_1 > w_2 > \cdots > w_n$. Let us notate the strategy to pack the transactions with top-m waiting times as MAX-PACK. It is clear MAX-PACK pack the transactions in the order of $x_1, x_2, \cdots, x_n$.

Assume by contradiction that there is another packing algorithm OP-PACK that can achieve larger fairness. Let OP-PACK packs the transactions in the order of $\sigma_1, \sigma_2, \sigma_n$, where σ is a permutation other than $(1, 2, \cdots, n)$. By definition, we have the following property:

$$\sum_{i=1}^{m} w_i > \sum_{i=1}^{m} w_{\sigma_i} \tag{4}$$

$$\forall 2 \le k < \lceil \frac{n}{m} \rceil, \sum_{i=1}^{km} w_i \ge \sum_{i=1}^{km} w_{\sigma_i} \tag{5}$$

$$\sum_{i=1}^{n} w_i = \sum_{i=1}^{n} w_{\sigma_i} \tag{6}$$

Assume the time to make a packed block to be committed to be t_p. Then the response times of the transactions using MAX-PACK are $w_1 + t_p, w_2 + t_p, \cdots, w_{m+1} + 2t_p, \cdots, w_n + \lceil \frac{n}{m} \rceil \cdot t_p$. The response times of the transactions using OP-PACK are $w_{\sigma_1} + t_p, w_{\sigma_2} + t_p, \cdots, w_{\sigma_{m+1}} + 2t_p, \cdots, w_{\sigma_n} + \lceil \frac{n}{m} \rceil \cdot t_p$. To

this end, the fairness of MAX-PACK and OP-PACK and their relationship are as follows:

$$\mathcal{J}_{Max-Pack} = \frac{(\sum_{i=1}^{n}(w_i + \lceil \frac{i}{m} \rceil \cdot t_p))^2}{n \cdot \sum_{i=1}^{n}(w_i + \lceil \frac{i}{m} \rceil \cdot t_p)^2} \tag{7}$$

$$\mathcal{J}_{OP-Pack} = \frac{(\sum_{i=1}^{n}(w_{\sigma_i} + \lceil \frac{i}{m} \rceil \cdot t_p))^2}{n \cdot \sum_{i=1}^{n}(w_{\sigma_i} + \lceil \frac{i}{m} \rceil \cdot t_p)^2} \tag{8}$$

$$\mathcal{J}_{OP-Pack} > \mathcal{J}_{Max-Pack} \tag{9}$$

Since the algorithms are running on the same set of transactions, we have

$$\sum_{i=1}^{n} w_i^2 = \sum_{i=1}^{n} w_{\sigma_i}^2 \tag{10}$$

$$\sum_{i=1}^{n}(w_i + \lceil \frac{i}{m} \rceil \cdot t_p) = \sum_{i=1}^{n}(w_{\sigma_i} + \lceil \frac{i}{m} \rceil \cdot t_p) \tag{11}$$

According to Eq. 7,8,9,11, we have:

$$\sum_{i=1}^{n}(w_i + \lceil \frac{i}{m} \rceil \cdot t_p)^2 > n \cdot \sum_{i=1}^{n}(w_{\sigma_i} + \lceil \frac{i}{m} \rceil \cdot t_p)^2 \tag{12}$$

Expand Eq. 12, we get:

$$\begin{aligned} &\sum_{i=1}^{n} w_i^2 + \sum_{i=1}^{n}(\lceil \frac{i}{m} \rceil \cdot t_p)^2 + 2\sum_{i=1}^{n}(w_i \cdot \lceil \frac{i}{m} \rceil \cdot t_p) > \\ &\sum_{i=1}^{n} w_{\sigma_i}^2 + \sum_{i=1}^{n}(\lceil \frac{i}{m} \rceil \cdot t_p)^2 + 2\sum_{i=1}^{n}(w_{\sigma_i} \cdot \lceil \frac{i}{m} \rceil \cdot t_p) \end{aligned} \tag{13}$$

According to Eq. 10,13, we have:

$$\sum_{i=1}^{n}(w_i \cdot \lceil \frac{i}{m} \rceil) > \sum_{i=1}^{n}(w_{\sigma_i} \cdot \lceil \frac{i}{m} \rceil) \tag{14}$$

Adding Eq. 4 and all the inequations in Eq. 5, we have

$$\sum_{i=1}^{\lceil \frac{n}{m} \rceil - 1} \sum_{j=1}^{im} w_j > \sum_{i=1}^{\lceil \frac{n}{m} \rceil - 1} \sum_{j=1}^{im} w_{\sigma_j} \tag{15}$$

Adding the inquestions in Eq. 14 and Eq. 15, we have:

$$\begin{aligned} \lceil \frac{n}{m} \rceil \sum_{i=1}^{n} w_i &= \sum_{i=1}^{\lceil \frac{n}{m} \rceil - 1} \sum_{j=1}^{im} w_j + \sum_{i=1}^{n}(w_i \cdot \lceil \frac{i}{m} \rceil) \\ &> \sum_{i=1}^{\lceil \frac{n}{m} \rceil - 1} \sum_{j=1}^{im} w_{\sigma_j} + \sum_{i=1}^{n}(w_{\sigma_i} \cdot \lceil \frac{i}{m} \rceil) \\ &= \lceil \frac{n}{m} \rceil \sum_{i=1}^{n} w_{\sigma_i} \end{aligned} \tag{16}$$

Obviously, Eq. 16 is contradictory with Eq. 6. As a result, the assumption does not hold and MAX-PACK achieves maximum fairness. □

Similarly, we can get the corollary that the larger the sum of the waiting times of the transactions is, the larger the fairness is. To this end, we can get the strategies to pack transactions with the largest, the 2-nd largest fairness and *etc.*. Suppose there are k nodes in the Blockchain network, then they can coordinate to use the strategies with the largest, the 2-nd largest, $\cdots$, and the k-th largest fairness to get the maximum throughput and moderate fairness. However, there is a still a gap towards finding the strategy with the k-th largest fairness. The gap is the KTH-SUM problem defined as follows.

Definition 5. KTH-SUM*: Given a set of n positive real numbers $X = \{x_1, x_2, \cdots, x_n\}$ and a positive integer $m < n$, there are $\binom{n}{m}$ distinct subsets of X of size m. Among the $\binom{n}{m}$ subsets, find the one with the k-th largest sum.*

Algorithm 1 An approaximate algorithm to find the subset of size m with k-th largest sum in a set X of n positive real numbers

```
a ← an array of size m
procedure APP-KTH-SUM(X, n, m, k)
    for tar ← m(m+1)/2 to ∞ do
        prevk ← k
        if DFS(n, m, k, 1, 0, 0, tar) then
            Sort X in decreasing order
            return x_{a[1]}, x_{a[2]}, ··· , x_{a[m]}
        end if
        if prevk = k then
            return FALSE
        end if
    end for
end procedure
procedure DFS(n, m, &k, d, p, sum, tar)        ▷ Here, k is passed by reference
    if d = m then
        if tar − sum > n then
            return FALSE
        end if
        a[d] ← tar − sum
        k ← k − 1
        if k = 0 then
            return TRUE
        end if
    end if
    for i ← p + 1 to ∞ do
        if sum + (2·i+m−d)·(m−d+1)/2 > tar then
            BREAK
        end if
        a[d] ← i
        if DFS(n, m, k, d + 1, i, sum + i, tar) then
            return TRUE
        end if
    end for
    return FALSE
end procedure
```

Actually, we do not need to solve the KTH-SUM problem exactly. Instead, we only need to find an approximate solution. Hence, we propose the algorithm APP-KTH-SUM as shown in Alg. 1. The intuition of the algorithm is shown in Fig. 6. We

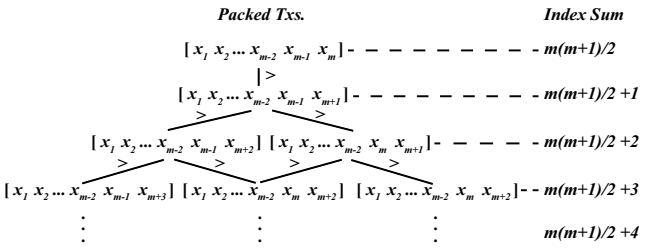

Fig. 6. Intuition for KTH-SUM problem

separate the subsets level by level. For a subset in a lower level, there must be a subset which has larger subset sum in the upper level. The level is determined by the sum of the index. In Alg. 1, we enumerate the sum of the index from the smallest possible one, *i.e.*, $\frac{m(m+1)}{2}$, to the infinity. For each target sum of the index, we use depth-first-search algorithm to find all the possible transaction combinations. For a target sum of the index, if there is not a single transaction combination whose index sum is the target, the procedure returns false, which means $k > \binom{n}{m}$. If k reaches 0, it means an approximate answer is found, and the corresponding transaction combination is returned.

Algorithm 2 Throughput-first and fairness-first packing algorithm running on node i

procedure TP&FAIR(X)
  $m \leftarrow$ the maximum number of transactions in a block
  $X' \leftarrow$ APP-KTH-SUM($X, |X|, m, i$)
  return X'
end procedure
procedure FAIR-FIRST(X)
  $m \leftarrow$ the maximum number of transactions in a block
  $X' \leftarrow$ APP-KTH-SUM($X, |X|, m, 1$)
  return X'
end procedure

Based on the proposed APP-KTH-SUM algorithm, we further propose two packing algorithms, *i.e.*, FAIR-FIRST and TP&FAIR, to coordinate the nodes in the Blockchain network. The algorithms are shown in Alg. 2. FAIR-FIRST is used when fairness is critical in the system while TP&FAIR sacrifices a little fairness for higher throughput. The two packing algorithms can be selected based on the required features of the Blockchain-based system. In our system, PHD-Chain is designed to use TP&FAIR for high throughput and moderate fairness while EMR-Chain is designed to use FAIR-FIRST since throughput is relatively less important. The intuition for the FAIR-FIRST packing algorithm is to let all the nodes work on the same transaction combination, which is of the maximum fairness. The intuition for the TP&FAIR packing algorithm is to let the nodes work on different transaction combinations, while can achieve top fairness respectively.

IV. SYSTEM IMPLEMENTATION AND EVALUATION

To demonstrate the effectiveness and practicability of BlocHIE, we implement BlocHIE in a minimal-viable-product version. As shown in Fig. 7, the implementation is divided into three layers, namely communication layer, Blockchain layer, and GUI layer.

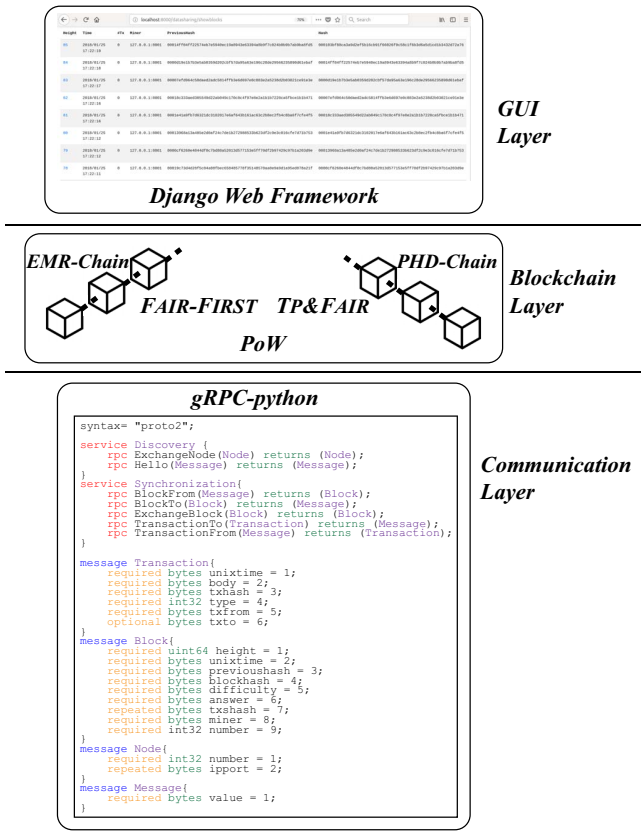

Fig. 7. Techniques for system implementation level by level

The bottom layer, *i.e.*, communication layer, is implemented using gRPC-python[1]. There are two services to support Blockchain-based system, *i.e.*, peer discovery service ("Discovery") and synchronization service ("Synchronization") as shown in Fig. 7. The "Discovery" service is used for discovering the nodes inside the Blockchain network. When a node is started, it will greet several static nodes (the same as bootnodes in Ethereum) and exchange the connectivity information with the static nodes. The block and transaction synchronization is achieved by the "Synchronization" service, which includes several remote procedure calls (RPCs) such as "BlockFrom", "BlockTo", "BlockFrom", "TransactionTo", and "TransactionFrom".

At the middle layer, two Blockchains, *i.e.*, EMR-Chain and PHD-Chain are implemented. The EMR-Chain employs the FAIR-FIRST transaction packing algorithm while the PHD-Chain utilizes the TP&FAIR transaction packing algorithm.

[1]https://grpc.io/

For the block committing algorithm, both EMR-Chain and PHD-Chain employs PoW.

Django web framework[2] is used in the top layer, *i.e.*, the GUI layer. It opens an HTTP port and presents HTML pages using the port. In this way, the users can submit data following the HTTP protocol. When some data is submitted, we invoke the methods on Blockchain layer to fulfill Blockchain functions.

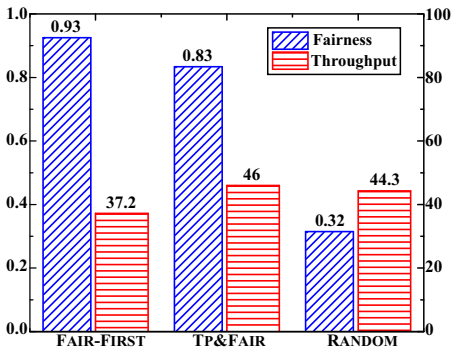

Fig. 8. Performance comparison of different packing algorithms

After implementation, we measure the performance of BlocHIE with 8 nodes. Each node is serving as both server and client, *i.e.*, sending requests and packing transactions at the same time. The frequency of sending requests of each node is around 7 tx/s. Moreover, we set the number of transactions inside a block, *i.e.*, m, to be 56, which is the approximate transaction generating rate. We compare the performance of TP&FAIR, FAIR-FIRST, and RANDOM concerning both fairness and throughput. Here, the RANDOM packing algorithm refers to the algorithm that randomly pick m transactions from pool. The result is shown in Fig. 8. We observe that in terms of fairness, both FAIR-FIRST and TP&FAIR outperform RANDOM significantly. Specifically, they achieve up to 2.9x and 2.6x higher fairness than RANDOM respectively. From the perspective of throughput, TP&FAIR achieves the maximum, *i.e.*, 46 tx/s, which improves FAIR-FIRST over 23.6%.

V. CONCLUSION

In this paper, we propose BlocHIE, a Blockchain-based platform for healthcare information exchange. We consider two kinds of healthcare data, *i.e.*, electronic medical records and personal healthcare data, and analyzed the different requirements to store and share them. Based on the analysis, we architect BlocHIE on two loosely-coupled Blockchains, *i.e.*, EMR-Chain for electronic medical records and PHD-Chain for personal healthcare data. Inside EMR-Chain, we integrate the techniques of off-chain storage and on-chain verification to take good care of privacy and authenticability. Moreover, we propose two transaction packing algorithms to enhance the system throughput and the fairness among users. Finally, the implementation and evaluation indicate the practicability and effectiveness of BlocHIE.

[2]https://www.djangoproject.com/

ACKNOWLEDGMENTS

This work is supported by Huawei Technologies Co. Ltd. with project code P15-0540 and RGC CRF with project number CityU C1008-16G.

REFERENCES

[1] B. E. Dixon and C. M. Cusack, "Measuring the value of health information exchange," in *Health Information Exchange*. Elsevier, 2016, pp. 231–248.
[2] X. Liu, K. Li, G. Min, Y. Shen, A. X. Liu, and W. Qu, "Completely pinpointing the missing rfid tags in a time-efficient way," *IEEE Transactions on Computers*, vol. 64, no. 1, pp. 87–96, 2015.
[3] S. Jiang, J. Cao, Y. Liu, J. Chen, and X. Liu, "Programming large-scale multi-robot system with timing constraints," in *Computer Communication and Networks (ICCCN), 2016 25th International Conference on*. IEEE, 2016, pp. 1–9.
[4] S. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, and K.-S. Kwak, "The internet of things for health care: a comprehensive survey," *IEEE Access*, vol. 3, pp. 678–708, 2015.
[5] F. S. Collins and H. Varmus, "A new initiative on precision medicine," *New England Journal of Medicine*, vol. 372, no. 9, pp. 793–795, 2015.
[6] J. Zhou, Z. Cao, X. Dong, and X. Lin, "Tr-mabe: White-box traceable and revocable multi-authority attribute-based encryption and its applications to multi-level privacy-preserving e-healthcare cloud computing systems," in *INFOCOM*. IEEE, 2015, pp. 2398–2406.
[7] N. Grozev and R. Buyya, "Inter-cloud architectures and application brokering: taxonomy and survey," *Software: Practice and Experience*, vol. 44, no. 3, pp. 369–390, 2014.
[8] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.
[9] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum Project Yellow Paper*, vol. 151, 2014.
[10] A. Lei, H. Cruickshank, Y. Cao, P. Asuquo, C. P. A. Ogah, and Z. Sun, "Blockchain-based dynamic key management for heterogeneous intelligent transportation systems," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1832–1843, 2017.
[11] H. Hou, "The application of blockchain technology in e-government in china," in *ICCCN*. IEEE, 2017, pp. 1–4.
[12] M. Turkanović, M. Hölbl, K. Košič, M. Heričko, and A. Kamišalić, "Eductx: A blockchain-based higher education credit platform," *IEEE Access*, 2018.
[13] Q. Xia, E. B. Sifah, K. O. Asamoah, J. Gao, X. Du, and M. Guizani, "Medshare: Trust-less medical data sharing among cloud service providers via blockchain," *IEEE Access*, vol. 5, pp. 14 757–14 767, 2017.
[14] Z. Shae and J. J. Tsai, "On the design of a blockchain platform for clinical trial and precision medicine," in *ICDCS*. IEEE, 2017, pp. 1972–1980.
[15] M. Jakobsson and A. Juels, "Proofs of work and bread pudding protocols," in *Secure Information Networks*. Springer, 1999, pp. 258–272.
[16] "Nxt: The blockchain application platform via proof-of-stake," https://nxtplatform.org/, accessed: 2018-01-28.
[17] "Slimcoin: A peer-to-peer crypto-currency with proof-of-burn," https://slimcoin-project.github.io/, accessed: 2018-01-28.
[18] "Litecoin: Open source p2p digital currency," https://litecoin.org/, accessed: 2018-01-26.
[19] "Cryptonight hash function," https://cryptonote.org/cns/cns008.txt, accessed: 2018-01-28.
[20] "Monero [xmr]: a brand new uprising cryptocurrency which originates from bitmonero," http://dogecoin.com/, accessed: 2018-01-28.
[21] R. Jain, D.-M. Chiu, and W. R. Hawe, *A quantitative measure of fairness and discrimination for resource allocation in shared computer system*. Eastern Research Laboratory, Digital Equipment Corporation Hudson, MA, 1984, vol. 38.
[22] K. Croman, C. Decker, I. Eyal, A. E. Gencer, A. Juels, A. Kosba, A. Miller, P. Saxena, E. Shi, E. G. Sirer *et al.*, "On scaling decentralized blockchains," in *International Conference on Financial Cryptography and Data Security*. Springer, 2016, pp. 106–125.
[23] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun, "On the security and performance of proof of work blockchains," in *CCS*. ACM, 2016, pp. 3–16.