applied sciences

MDPI

Article

MedChain: Efficient Healthcare Data Sharing via Blockchain

Bingqing Shen, Jingzhi Guo * and Yilong Yang

Faculty of Science and Technology, University of Macau, Macau SAR 999078, China;
daniel.shen@connect.umac.mo (B.S.); yylonly@gmail.com (Y.Y.)
* Correspondence: jzguo@umac.mo; Tel.: +853-8822-4360

Received: 27 November 2018; Accepted: 28 December 2018; Published: 22 March 2019

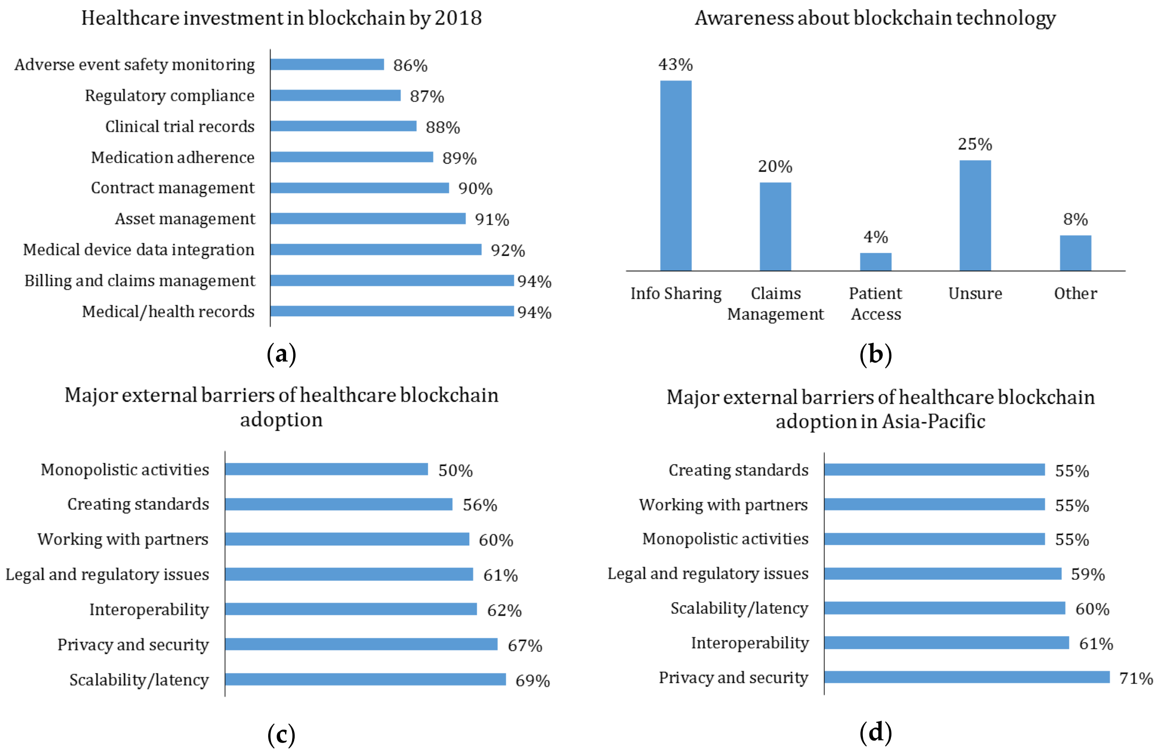
Abstract: Healthcare information exchange is an important research topic, which can benefit both healthcare providers and patients. In healthcare data sharing, many cloud-based solutions have been proposed, but the trustworthiness of a third-party cloud service is questionable. Recently, blockchain has been introduced in healthcare record sharing, which does not rely on trusting a third party. However, existing approaches only focus on the records collected from medical examination. They are not efficient in sharing data streams continuously generated from sensors and other monitoring devices. Today, IoT devices have been widely deployed and sensors and mobile applications can monitor patients' body conditions. The collected data are shared to laboratories and institutions for diagnosis and further study. Moreover, existing approaches are too rigid to efficiently support metadata change. In this paper, an efficient data-sharing scheme is proposed, called MedChain, which combines blockchain, digest chain, and structured P2P network techniques to overcome the above efficiency issues in the existing approaches for sharing both types of healthcare data. Based on MedChain, a session-based healthcare data-sharing scheme is devised, which brings flexibility in data sharing. The evaluation results show that MedChain can achieve higher efficiency and satisfy the security requirements in data sharing.

Keywords: blockchain; healthcare data; electronic health record; data stream; healthcare information exchange; data sharing; peer-to-peer; decentralization; digest chain

1. Introduction

In the industry, healthcare is an important sector. In addition to traditional medical examination, patient's body states, including heart rate, diabetes, electroencephalogram, and other vital biomedical signals can be monitored by applying various medical tracking devices for diagnosis [1,2] or health quality improvement [3]. Sharing of such a huge amount of data among organizations can facilitate medical diagnosis, biomedical research, and policy making. For example, a doctor may need the medical history of a patient stored in different hospitals when deciding on the best treatment. Moreover, this market will create a big impact in the economy [4]. In healthcare data sharing, user trust is a key factor for success. Any deficiency could result in distrust among patients towards the e-healthcare market [5].

For scalability, flexibility, and economic reasons, some cloud-based healthcare data sharing schemes [6] have been proposed through data encryption and operation anonymization. However, users are always hesitant to transfer their private and sensitive data to the cloud due to its potential risks [7]. Recently, blockchain-based solutions have been widely discussed [8]. Blockchain can achieve many compelling features without trusting a third party. The most important one is tamper-proof, which is achieved by the special data structure and the consensus mechanism. Moreover, data stored on a blockchain are highly reliable and available through replication. With the above advantages, the market evidence has shown the potential of the blockchain-based solutions from both a profit shift (Figure 1a) and management awareness (Figure 1b).



(**a**) (**b**) (**c**) (**d**)

Figure 1. The potential of blockchain-based solution and major external barriers of adoption in the healthcare industry: (**a**) Healthcare investment in blockchain by 2018 [9]. (**b**) Awareness about blockchain technology among medical practice administrators and executives [10]. Major external barriers of healthcare blockchain adoption in (**c**) the world [11] and (**d**) Asia-Pacific [12], answered by respondents.

However, there are still some challenges in healthcare blockchain adoption. Within the major technical barriers, efficiency (including scalability and latency) is one of the top concerns (Figure 1a,d). Existing blockchain-based solutions are less efficient in sharing data streams from Internet-of-Things (IoT) devices. First, the data from IoT devices are time-series data streams, such as ECG signals. In storage, they are cut into many data chunks. Different from a single healthcare record, accessing a data stream requires access to all the data chunks. Existing schemes are designed to verify the integrity for a single record. For verifying a data stream, they need to download the digest of each chunk and check the integrity for all of them, which is inefficient, especially for accessing a long stream. Second, the description of data could be changed from time to time. For example, a new tag or category could be added to the existing data. Managing mutable information on blockchain either needs to add new blocks, which consumes more storage space, or needs to re-write the entire blockchain. Moreover, data sharing is a dynamic process. When a sharing is over, some temporary information, such as the location of the actual data and the cryptographic keys used for security purpose, needs to be cleared, which prevents them from littering the storage space. Existing schemes are incapable of providing such a mechanism, since the information stored on blockchain is immutable. When IoT is applied in healthcare, it can be expected that more data will be shared and storage space overhead will become an outstanding problem. Thus, efficiently sharing healthcare data has become an emergent problem.

To solve the problem, a new healthcare data sharing solution called MedChain, is proposed, which introduces the following novelties. First, it leverages two separate decentralized networks: a blockchain network and a peer-to-peer (P2P) storage network. The blockchain network stores the fingerprint of data, session, and operation, such as data digest, which are immutable, while the P2P storage network stores the description of data and session, which are mutable. By separating the mutable part to another P2P network, data description can be easily updated without bringing additional overhead to the immutable part. Second, a new data structure, called digest chain, is proposed to facilitate data stream verification. By concatenating the chunks of digest of the same data stream into a chain, the plethoric digest download and integrity check problem can be solved. Third, based on the proposed architecture, a session is introduced in the data sharing process for packaging and removing the mutable information, which can largely reduce storage overhead. Moreover, the security properties of the system are validated, which is crucial in healthcare data sharing. Thus, compared with state-of-the-art works, this paper has the following contributions.

(1) It has described a MedChain data-sharing framework for flexibly managing different types of information derived from healthcare data.
(2) It has also devised a chained digest creation approach to efficiently check the integrity of shared medical IoT data stream.
(3) It has provided a session-based data-sharing scheme for achieving efficiency improvement and the security, integrity, auditability, and privacy-preservation goals.

The rest of this paper is devised as follows. Section 2 reviews the related work. Section 3 describes the proposed model. Section 4 presents the data-sharing scheme based on the proposed model. The analysis of the security properties and the experiment results are discussed in Section 5. Lastly, Section 6 concludes the paper.

2. Related Works

In the medical and healthcare sector, legacy systems generally only exchange medical resources internally [13] and are not interoperable with external systems [14]. Yet, evidences [15,16] show numerous benefits from connecting these systems for integrated and improved healthcare, calling health informatics researchers for an interconnection solution among different organizations. One of the most important challenges is inter-organizational data sharing [17], demanding the medical data collected by one healthcare provider to be securely accessible to other entities, such as a doctor or a research organization.

Cloud computing is considered to be an immediate solution for medical data storage and sharing [6,18,19] because it is scalable, highly available, and flexible in pricing. Yet, due to the privacy and confidentiality requirements in healthcare data sharing, additional security means must be applied to mitigate the risks of cloud sourcing in healthcare and public health industry [7]. Many studies [4,20] have been conducted to address the security and privacy issues. For example, Reference [21] analyzes the security and privacy issues in the access and management of EHRs. References [22,23] explores the solution for search over encrypted health records in a public cloud. Reference [24] studies the identity exposure problem in cloud-based healthcare applications and proposes an anonymous authentication approach. However, cloud services are not fully trusted by users, due to the infrastructure security, data ownership, and vendor lock-in issues in cloud sourcing [7,25,26].

Our experience [18] shows the importance of inter-organizational healthcare data sharing. In Macau, for example, the hemodialysis center does not provide a platform for examination result sharing. Patients have to carry the paper/CD-based results to hospitals for diagnosis. Our previous work [18] provides a hybrid cloud-based medical resource sharing solution for electronic healthcare record (EHR) sharing, which relies on the trust of an external cloud service provider. This work replaces the cloud with two decentralized networks to remove the needs of the external party.

To win user trust, blockchain-enabled medical record sharing has been extensively discussed within the last two years [8]. Yue et al. reported their early adoption in Reference [27]. They treat blockchain as a database for storing health records. Likewise, References [28,29] use permissioned blockchain as a repository of health data to attain access accountability and data integrity. These approaches need to transfer the actual data to blockchain servers, bringing extra overhead in data transmission. Reference [30] uses blockchain to store only the address of shared data. However, data sharing is not sufficiently discussed.

Recently, MeDShare [31] enables the sharing of patient medical data to third-party research institutes in cloud repositories. It uses a smart contract for data access auditing and access control. However, MeDShare also shares the user trust issue in cloud computing. In another design for medical image sharing [32], the source files are still stored at the end of healthcare providers and only the URLs for file access are stored on the blockchain. Unfortunately, these solutions do not provide an efficient approach for data search over the blockchain. Since blockchain is not search-friendly, looking up a specific record would be very slow with the increase of data.

MedRec [33] and MedBlock [34] provide a breadcrumb mechanism for a record search. Breadcrumbs maintain the address of blocks containing the records of a patient, grouped by a healthcare provider or department. Reference [35] implements the keyword-based search over encrypted data for record matching. However, they still share three limitations. First, a blockchain only stores immutable records. Yet, the actual locations of data are likely to be changed. When a URL is modified, a new block containing the new URL has to be generated rather than modifying the old block (due to content immutability of blockchain). Second, their data sharing schemes do not reclaim the space after a sharing. Thus, a data sharing session also takes a new block [35]. Both limitations will result in high storage overhead. Moreover, the breadcrumbs mechanism is only efficient in looking up a single record. However, in data stream sharing, it has $O(n)$ communication overhead, where n is the number of data chunks.

3. MedChain Model

This section discusses the MedChain model. First, the overall system architecture and data representations are introduced. Then, the details of the blockchain service and the directory service are introduced.

3.1. System Architecture

MedChain is constructed on a decentralized network, which connects all healthcare providers, including hospitals, medical centers, clinics, and healthcare corporates. The MedChain network contains two types of peer nodes: super peers and edge peers. A simple but reasonable peer selection approach is adopted. Super peers consist of the servers from large healthcare providers, such as national hospitals, which are more capable in computing and storage, providing the main infrastructure of data sharing. The edge nodes are the servers from small providers such as community clinics, which only store the actual patient data. The overall network model is shown in Figure 2, in which a trusted Certificate Authority (CA) is employed for certificating and validating public keys. It contains two sub-networks: a blockchain service network and a directory service, which store different types of information derived from healthcare data.

The resources of a super peer is divided into three modules: blockchain service, directory service, and healthcare database (HDB), as shown in Figure 3. The blockchain server maintains a complete blockchain for verifying data integrity and auditing activities. The directory server maintains the inventory of user healthcare data, maps them to the actual location of storage, and manages sessions for data sharing. The servers of the two types on all super-peers form two sub-networks. The last component, HDB, stores the actual healthcare data of patients. Note that MedChain does not require the healthcare provider to migrate the actual data to the new system. Instead, it provides the reference to the data in the legacy system for access. Thus, it is an integrated solution, which can reduce the difficulty of adoption, since most healthcare providers are reluctant to migrate their data to a new platform.

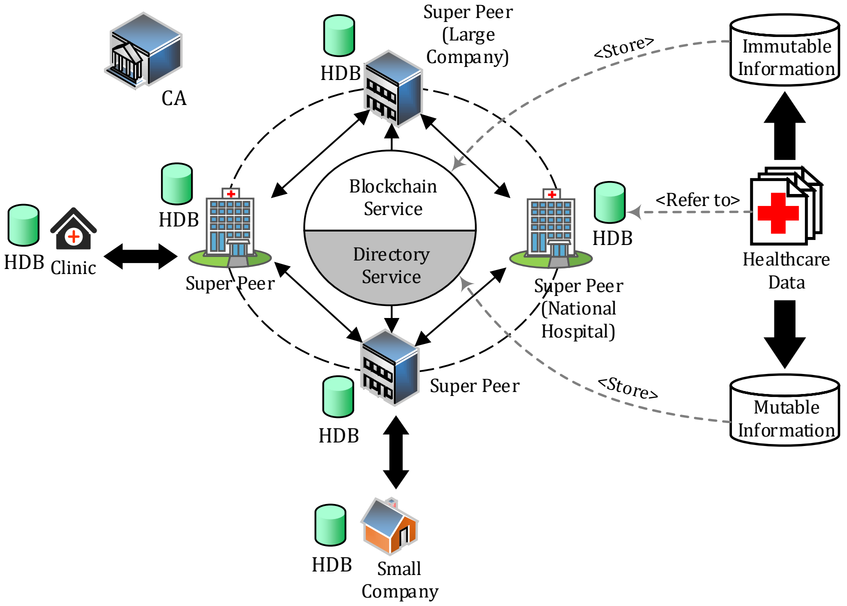


Figure 2. MedChain architecture.

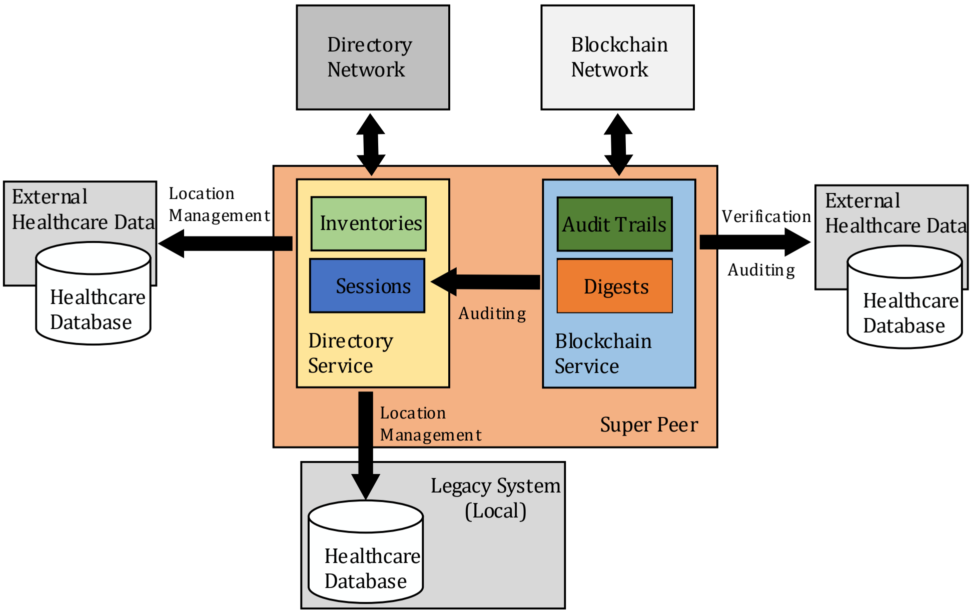


Figure 3. Modules of a super peer.

3.2. Preliminaries

Before describing the details of the system, some preliminaries are introduced first to facilitate the subsequent description.

3.2.1. Healthcare Data

Healthcare data describes a patient's health state, in the form of electronic healthcare record, e.g., a CT image, or data stream collected by a medical sensor, e.g., ECG signal. Each healthcare record/data stream is assigned a unique ID (*DID*). A data stream is a time series record, which maps each sampled value to a time label. A data stream is cut and saved into multiple data chunks. One chunk records the sampled value within a time interval. Thus, each chunk has a start time and an end time labeling the range of the data chunk. All data chunks of the same data stream share the same *DID*, but different chunk indexes.

The derived information from healthcare data can be classified into immutable information and mutable information. The immutable information maintains the security, integrity, and authenticity of data, which includes the identity of the related users, the digest of the data, the endorsement (e.g., digital signature) from the data provider, and the data collection range of a stream. The mutable information facilitates data query and access, which includes the description, the tags, and the network location (i.e., URL) of the actual data.

For healthcare data, the identity of the data, the network location, and the start time, and end time of the chunk for data stream belong to the privacy of a patient, as well as the actual data. Thus, privacy protection is needed in MedChain to disassociate *DID*, data location, and the actual data with data owner (i.e., patient).

3.2.2. User Roles

MedChain contains three user roles: patient, requester, and healthcare provider.

(1) Patient: shares their data through MedChain with, for instance, a doctor, an insurance company, or a research center for medical consultancy.
(2) Requester: who could be, e.g., a doctor, asks a patient to share some of her healthcare data through MedChain.
(3) Healthcare provider: maintains the actual healthcare data of patients.

3.2.3. Cryptographic Keys

Cryptographic keys are applied to fulfill the security and privacy requirements of the system design. The keys to achieve confidentiality over untrusted communication channels and storage space are highlighted as follows.

- Patient public-private key pair ($PK_{pa\mathrm{t}}/SK_{pat}$).
- Healthcare provider public-private key pair (PK_{sp}/SK_{sp}).
- Requester public-private key pair (PK_{req}/SK_{req}).
- Inventory secret S_{data}: a symmetric secret key for patient inventory access, generated by a healthcare provider.
- Section secret K_{sec}: a symmetric secret key for accessing a section of a session, generated by the patient.

In the implementation, MedChain employs elliptic curve cryptography (ECC) [36] for public-private key pair generation. Moreover, the following notation of cryptographic functions are employed for description consistency.

- $E_{key}(m)$: encrypts message m with *key*.
- $D_{key}(m)$: decrypts message m with *key*.
- $H(m)$: creates a hash code/digest of content m.

For authentication and an integrity purpose, a message m sent by a user is signed with the user's private key to generate the signature appended on m. Later, the signature will be verified by the public key of the user.

3.2.4. Other Notations

Addition to the above denotations, the data formats of blockchain, directory, and messages are defined with basic set and logic notations to facilitate description. Throughout the paper, the symbols for representing the relations of elements are listed in Table 1.

Table 1. Notations.

Notation	Description
$:=$	Definition
$\langle \cdots \rangle$	Tuple, representing data or message format
$\{\cdots\}$	Set, representing one or more items of the same type
$(, \cdots)$	Optional element(s)
$\mid$	Exclusive disjunction (XOR)
$\Vert$	String concatenation operation
$.$	Membership
$\rightarrow$	Message transmission
$\leftarrow$	Assignment

3.3. Blockchain Service

A blockchain is a distributed ledger recording the events of healthcare data generation and data sharing. It is composed of a growing number of blocks. In MedChain, each block contains one or more events identified by event hash. An event hash is computed by hashing the event content, as the event fingerprint. A block also has a block header, which contains:

- Merkle Root. The root of the Merkle tree [37] constructed by all the event hashes in the block.
- Timestamp: The time of when the block is created.
- Block hash: The hash code computed based on the hash of the last block, the Merkle root, and the timestamp.

The cascaded hash computing at the event level (event hash), the block level (Merkle root), and the chain level (block hash) ensures the content immutability of a blockchain. If someone wants to modify the block information, he/she has to modify the entire chain. Yet, any tampering with the content can be easily detected by re-generating the hash codes and comparing them with the original ones. Thus, the blockchain is effective with storage immutable information, but weak when storing mutable information. The MedChain blockchain structure is illustrated in Figure 4, which shows a block with two events.

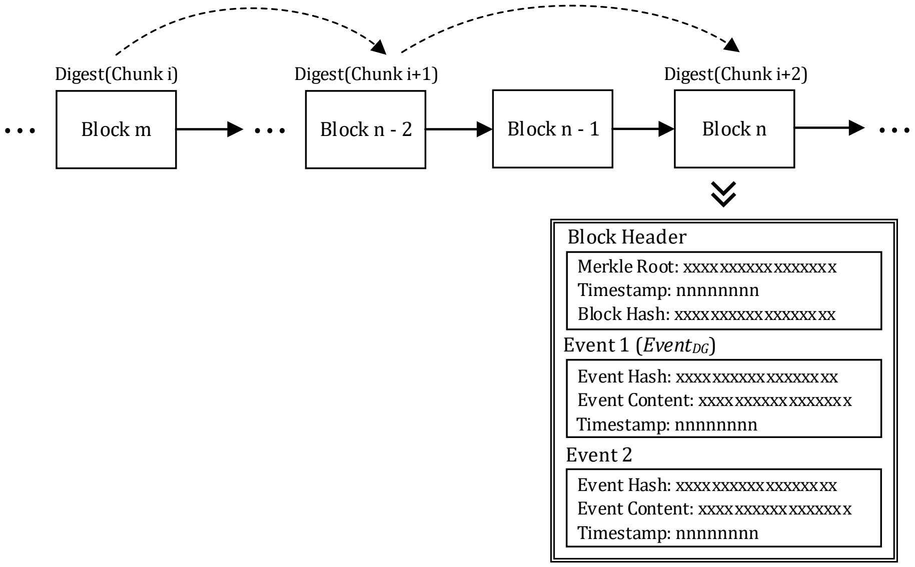

Digest(Chunk i)
Digest(Chunk i+1)
Digest(Chunk i+2)
Block m
Block n - 2
Block n - 1
Block n
Block Header
Merkle Root: xxxxxxxxxxxxxxxxx
Timestamp: nnnnnnnn
Block Hash: xxxxxxxxxxxxxxxxxx
Event 1 (Event_DG)
Event Hash: xxxxxxxxxxxxxxxxxx
Event Content: xxxxxxxxxxxxxxxxx
Timestamp: nnnnnnnn
Event 2
Event Hash: xxxxxxxxxxxxxxxxxx
Event Content: xxxxxxxxxxxxxxxxx
Timestamp: nnnnnnnn

Figure 4. An illustration of the MedChain blockchain structure with a digest chain.

In MedChain, the blockchain service manages the immutable information of healthcare data. Two types of events are recorded on a blockchain: data generation event and session creation event. The immutable information and the format of events will be elaborated in detail in the following paragraphs.

Data Generation Event (*Event*$_{DG}$). A data generation event is created when a healthcare record file or a data chunk file is generated. It contains *DID*, chunk index, patient public key, healthcare provider public key and signature (*Sign*$_{sp}$), data digest, and the reference (*Ref*$_L$) to the last chunk location (identified by the block hash (*BH*$_L$) and the event hash (*EH*$_L$)) on the blockchain (for data stream only) and event type. As shown in Def. (1), the privacy-sensitive items, including *DID*, chunk index, and *PK*$_{pat}$, are encrypted with *PK*$_{pat}$ for disassociating a patient with a healthcare provider. The data digest is created by hashing the data file for data integrity check. See the sub-section below for the details of the digest chain.

$$\begin{aligned}
&Event_{DG}\langle Content, PK_{sp},\ Digest, Sign_{sp}(, Ref_L),\ Type\rangle\\
&ContentE_{PK_{pat}}(DID(,\ Index), PK_{pat})\\
&DigestH(Data) \mid H(Chunk(i) \parallel Digest_{i-1})\\
&Ref_L\langle BH_L, EH_L\rangle\\
&Type \leftarrow {''AddData''}
\end{aligned} \tag{1}$$

Session Creation Event (*Event*$_{SC}$). A session creation event is created when a patient grants the access of some of the healthcare data to a requester. It contains a list of *DID*s. For the data stream, the start time (*st*) and end time (*et*) can be specified as an access constraint. It also contains the public key of the patient and the requester for identifying the session participants, as well as the signature of the patient (*Sign*$_{pat}$). The formal definition of the session creation event is shown in Def. (2). For de-identification, *DID*, *st*, *et*, and *PK*$_{req}$ are encrypted with *PK*$_{pat}$. The session digest is generated by hashing the *DID*s and the time constraints of the shared data and the session participants' public keys.

$$\begin{aligned}
&Event_{SC}\langle Content, PK_{pat}, Digest, Sign_{pat},\ Type\rangle\\
&ContentE_{PK_{pat}}(\{DID(, st,\ et)\}, PK_{req})\\
&DigestH(\{DID(, st,\ et)\} \parallel PK_{req} \parallel PK_{pat})\\
&Type \leftarrow {''CreateSession''}
\end{aligned} \tag{2}$$

The blockchain servers run the blockchain service on all super peers, which collectively provide a consortium blockchain network [38]. Each blockchain server maintains a complete blockchain and they run a distributed consensus algorithm to collectively determine the content of the next block. MedChain is not coupled with a particular consensus protocol. The current implementation adopts BFT-SMaRt [39] for its simplicity. It can be easily changed to another protocol with minimal adaptation, such as Proof-of-Stake [32].

Digest Chain

To efficiently verify the data stream, the digest generation algorithm is modified. For a data stream, the digest of the *i*th chunk is created by hashing ($H(\cdot)$) the content, which consists of the chunk data (*Chunk*(i)) and the digest of the last chunk ($Digest_{i-1}$), which was formulated by the $H(Chunk(i) \parallel Digest_{i-1})$. Then, multiple digests of the chunks covering a continuous sampling time period forms a digest chain. As illustrated in Figure 4, the chunks of a digest chain could be distributed on different blocks and they are connected with the reference *Ref*$_L$ in the *Event*$_{DG}$. Thus, to generate a new block, only the block containing the digest of the last chunk needs to be retrieved through *Ref*$_L$ for querying the digest of the last chunk.

It is important that, with a digest chain, downloading and verifying the digest of one chunk can check the integrity of all the previous chunks. If the last chunk is valid, the integrity check can stop. Otherwise, it will repeat until a valid chunk is found. This mechanism allows early termination in the data integrity check, which increases system responsiveness in data stream access. Note that a digest chain does not need to be verified by the blockchain servers in new block generation. It can be left to data access, which is efficient especially for using less data.

3.4. Directory Service

The directory service is also an important component in the MedChain model, managing the mutable information of healthcare data for data identification, location, and sharing. MedChain provides two types of directories: patient inventory and session. The mutable information and the format of directories are introduced below in detail.

Inventory. An inventory records all the description details of a patient's healthcare data maintained by a healthcare provider. As defined in Def. (3), it includes the inventory ID (*InvID*), the inventory content (*InvContent*), and the encrypted inventory secret for healthcare providers (ES_{sp}) and patients (ES_{pat}). The inventory ID uniquely identifies the inventory with the association of the patient and the healthcare provider. For a privacy purpose, such association is obfuscated by the inventory secret S_{data}. The inventory content contains a list of data descriptions, including the *DIDs*, metadata (e.g., the summary and the tags of the data), the locations of the actual data (i.e., URLs), and the references (*Ref* or Ref_L) to the blockchain for data integrity check. For a single record, *Ref* refers to the location (identified by the block hash (*BH*) and event hash (*EH*)) of the $Event_{DG}$ on the blockchain. For a data stream, Ref_L refers to the location of the last chunk $Event_{DG}$.

$$\begin{aligned}
&Inventory\langle InvID, InvContent,\ ES_{sp}, ES_{pat}\rangle\\
&InvIDH(PK_{user} \parallel S_{data})\\
&InvContentE_{S_{data}}(\{DID, Metadata,\ URL, Ref|Ref_L\})\\
&Ref\langle BH, EH\rangle\\
&Ref_L\langle BH_L, EH_L\rangle\\
&ES_{sp|pat}E_{PK_{sp}|PK_{pat}}(S_{data})
\end{aligned} \tag{3}$$

A patient may have multiple inventories from different healthcare providers (i.e., different S_{data}). In addition, a provider can divide the inventory of a patient into multiple inventories, e.g., according to the hospital departments, which provides more flexibility in inventory storage and management.

Session. A session describes a data sharing between a patient and a third party, e.g., a doctor, called a requester. Def. (4) shows the session definition. It includes a session ID (*SID*) and multiple sections. Session ID is the event hash of the corresponding session creation event in the blockchain. A session contains the directory of all the shared data, but it divides them into multiple sections according to patient inventories to facilitate management. Actually, each section contains some of the entries copied from the corresponding inventory (after decryption by S_{data}). Thus, the section ID is the *InvID* of the inventory. Each entry of a section is composed of the *DID*, the metadata, the URL of the actual data, and the reference (*Ref* or Ref_L) to the $Event_{DG}$ in the blockchain. For the data stream, the range of data allowed to access can also be specified (i.e., by *st* and *et*). In this case, Ref_L refers to the last shared chunk. For privacy, the entries of a section are encrypted by the section secret K_{sec}. K_{sec} is shared to the requester for accessing the shared data. It is also shared with the corresponding healthcare provider for metadata and URL update and data access verification. To securely distribute the section secret, K_{sec} is encrypted with the public key of the section participants to EK_{sp}, EK_{req}, and EK_{pat}. Notably, different sections and sessions are encrypted with different K_{sec}.

$$\begin{array}{l} Session\langle SID, \{Section\}\rangle \\ Section\langle SectionID, \{Section\}\rangle \\ SIDSCEvent.EventHash \\ SectionIDInvID \\ Section\langle E_{K_{sec}}(\{Entry\}), EK_{sp}, EK_{req}, EK_{pat}\rangle \\ Entry\langle DID,\ Metadata, URL(,st,et), Ref_L)\rangle \\ Ref_L\langle BH, EH\rangle|\langle BH_L, EH_L\rangle \\ EK_{sp|req|pat} E_{PK_{sp}|PK_{req}|PK_{pat}}(K_{sec}) \end{array} \quad (4)$$

Besides the blockchain service, super peers are also connected to provide the directory service. For directory lookup, MedChain employs Chord [40] to map a directory ID (i.e., *InvID* or *SID*) to a server ID, which can also be replaced with any other P2P routing protocol implementing the distributed hash table (DHT) function [41].

Search over Blockchain

In MedChain, a healthcare record or data stream is identified by its unique identity (*DID*). On the blockchain, however, *DID* is encrypted together with patient identity (i.e., PK_{pat}) in the *Content* section for a privacy purpose (see Def. (1)). Given a *DID*, the entire blockchain has to be downloaded, traversed, and decrypted to find the record as in some existing works.

The directory service can be treated as the index of the events on the blockchain, providing an efficient way for search over blockchain. Like the breadcrumbs mechanism in References [33,34], each patient inventory contains a reference (either *Ref* or Ref_L) to the corresponding record on the blockchain (see Def. (3)). A reference consists of a block hash (*BH*) and an event hash (*EH*). On a blockchain server, the events are stored on a database indexed first by block hash and then by event hash. Thus, an efficient search algorithm (e.g., binary search) can be applied for event lookup. If the binary search is applied, the search complexity is $O(log(p) \cdot log(q))$ where p is the number of blocks and q is the number of events in a block, which is much lower than $(p \cdot q)$ if the entire blockchain is traversed (as in References [31,32]). The same search process can be applied to event lookup for a session, since inventory and session share the same data structure in healthcare data description (Def. (3) and Def. (4)).

The directory service maps a data stream to the last chunk generation event on the blockchain with the last chunk block hash (BH_L) and event hash (EH_L) (see Def. (3)). On the blockchain, each chunk, except the first one, links to the previous chunk of the same stream by referring to its block hash and event hash (see Def. (1)). Thus, to download all the chunks of a data stream on the blockchain, a client first locates the position of the last chunk (by referring to the Ref_L in Def. (3)) and then recursively search the previous chunk (by referring to the Ref_L in Def. (1)) until the first one.

A more fine-grained and user-friendly chunk search approach can be designed by adding the chunk start time and end time into each chunk generation events on the blockchain to support chunk-level data search, which will be implemented in our future work.

4. Healthcare Data Sharing

This section describes the session-based healthcare data-sharing scheme in detail, which includes data generation, session management, and key management. The first two processes describe the main activities involved in data sharing, while the third part renders an efferent key management design, since many types of keys are involved in the scheme.

4.1. Data Generation

The data generation process includes adding the data generation event to the blockchain service and the data description to the directory service. Figure 5 illustrates this procedure, which can be described with the following steps.

- *Step 1.* The healthcare provider collects the healthcare data from a patient through medical devices and sensors.
- *Step 2–3.* The healthcare provider creates an $Event_{DG}$ and sends it to the blockchain. The blockchain service adds the $Event_{DG}$ to a new block together with other received events, and then replies the block hash and the event hash to the provider.
- *Step 4.* The healthcare provider adds a new entry to the patient's inventory for describing the data. If the inventory does not exist, a new inventory will be created. Later, the patient can access it by decrypting the inventory with S_{data}, which is retrieved by decrypting the corresponding ES_{pat} with his/her SK_{pat}.

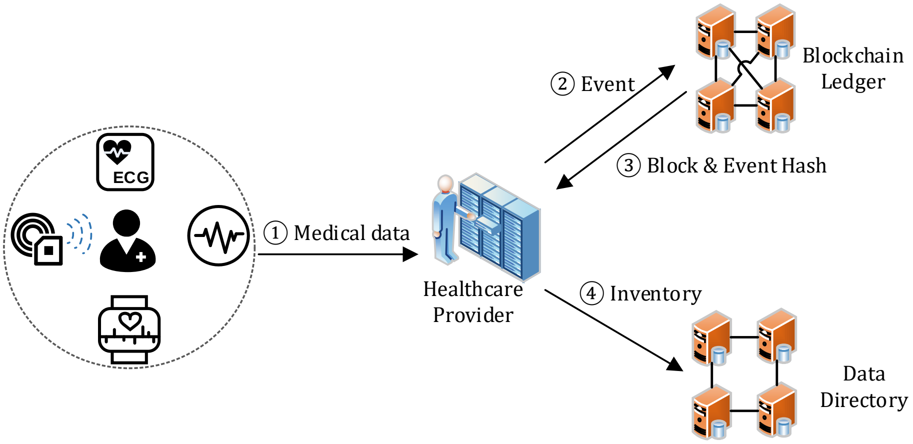


Figure 5. Process of new healthcare data generation.

4.2. Session Management

After the information of the data is added into the blockchain service and the directory service, a patient can grant access to his/her healthcare data to a requester through a session. The complete data sharing process is shown in Figure 6.

- *Step 1–2.* As per request, the patient selects the data from his/her inventory for sharing, creates a session with the selected data descriptions and the PK_{req}, encrypts them with SK_{pat}, generates the session digest to maintain session integrity, creates an $Event_{SC}$, and sends it to the blockchain service. The blockchain service adds the $Event_{SC}$ to a new block together with other received events, and appends the block to the end of the blockchain. Then, it sends back the *SID* (i.e., the event hash of $Event_{SC}$) to the patient.
- *Step 3–4.* The patient then creates a session in the directory service with the received *SID*, and notifies the requester and the healthcare provider of the session by sending them the *SID*. He/she uses K_{sec} to encrypt the content of the session and then PK_{req} to encrypt the session key.
- *Step 5–6.* With the *SID*, the requester can find and access the session. After decrypting the session key (K_{sec}) with his/her SK_{req} and the session with the K_{sec}, the requester learns the actual location of the shared data and sends the request to the healthcare provider for data access.
- *Step 7–8.* On receiving the data access request, the healthcare provider checks the session state in the directory service. If the session exists, it then verifies the request, including the message signature and the range of the requested. If they are all valid, the healthcare provider returns the data to the requester. In data transmission, a secure channel can be established through asymmetric encryption on the data with PK_{req}.
- *Step 9.* The requester, after receiving and decrypting the data with SK_{req}, verifies the data integrity by downloading the data digests from the blockchain service. If the data is valid, then he/she can

access the data. If the data is invalid due to, for example, data loss/corruption during storage or transmission, an alert will be triggered, which is out of the scope of this paper.

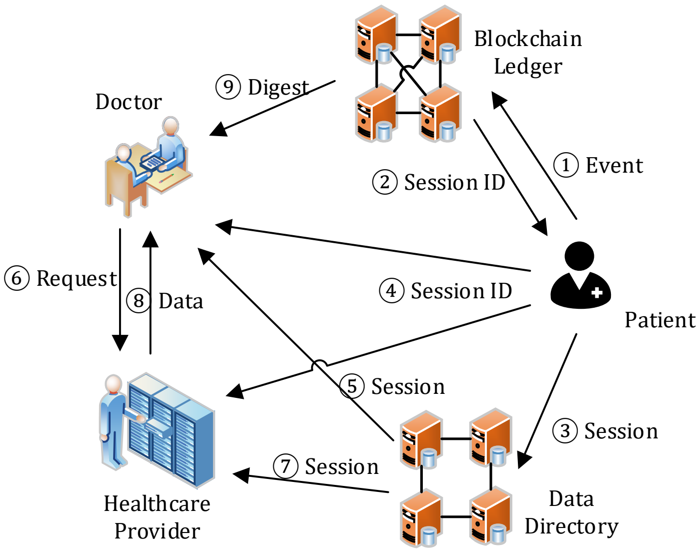


Figure 6. Process of session-based data sharing.

A patient can revoke the access of a requester by closing the session. Session removal is implemented by removing the session in the directory service to prevent the requester from future access and to recycle storage space.

For each data stream in a session, a patient can determine the data access range by specifying the start time (*st*) and end time (*et*) of the shared segment. In a patient-centric sharing scheme, a healthcare provider will only return the chunks within the range to a requester. Let st_c and et_c be the data collection start time and time of a chunk *c*. The chunks allowed to access are $\{c \mid st_c \geq st \wedge et_c \leq et\}$. In a user interface design, we plan to align the pace of time selection with the pace of chunk file generation to avoid time specifying in the middle of a chunk. For instance, if the chunks of a stream are generated per hour, then *st* and *et* will be varied by hour and from the start time of the first chunk.

Note that the above process is flexible at several places. First, a patient can leave the session open for future access by the same doctor, if subsequent consultations are needed or the data is used for medical research. Then, the end time of a data stream in a session can be left empty to enable the doctor to track the patient's condition as long as needed. If the end time is empty, the sharing scheme will not restrict a requester from accessing any chunk of the stream from the specified start time. If both the start time and the end time are empty, no chunk access restriction will be imposed to the requester. Moreover, MedChain does not aim to replace the existing EHR systems at hospitals, but a complementary solution specific for inter-organizational data sharing. Thus, for the doctors who need the local patient records generated at the hospital, they can still retrieve it through existing procedure and the legacy system.

4.3. Key Management

The application provides each user (a patient, requester, or healthcare provider) a software key case for storing the received cryptographic keys. As shown in Figure 7a, a user first generates and stores his/her key pair (public key and private key) in the key case. The public key is signed and verified by the Certificate Authority (CA). Users also store the keys of others for data sharing. Specifically, when a patient registers at a healthcare service, the patient and the healthcare provider exchange their respective public keys for inventory creation. When a requester needs access to a patient's data, his/her public key will be included in the request and cached in the key case of the patient for a session creation. When a session is revoked, the patient will remove the corresponding key (i.e., PK_{req}) from the key case. The first time of use, a user will send a copy of his/her public key to the communicating super-peer for message authentication. Moreover, the keys of patients can be shared to family members for emergent healthcare provision. Yet, a more sophisticated solution for emergency access is left to our future work.

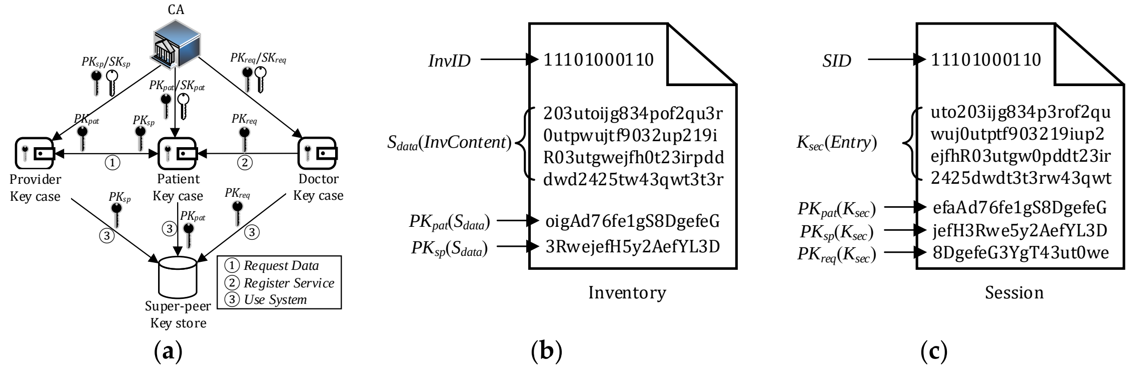


Figure 7. Key Management Scheme: Keys are stored in (**a**) key case, (**b**) inventory, and (**c**) session through encryption.

The symmetric keys (i.e., S_{data} and K_{sec}) are not stored in user key cases, but stored along with the information they encrypted (an inventory or a session section). The symmetric key storage is illustrated in Figure 7b,c. S_{data} is stored within the patient inventory and encrypted by PK_{sp} and PK_{pat}. K_{sec} is stored within each session section and encrypted by PK_{pat}, PK_{sp}, and PK_{req}. Since key cases do not need to store these keys, which are temporarily generated in data sharing (esp. for session keys since inventories are likely to be long-term maintained), this design can largely reduce key management overhead. When the data-sharing network connects to a huge number of healthcare providers and requesters, such a benefit will become more important for the reduced space requirement, and more user-friendly especially for the application clients built on mobile devices.

5. Evaluations

In this section, the accuracy and performance of MedChain are evaluated. First, the system security is analyzed by studying the related properties. Then, the system efficiency is studied through theoretical analysis and experiments. Lastly, the evaluation is summarized through a comprehensive discussion.

5.1. Security Analysis

This section validates the security requirement fulfillment, including the analysis on attack resistance, privacy protection, and non-repudiation. Assume the users of the system never expose their private keys and secret keys to others. The encryption algorithm cannot be compromised and an adversary cannot recover the keys. In addition, the authorized requesters are assumed to be trustworthy. For example, related laws and regulations are enacted to restrain healthcare data secondary usage. To facilitate description, some conventions are made first. A message m sent from C to B is denoted by $C \rightarrow B : \langle m, E_{SK_C}(m) \rangle$ where $E_{SK_C}(m)$ represents the signature of the message signed with C's private key. $C(A)$ represents the scenario that C is disguised as A. Let PK_{adv} and SK_{adv} be the public-private key pair of an adversary.

Resilient to Masquerade Attack: An adversary may pretend to be a user to gain unauthorized access to a data inventory, a session, or a ledger. For example, an adversary C pretends to be patient A and sends PK_A to healthcare provider B for inventory creation for patient A: $C(A) \rightarrow B$: ($m = PK_A$, $E_{SK_C}(m)$). This attack can be defeated by checking the message signature. In the above example, B can discover that $D_{PK_A}\left(E_{SK_{adv}}(m)\right) \neq m$ and ignore the inventory creation request.

Resilient to Replay Attack: An adversary may intercept the message sent from one party to another party and replay the message to access the information without authorization. There could be three cases of a replay attack to MedChain.

Case 1. Assume an adversary C can intercept a message from a healthcare provider A to a patient B for accessing the inventory and the data on a directory service node D.

$$\begin{aligned} A \rightarrow C(B) &: \langle m = \langle InvID, E_{PK_A}(S_{data}) \rangle, E_{SK_A}(m) \rangle \\ C(B) \rightarrow D &: \langle m = \langle InvID, E_{PK_A}(S_{data}) \rangle, E_{SK_A}(m) \rangle \\ D \rightarrow C(B) &: \langle m' = \langle InvID \rangle, E_{SK_D}(m') \rangle \end{aligned}$$

With the *InvID*, C can ery the information from the directory service. However, without SK_A, C can neither learn the identity of B nor decrypts $E_{PK_A}(S_{data})$ for accessing the inventory of B under the *InvID*.

Case 2. Assume an adversary C can successfully intercept an inventory update message from a healthcare provider A to a directory node B for tampering the inventory.

$$\begin{aligned} A \rightarrow C(B) &: \langle m = \langle InvID, E_{S_{data}}(InvContent) \rangle, E_{SK_A}(m) \rangle \\ C(B) \rightarrow B &: \langle m' = \langle InvID, E_{PK_A}(InvContent') \rangle, E_{SK_A}(m) \rangle \end{aligned}$$

Since *InvContent* $\neq$ *InvContent′*, $D_{PK_A}(E_{SK_A}(m)) \neq m'$. Message authentication will fail on B and B will not approve the change on the inventory. The similar inference can be applied to the replay attack on session change and session removal.

Case 3. Assume an adversary C can successfully intercept a session request message from a requester A to a directory node B for accessing the session.

$$\begin{aligned} & A \rightarrow C(B) : \langle m = SID, E_{SK_A}(m) \rangle \\ & C(B) \rightarrow B : \langle m = SID, E_{SK_A}(m) \rangle \\ & B \rightarrow C(A) : \langle m', E_{SK_A}(m'),\ where\ m' \\ & \qquad\qquad = \left\{ \left\langle E_{K_{sec}}(Entry), E_{PK_{sp}}(K_{sec}), E_{PK_{req}}(K_{sec}), E_{PK_{pat}}(K_{sec}) \right\rangle \right\} \end{aligned}$$

Without K_{sec}, SK_{sp}, SK_{req}, and SK_{pat}, C can only get a piece of encrypted information without the decryption method. Thus, C cannot access the content of the session. The similar inference can be applied to the same replay attack between a user or a healthcare provider and a directory server.

Forward Secrecy: An adversary may have compromised the secret key of one session section K_{sec} and tries to use it to access other sections and sessions. However, different sections share different K_{sec}. Thus, the adversary cannot use the same section key to access other sections and sessions. The similar inference can be applied to the forward secrecy of S_{data}.

Data Integrity: The data in integrity analysis can be classified into on-chain data and off-chain data. The on-chain data integrity is ensured by the immutability of the blockchain [8]. The off-chain data can be further divided into healthcare data and directory information. The integrity of healthcare data is verified with the data digest stored on the blockchain. Since the data storage on blockchain is tamper-proof, users can trust the healthcare data if they pass the integrity check.

On the other hand, malicious servers could tamper the directory information from an internal attack. Yet, it will not be a disaster for two reasons. First, all sensitive information is encrypted before storage. Thus, they will not be leaked out even if the directory information is tampered with. Most likely, it will result in data not found (due to ID corruption) or decryption failure (due to content corruption), and users will then be informed of it. Second, once data corruption occurs, the healthcare provider (for inventory) or the patient (for session) can easily recover the directory information. The healthcare provider can reconstruct an inventory with the data from the local legacy system, while the patient can reconstruct the corrupted session with the data from his/her inventory.

Directory information reconstruction may bring additional overhead, since Ref/Ref_L could be lost and only be reconstructed by traversing the blockchain for the event search. Without a single point of failure, fortunately, the decentralized network structure can minimize the chance of directory information recovery by storage redundancy [41].

Privacy protection: Privacy protection is achieved from information encryption. In terms of P2P storage, an adversary can retrieve some directory files. However, the adversary cannot access the content of the inventories without S_{data}. An adversary can also intercept some messages between healthcare providers and directory servers, learning the *InvID*, PK_{pat}, and PK_{sp}. Without S_{data}, however, the adversary cannot derive *InvID* with $H(PK_{pat} \parallel S_{data})$, unable to associate a patient with a healthcare provider. Therefore, the location of the actual data is not exposed. Similarly, an adversary cannot learn the content of a section without K_{key}, which can only be retrieved with SK_{sp}, SK_{req}, and SK_{pat}. In terms of blockchain, an adversary cannot access the encrypted content of the events without SK_{pat}, including the patient identity PK_{user}, PK_{req}, and the *DID*s. Therefore, the adversary cannot identify a patient from an $Event_{DG}$ or a requester from an $Event_{SC}$, which is unable to associate a patient with any healthcare record or diagnosis.

Non-Repudiation of Unauthorized Data Access: In one case, an adversary may successfully retrieve and abuse some patient data without authorization. The digest in a block and the immutability property of blockchain can provide the evidence that a session, which authorizes the adversary the data access, does not exist, proving the illegality of the behavior. In another case, an adversary may access the data out of the authorized access range and deny it. The immutability property of blockchain can also provide evidence for the range of data sharing (i.e., *st*, *et*) in an $Event_{SC}$, which can provide the evidence that the data access is out of the range.

5.2. Efficiency Analysis

5.2.1. Theoretical Analysis on Digest Chain

Given a data stream of n chunks. Suppose the event of chunk corruption is an independent and identically distributed random variable with probability p. The cost to download and verify the integrity of a chunk is b. If no chunk is corrupted, then the total verification cost is b. On the other hand, if the first $n-1$ chunks are correct but the last chunk is corrupted, the verification cost is $2b$. Moreover, if the first $n-2$ chunks are correct, but the second last is corrupted. The verification cost is $3b$, no matter whether the last chunk is correct or not. Notably, if the second chunk is corrupted, all the digests need to be downloaded and verified, no matter whether the first chunk is correct.

Through the above induction, the expected cost of verifying a data stream of length n, denoted by $C(n)$, can be calculated.

$$\begin{aligned}
C(n) &= E\left(\sum_{i=1}^{n} P(X=i)\cdot i\cdot b\right) \\
&= (1-p)^n b + (1-p)^{n-1} p\cdot 2b + \cdots + (1-p)^2 (n-1)b + p\cdot nb \\
&= (1-p)^n b + p^2\cdot nb + \sum_{i=1}^{n-1} (1-p)^{n-i} p\cdot (i+1)b \\
&\geq (1-p)^n b + p^2\cdot nb + pb\cdot (1-p)n \\
&\geq (1-p)^n b + p\cdot nb
\end{aligned}$$

Let $C'(n)$ be the cost that n chunks are individually verified. $C'(n) = n \times b$. Then the performance improvement, denoted by ΔC, can be calculated as follows.

$$\begin{aligned}
\Delta C &= C'(n) - C(n) \\
&\leq nb - (1-p)^n b - p\cdot nb \\
&\leq b(1-p)[n - (1-p)^{n-1}]
\end{aligned}$$

It can be easily found that, for a given probability of chunk corruption p, the upper bound of ΔC increases along with the length of stream n. It implies that the digest chain mechanism can substantially improve system performance when it is applied to longer data streams.

5.2.2. Theoretical Analysis on Communication Overhead

The major communication overhead in system usage includes a blockchain service query, directory service query, new block creation, and data encryption/decryption. The estimated costs of different operations are listed in Table 2. Specifically, the Add Data operation (see Def. (1) and (3)) includes the time of inventory query (T_{Qd}), the time of inventory encryption (T_E), the time of digest computation (T_D), and the time of block generation (T_B). If a new chunk of data stream is added, the time of the last chunk query from the blockchain service is also included (T_{Qb}). In the Add Session operation (see Def. (2) and (4)), the overall latency is related to the number of sections (k) in a session. For each section, the cost includes the healthcare data entry retrieval (T_{Qd}) from the directory service as well as the local cost from the section encryption (T_E). For data retrieval (Figure 6), the overhead includes the time of session query (T_{Qd}), decryption for k sections ($k \cdot T_{E'}$), and digest verification for all the retrieved data ($k \times n$ ($T_{Qb} + T_D$), n: the number of entries in a section). T_{Qb} and T_B depend on the network size and the selected protocol. For the directory service, T_{Qb} increases in $O(log_2(m))$ [40] where m is the number of super-peers. For the blockchain service, T_{Qd} minimally involves five communication steps if BFT-SMaRt [39] is applied.

Table 2. Communication overhead comparison.

Operation	MedChain	Existing Solutions
Add Data	$(T_{Qb} +)\ T_{Qd} + T_E + T_D + T_B$	$T_E + T_D + T_B$
Add Session	$k \cdot \cdot \cdot (T_{Qd} + T_E) + T_D + T_B$	N/A
Retrieve Data	$T_{Qd} + k \cdot \cdot \cdot (T_{E'} + n \cdot \cdot \cdot (T_{Qb} + T_D))$	$k \cdot \cdot \cdot n \cdot \cdot \cdot (T_{E'} + T_{Qb} + T_D)$

Table 2 also lists the communication overhead in the existing blockchain-based data sharing schemes [32–34], which only uses asymmetric encryption for data sharing. Due to the lack of session management, they cannot effectively revoke data access. They also do not involve a directory service to manage mutable data, which results in reduced communication overhead. However, they will incur more overhead in description modification, which will be discussed in Section 5.3.

5.2.3. Performance Analysis through Experiments

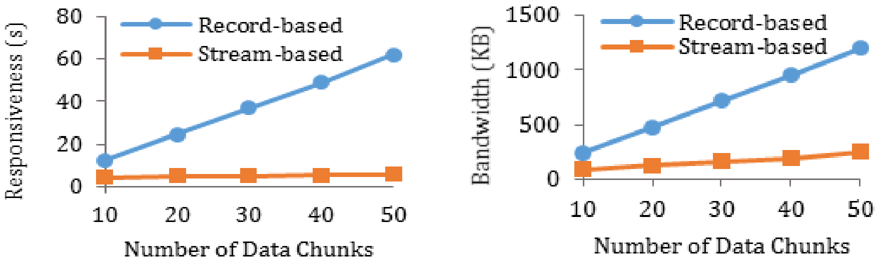
For demonstration, a prototype MedChain [42] system is created, which includes a blockchain service module, a directory service module, and a test client. Moreover, WANem [43] is employed to emulate the latency of a wide area network (WAN). For a privacy purpose, no real-life patient medical data is employed in experiments. Instead, the synthetic data is applied (together with the MedChain prototype), which follows the same ECG recording format as in PhysioNet [44,45]. The parameters of the experiment environment are listed in Table 3 for reference. All involved nodes are simulated on a single machine and connected through the emulated WAN.

Table 3. Parameters of the experiment environment.

Parameter	Value
Machine Specs	Dell server (CPU: 2.93 GHz quad-core Intel Core i7 870, Cache: 8 M, RAM: 16 G of 2400 MHz DDR3, OS: Windows 7)
Network Latency	50 ms + Jitter (Normal Distribution)
Directory Node Concurrency	10 tasks
Blockchain Node Concurrency	30 tasks
Maximal Block Size	400 events

In the experiments, the performance of MedChain is benchmarked against existing models [32–34] to empirically demonstrate the overhead incurred in different data sharing processes. We summarized their characteristics as the variables in the evaluation. Specifically, existing models only support electronic healthcare records, while MedChain also supports data streams. Thus, in the efficiency comparison related to data types, existing models are labeled by "Record-based" and MedChain is labelled by "Stream-based". Moreover, existing models store all data descriptions on blockchain, while Medchain separates mutable information storage to directory. In the efficiency comparison related to storage location, existing models are labeled by "On blockchain" and MedChain is labelled by "On directory."

The first experiment studies the communication overhead in data access. Compared with existing blockchain-based solutions, MedChain encodes the digest of data stream into a digest chain. This approach can facilitate the integrity check in data access, since only the last chunk of a stream needs to be extracted from the blockchain in most cases. In contrast, existing schemes have to download all the digests from the blockchain for validating the integrity of each chunk, which brings more communication overhead. The comparison result is shown in Figures 8 and 9. If only the data stream is shared (Figure 8), the existing schemes (marked as record-based) shows more communication overhead than MedChain (marked as stream-based) in responsiveness and bandwidth consumption. Especially, such an advantage becomes evident when a longer data stream needs to be shared.



(**a**) (**b**)

Figure 8. Communication overhead in sharing data stream through record-based and stream-based digest generation, measured in (**a**) responsiveness and (**b**) bandwidth.

In another scenario, both records and data stream are shared in one session. Without loss of generality, they have the same numbers. As shown in Figure 9a,b, MedChain still performs better over existing schemes, especially with the increase of data quantity. Yet, such an advantage is reduced, because, apart from the data stream, each record file is independent so that sharing different records requires downloading all the digests for integrity check. Figure 9c,d further testifies this effect. However, it also shows the power of MedChain in sharing more data chunks.

Second, the storage overhead is studied. Existing schemes store everything into blockchain, including the description of data, the description of sessions, and the actual data locations. On the other hand, the blockchain in MedChain only contains the fingerprints for the purpose of non-repudiation and integrity check. Other information is stored on the directory servers, which are mutable. The mutable part of session descriptions are only needed by the end of data sharing such as the URLs and the references (i.e., *BH* and *EH*) to the blocks and events in the blockchain. These data can be removed from the directory servers when the session is revoked, while blockchain does not support the removal operation for tamper-proof purpose. Figure 10 shows the comparison of MedChain with existing schemes in storage overhead. The experiment is run in two settings, different number of data (Figure 10a, 10 sessions), and different number of sessions (Figure 10b, 10 medical records). In the first setting, each run includes the procedure of data generation, session creation, and session removal, which can simulate the scenario of sharing different data. In the second setting, data generation events (i.e., $Event_{DG}$) are only created in the first run, simulating the scenario of sharing the same set of data to different requesters. In both settings, MedChain has less storage overhead than existing schemes because the mutability part of a session is removed when the sharing is over. Even if the session removal (SR) is not reported to the blockchain, existing approaches still consume more storage spaces.

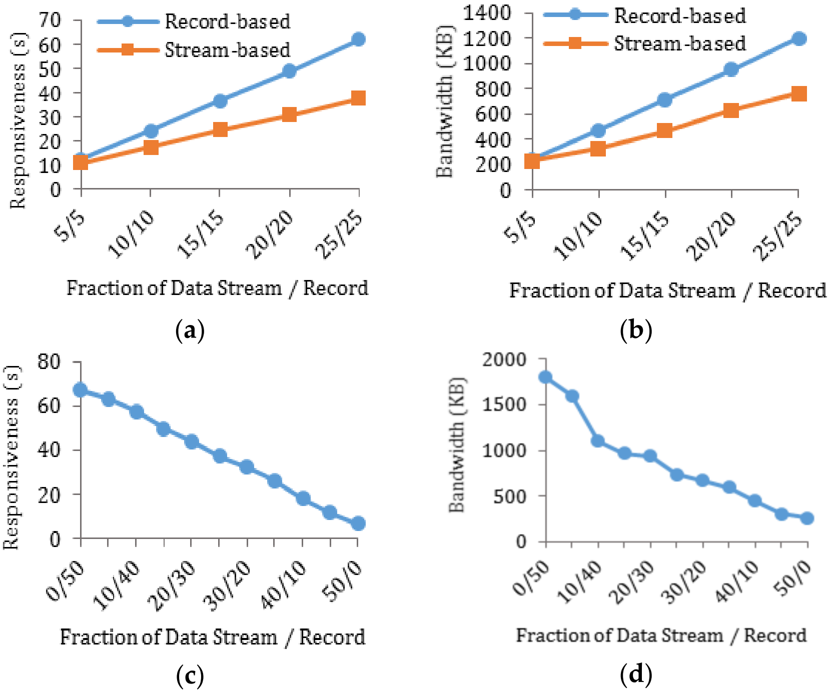


Figure 9. Communication overhead in sharing both data stream and records through record-based and stream-based digest generation, measured in (**a**) responsiveness and (**b**) bandwidth. Communication overhead of MedChain in different fraction of data stream to record, measured in (**c**) responsiveness and (**d**) bandwidth.

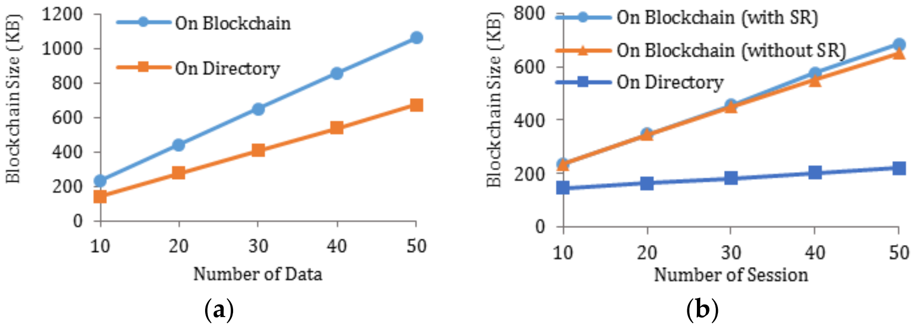


Figure 10. Study the communication overhead of MedChain by comparing the mutable information storage on blockchain with the storage on the directory in (**a**) different numbers of data generated per session and (**b**) different numbers of sessions with 10 medical records to share per session.

In the last experiment, the scalability of MedChain is studied. The system latency is mainly generated by the directory service, since the number of communication steps in a BFT-based consensus is fixed in a normal case [39]. Thus, only the number of directory nodes is increased to test the system scalability. Figure 11 shows the test result. It can be found that the trend of the system responsiveness increase along with the number of directory nodes, which is sub-linear. The result shows the high scalability of MedChain due to the underlying DHT network. The variation (indicated by the error bar in Figure 11) also grows with the number of the directory nodes because the indeterminism of the hop number in a P2P routing also increases.

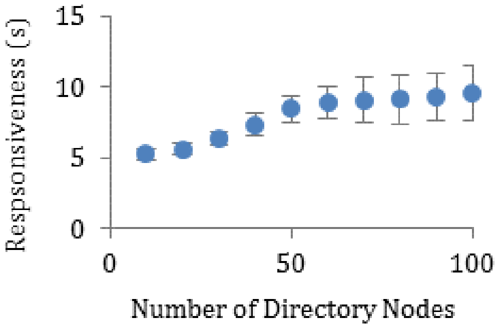


Figure 11. System responsiveness with the number of directory nodes.

The scalability in terms of the transaction rate and the number of nodes is also tested. Different types of nodes are separately scaled since blockchain servers and directory servers are connected to different network topologies and they scale for different purposes. For the blockchain service, as it is implemented by the BFT-SMaRt [39] protocol, the number of blockchain nodes (N_1) is determined by the extent of fault-tolerance: $N_1 = 3f + 1$ where f represents the maximal number of faulty nodes. For the directory service, the number of directory nodes (N_2) scales to increase storage space. Thus, in the experiment, $N_1 = 4$ ($f = 1$), 7 ($f = 2$), and 10 ($f = 3$), while $N_2 = 10, 20, \ldots, 100$, which is consistent with the last experiment. Figure 12 shows the experimental result. The transaction rate decreases with the increase of both blockchain nodes and directory nodes because a transaction involves the consensus process run by the blockchain nodes followed by the routing process run by the directory nodes. Yet, the main factor is the number of directory nodes, as shown in Figure 12. The trend line shows the sub-linear decrease of the transaction rate with directory nodes, which is consistent with the result of the last experiment. Thus, in view of both the responsiveness and transaction rate, MechChain is scalable.

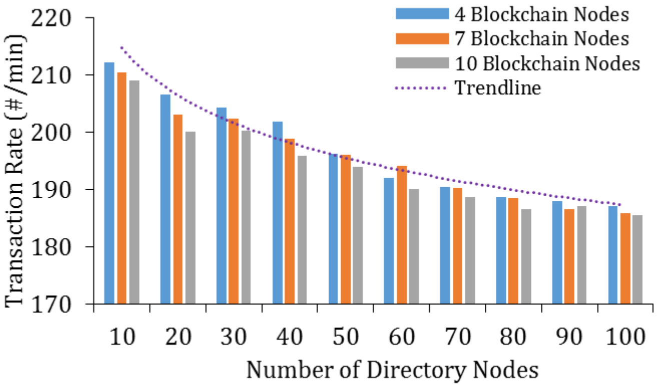


Figure 12. Transaction rates with a different number of blockchain nodes and directory nodes.

5.3. Discussion

Without compromising security, privacy, and scalability, MedChain shows higher efficiency than the existing designs in the previously discussed communication overhead and storage overhead. Furthermore, another advantage is flexibility. By separating the mutable information of healthcare data from the immutable information, MedChain allows healthcare providers to readily update the description of data with minimal overhead. For example, a hospital may change its domain name, which results in the change of all the data locations. If they are stored on blockchain, either a description update is not allowed, or new events and blocks have to be generated for new descriptions, increasing both computation and storage overhead. Such an advantage is intuitive. Therefore, it is not shown in the experiment.

The scalability from the patient point of view is a common issue of all patient-centric solutions, since patients have to respond to each data-sharing request and add each requested data into a session for access authorization. This is a tradeoff between patient controllability and overhead. By controlling session creation and revocation, MedChain is no worse than other solutions. If a patient wants to reduce the overhead from the same requester, he/she can leave the session open forever. Then the data sharing overhead in MedChain is similar to Reference [32]. A more elegant solution could be constructed by introducing attribute-based data sharing and medical history-based data sharing to further reduce overhead, which is complementary to the session-based scheme. Attribute-based data sharing [46] can allow a patient to share his/her healthcare data to a group of requesters tagged with the same set of attributes, e.g., physician and biomedical laboratory. Medical history-based data sharing [47] can enable a patient to grant access to all his/her medical data related to a specific disease/symptom. We will study them in future work.

In summary, Table 4 compares MedChain with other solutions. The result shows the advantage of the proposed scheme in many aspects.

Table 4. Comparison between MedChain and related solutions.

Metrics	[28]	[30]	[31]	[32]	[33]	[34]	[35]	MedChain
Tamper-proof	Y	Y	Y	Y	Y	Y	Y	Y
Non-Repudiation	Y	Y	Y	Y	Y	Y	Y	Y
Attack Resistance	Y	Y	Y	Y	Y	Y	Y	Y
Access Control	N	N	Y	Y	Y	N	Y	Y
Access Revocation	N	N	Y	N	N	N	Y	Y
Privacy-Preserving	Y	N	N	N	N	Y	Y	Y
Block Search	N	N	N	N	Y	Y	Y	Y
Metadata Update	N	N	N	N	N	N	N	Y
Storage Space Recycling	N	N	N	N	N	N	N	Y
Data Stream Support	N	N	N	N	N	N	N	Y

6. Conclusions and Future Work

Within the last decade, healthcare information exchange (HIE) technology has been widely discussed to facilitate patient diagnosis, R&D collaboration, and wellness improvement. Studies [48] have shown that HIE can benefit service providers in utilization increase, cost reduction, quality improvement, coordination of care, patient experience improvement, and disease surveillance. To provide patients with better control over their healthcare data and reduce information fragmentation, we proposed a patient-driven healthcare data-sharing framework, called MedChain. Compared with the existing blockchain-based approaches, our work has shown its higher efficiency in data sharing without a security compromise, through a dual-network architecture, a session-based data sharing scheme, and a digest chain structure. The result of the work is significant, since efficiency has been shown to be one of the top issues in healthcare blockchain adoption.

Our experience suggests that a gradual integration and improvement of healthcare data sharing systems serves the interest of all parties. Healthcare providers are reluctant to migrate their systems and data to a new platform. Thus, in the preceding project [18], we integrated the new solution for inter-organizational data sharing with the legacy systems rather than totally discard them. MedChain extends this idea by developing a decentralized framework to attain more scalability without trusting a third party. Meanwhile, the scope of healthcare data is expanded by including data streams from various tracking devices, which can further help doctors and medical researchers. Thus, MedChain can maximize all parties' interest.

The current version of MedChain contains some limitations and, thus, leaves the space for upgrade. Some of them have already been mentioned in previous sections. Here, we discuss other possible improvements.

First is the data movement issue. If a patient migrates to a new city, access to his/her data in the previous hospital will always require inter-organizational data sharing, which results in additional overhead in diagnosis. A possible improvement is to implement data transfer between two healthcare organizations. In practice, however, medical records are in custody at healthcare providers. Thus, data movement requires design at both the technical and the regulatory level.

Another limitation of this research is that the access of data requires patient's manually approval, which increases the operation overhead of the patient and the access latency of the requester. The limitation could become a bottleneck in an emergent healthcare provision. Thus, in the future, we plan to work on automatic data sharing such that an authorized device, such as a smart phone, can initiate a data sharing session when it receives the request from a doctor. It could be achieved by adding the related rules and conditions configured by users.

Moreover, currently, MedChain still needs healthcare providers to manually upload the information of data streams to the blockchain service and the directory, which is not efficient. This process can be improved by enabling mobile and fog devices [1] to automatically generate and upload the information to the super-peers along with their data collection work. We will also investigate the feasibility of it in our future work.

Author Contributions: Methodology and conceptualization, B.S. and Y.Y. software, formal analysis, and writing—original draft preparation, B.S. validation and investigation, Y.Y. writing—review and editing, J.G. and Y.Y. supervision, funding acquisition, and project administration, J.G.

Funding: This research was partially funded by the MYRG (Multi-Year Research Grant) of University of Macau, research grant number MYRG2017-00091-FST.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Farahani, B.; Firouzi, F.; Chang, V.; Badaroglu, M.; Constant, N.; Mankodiya, K. Towards fog-driven IoT eHealth: Promises and challenges of IoT in medicine and healthcare. *Future Gener. Comput. Syst.* **2018**, *78*, 659–676. [CrossRef]
2. Hossain, M.; Islam, S.R.; Ali, F.; Kwak, K.S.; Hasan, R. An Internet of Things-based health prescription assistant and its security system design. *Future Gener. Comput. Syst.* **2018**, *82*, 422–439. [CrossRef]
3. Badawi, H.F.; Dong, H.; Saddik, A.E. Mobile cloud-based physical activity advisory system using biofeedback sensors. *Future Gener. Comput. Syst.* **2017**, *66*, 59–70. [CrossRef]
4. MarketsandMarkets Research. IoT Healthcare Market by Component (Medical Device, Systems & Software, Service, Connectivity Technology), Application (Telemedicine, Work Flow Management, Connected Imaging, Medication Management), End User, and Region—Global Forecast to 2022. Available online: https://www.marketsandmarkets.com/Market-Reports/iot-healthcare-market-160082804.html (accessed on 21 December 2018).
5. Sahi, M.A.; Abbas, H.; Saleem, K.; Yang, X.; Derhab, A.; Orgun, M.A.; Yaseen, A. Privacy Preservation in e-Healthcare Environments: State of the Art and Future Directions. *IEEE Access* **2018**, *6*, 464–478. [CrossRef]
6. Chen, Y.Y.; Lu, J.C.; Jan, J.K. A secure EHR system based on hybrid clouds. *J. Med. Syst.* **2012**, *36*, 3375–3384. [CrossRef] [PubMed]
7. Abrar, H.; Hussain, S.J.; Chaudhry, J.; Saleem, K.; Orgun, M.A.; Al-Muhtadi, J.; Valli, C. Risk Analysis of Cloud Sourcing in Healthcare and Public Health Industry. *IEEE Access* **2018**, *6*, 19140–19150. [CrossRef]
8. Gordon, W.J.; Catalini, C. Blockchain Technology for Healthcare: Facilitating the Transition to Patient-Driven Interoperability. *Comput. Struct. Biotechnol. J.* **2018**, *16*, 224–230. [CrossRef]
9. The Economist Intelligence Unit of IBM Institute for Business Value. Healthcare Rallies for Blockchains: Keeping Patients at the Center. Healthcare and Blockchain Executive Report. 2017. Available online: http://www.ibm.biz/blockchainhealth (accessed on 21 December 2018).
10. MGMA. Awareness about Blockchain Technology in the U.S. among Medical Practice Administrators and Executives as of 2017. In Statista—The Statistics Portal. Available online: https://www.statista.com/statistics/828392/knowledge-of-blockchain-technology-and-its-areas-of-impact-in-healthcare/ (accessed on 21 December 2018).

11. Cognizant. Healthcare: Blockchain's CurativePotential for HealthcareEfficiency and Quality. Digital Systems & Technology. 2017. Available online: https://www.cognizant.com/whitepapers/healthcare-blockchains-curative-potential-for-healthcare-efficiency-and-quality-codex2995.pdf (accessed on 21 December 2018).
12. Cognizant. Major External Barriers to Adopt Blockchain in Companies in Asia-Pacific in 2017. In Statista—The Statistics Portal. Available online: https://www-statista-com.unh-proxy01.newhaven.edu/statistics/882562/asia-pacific-external-barrier-to-adopt-blockchain-in-companies/ (accessed on 21 December 2018).
13. Miller, A.R.; Tucker, C. Health information exchange, system size and information silos. *J. Health Econ.* **2014**, *33*, 28–42. [CrossRef]
14. Sinaci, A.A.; Erturkmen, G.B.L. A federated semantic metadata registry framework for enabling interoperability across clinical research and care domains. *J. Biomed. Inform.* **2013**, *46*, 784–794. [CrossRef]
15. Featherstone, I.; Keen, J. Do integrated record systems lead to integrated services? An observational study of a multi-professional system in a diabetes service. *Int. J. Med. Inform.* **2012**, *81*, 45–52. [CrossRef]
16. Rinner, C.; Sauter, S.K.; Endel, G.; Heinze, G.; Thurner, S.; Klimek, P.; Duftschmid, G. Improving the informational continuity of care in diabetes mellitus treatment with a nationwide shared EHR system: Estimates from Austrian claims data. *Int. J. Med. Inform.* **2016**, *92*, 44–53. [CrossRef] [PubMed]
17. Hyppönen, H.; Reponen, J.; Lääveri, T.; Kaipio, J. User experiences with different regional health information exchange systems in Finland. *Int. J. Med. Inform.* **2014**, *83*, 1–18. [CrossRef]
18. Yang, Y.; Li, X.; Qamar, N.; Liu, P.; Ke, W.; Shen, B.; Liu, Z. MedShare: A Novel Hybrid Cloud for Medical Resource Sharing among Autonomous Healthcare Providers. *IEEE Access* **2018**, *6*, 46949–46961. [CrossRef]
19. Yang, Y.; Quan, Z.; Liu, P.; Ouyang, D.; Li, X. MicroShare: Privacy-Preserved Medical Resource Sharing through MicroService Architecture. *Int. J. Biol. Sci.* **2018**, *14*, 907–919. [CrossRef] [PubMed]
20. Walker-Roberts, S.; Hammoudeh, M.; Dehghantanha, A. A Systematic Review of the Availability and Efficacy of Countermeasures to Internal Threats in Healthcare Critical Infrastructure. *IEEE Access* **2018**, *6*, 25167–25177. [CrossRef]
21. Liu, Y.; Zhang, Y.; Ling, J.; Liu, Z. Secure and fine-grained access control on e-healthcare records in mobile cloud computing. *Future Gener. Comput. Syst.* **2018**, *78*, 1020–1026. [CrossRef]
22. Wang, H. Anonymous Data Sharing Scheme in Public Cloud and Its Application in E-health Record. *IEEE Access* **2018**, *6*, 27818–27826. [CrossRef]
23. Mehmood, A.; Natgunanathan, I.; Xiang, Y.; Poston, H.; Zhang, Y. Anonymous Authentication Scheme for Smart Cloud Based Healthcare Applications. *IEEE Access* **2018**, *6*, 33552–33567. [CrossRef]
24. Yao, X.; Lin, Y.; Liu, Q.; Zhang, J. Privacy-preserving search over encrypted personal health record in multi-source cloud. *IEEE Access* **2018**, *6*, 3809–3823. [CrossRef]
25. Zissis, D.; Lekkas, D. Addressing cloud computing security issues. *Future Gener. Comput. Syst.* **2012**, *28*, 583–592. [CrossRef]
26. Yüksel, B.; Küpçü, A.; Özkasap, Ö. Research issues for privacy and security of electronic health services. *Future Gener. Comput. Syst.* **2017**, *68*, 1–13. [CrossRef]
27. Yue, X.; Wang, H.; Jin, D.; Li, M.; Jiang, W. Healthcare data gateways: Found healthcare intelligence on blockchain with novel privacy risk control. *J. Med. Syst.* **2016**, *40*, 218–225. [CrossRef] [PubMed]
28. Al Omar, A.; Rahman, M.S.; Basu, A.; Kiyomoto, S. *MediBchain: A blockchain based privacy preserving platform for healthcare data, Security, Privacy, and Anonymity in Computation, Communication, and Storage, Guangzhou, China, 12–15 December, 2017*; Wang, G., Atiquzzaman, M., Yan, Z., Choo, K.K., Eds.; Springer: Cham, Switzerland, 2017.
29. Guo, R.; Shi, H.; Zhao, Q.; Zheng, D. Secure attribute-based signature scheme with multiple authorities for Blockchain in electronic health records systems. *IEEE Access* **2018**, *6*, 11676–11686. [CrossRef]
30. Li, H.; Zhu, L.; Shen, M.; Gao, F.; Tao, X.; Liu, S. Blockchain-Based Data Preservation System for Medical Data. *J. Med. Syst.* **2018**, *42*, 1–13. [CrossRef] [PubMed]
31. Xia, Q.; Sifah, E.B.; Asamoah, K.O.; Gao, J.; Du, X.; Guizani, M. MeDShare: Trust-less medical data sharing among cloud service providers via blockchain. *IEEE Access* **2017**, *5*, 14757–14767. [CrossRef]
32. Patel, V. A framework for secure and decentralized sharing of medical imaging data via blockchain consensus. *Health Inform. J.* **2018**, 1–14. [CrossRef] [PubMed]

33. Azaria, A.; Ekblaw, A.; Vieira, T.; Lippman, A. MedRec: Using blockchain for medical data access and permission management. In Proceedings of the 2016 2nd International Conference on Open and Big Data (OBD), Vienna, Austria, 22–24 August 2016; pp. 25–30. [CrossRef]
34. Fan, K.; Wang, S.; Ren, Y.; Li, H.; Yang, Y. MedBlock: Efficient and Secure Medical Data Sharing Via Blockchain. *J. Med. Syst.* **2018**, *42*, 1–11. [CrossRef]
35. Zhang, A.; Lin, X. Towards Secure and Privacy-Preserving Data Sharing in e-Health Systems via Consortium Blockchain. *J. Med. Syst.* **2018**, *42*, 1–18. [CrossRef]
36. Koblitz, N. Elliptic curve cryptosystems. *Math. Comput.* **1987**, *48*, 203–209. [CrossRef]
37. Merkle, R.C. A digital signature based on a conventional encryption function. In Proceedings of the Conference on the Theory and Application of Cryptographic Techniques, Santa Barbara, CA, USA, 16–20 August 1987; Pomerance, C., Ed.; Springer: Berlin/Heidelberg, Germany, 1987.
38. Gramoli, V. From blockchain consensus back to byzantine consensus. *Future Gener. Comput. Syst.* **2017**, in press. [CrossRef]
39. Sousa, J.; Bessani, A. From Byzantine Consensus to BFT State Machine Replication: A Latency-Optimal Transformation. In Proceedings of the 2012 Ninth European Dependable Computing Conference, Sibiu, Romania, 8–11 May 2012; pp. 37–48. [CrossRef]
40. Stoica, I.; Morris, R.; Karger, D.; Kaashoek, M.F.; Balakrishnan, H. Chord: A scalable peer-to-peer lookup service for internet applications. In Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM 01), San Diego, CA, USA, 27–31 August 2001; ACM: New York, NY, USA, 2001.
41. Lua, E.K.; Crowcroft, J.; Pias, M.; Sharma, R.; Lim, S. A survey and comparison of peer-to-peer overlay network schemes. *IEEE Commun. Surv. Tutor.* **2005**, *7*, 72–93. [CrossRef]
42. MedChain Source Code. Available online: https://github.com/sunniel/MedChain (accessed on 21 December 2018).
43. WANem—The Wide Area Network Emulator. Available online: http://wanem.sourceforge.net (accessed on 21 December 2018).
44. Moody, G.B.; Mark, R.G. The impact of the MIT-BIH Arrhythmia Database. *IEEE Eng. Med. Biol. Mag.* **2001**, *20*, 45–50. [CrossRef] [PubMed]
45. Goldberger, A.L.; Amaral, L.A.N.; Glass, L.; Hausdorff, J.M.; Ivanov, P.C.; Mark, R.G.; Mietus, J.E.; Moody, G.B.; Peng, C.K.; Stanley, H.E. PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. *Circulation* **2000**, *101*, e215–e220. [CrossRef] [PubMed]
46. Baden, R.; Bender, A.; Spring, N.; Bhattacharjee, B.; Starin, D. Persona: An online social network with user-defined privacy. In Proceedings of the ACM SIGCOMM 2009 Conference on Data Communication (SIGCOMM 09), Barcelona, Spain, 17–21 August 2009; ACM: New York, NY, USA, 2009.
47. Ananth, C.; Karthikeyan, M.; Mohananthini, N. A secured healthcare system using private blockchain technology. *J. Eng. Technol.* **2018**, *6*, 42–54.
48. Rahurkar, S.; Vest, J.R.; Menachemi, N. Despite the spread of health information exchange, there is little evidence of its impact on cost, use, and quality of care. *Health Aff.* **2015**, *34*, 477–483. [CrossRef] [PubMed]