Article

Secure Delivery Scheme of Common Data Model for Decentralized Cloud Platforms

Jae Hyuk Cho ^{1,*} , Yunhee Kang ² and Young B. Park ³¹ Department of Electronic Eng., Soongsil University, Seoul 06978, Korea² Division of ICT, Baekseok University, Cheonan 31065, Korea; yhkang@bu.ac.kr³ Department of Software Science, Dankook University, Yongin 16891, Korea; ybpark@dankook.ac.kr

* Correspondence: chojh@ssu.ac.kr

Received: 7 September 2020; Accepted: 12 October 2020; Published: 13 October 2020

Featured Application: Application: The proposed Secure-Cloud Common Data Model (SC-CDM) system is designed as a blockchain-based platform using a distributed ledger. It provides reliable confidentiality, security and expandability to enhance the usability of CDM, a data format that can facilitate data analysis.

Abstract: The Common Data Model (CDM) is being used to deal with problems caused by the various electronic medical record structures in the distributed hospital information system. The concept of CDM is emerging as a collaborative method of exchanging data from each hospital in the same format and conducting various clinical studies based on shared data. The baseline of a CDM system is centralized with an infrastructure typically controlled by a single entity with full authority. The characteristics of this centralized system can pose serious security issues. Therefore, the proposed SC-CDM system is designed as a platform for distributed ledger and provides data with a high level of confidentiality, security, and scalability. This framework provides a reference model that supports multiple channels, using secure CDM as an encryption method. The data confidentiality of CDM is guaranteed by asymmetric and symmetric protocols. Delivering CDM is protected by a symmetric key signed by the CDM creator and maintains lightweight distributed ledger transactions on Inter Planetary File System (IPFS), which acts as a file share. To deliver an encrypted CDM on the SC-CDM platform, the CDM is encrypted with a block cipher by a random symmetric key and Initialization Vector (IV). The symmetric key protocol is used for the fast encryption of large-capacity data. The SC-CDM is implemented the repository with IPFS for storing the encrypted CDM, in which symmetric key, two hash values, and IV are shared through blockchain. Data confidentiality of SC-CDM is guaranteed by only registered users accessing the data. In conclusion, the SC-CDM is the first approach to demultiplexing with the data confidentiality proof based on asymmetric key cryptography. We analyze and verify the security of SC-CDM by comparing qualitative factors and performance with existing CDM. Moreover, we adopt a byte-level processing method with encryption to ensure efficiency while handling a large CDM.

Keywords: common data model; distributed ledger; SC-CDM; IPFS; symmetric and asymmetric protocol; blockchain

1. Introduction

In recent years, medical services have shifted from treatment to prevention and management of diseases. The importance of using medical information is on the rise for efficient patient health management [1]. A Hospital Information System (HIS) manages Electronic Medical Record (EMR) that is a digital version of a patient chart containing patient's medical history. Each hospital has a different

structure of EMR in HIS; hence, each hospital can only conduct research based on the patient data of its own HIS form. However, it is very difficult to draw the ultimate conclusion because the patient groups in each hospital tend to show different characteristics. Data sharing is the fundamental step to gain maximum benefit from research innovations. Clinical studies based on patient data from numerous hospitals can provide more meaningful results. To handle the problem caused by diverse structure of those HIS, needs of integrated Common Data Model (CDM) have been recognized [2].

Distributed storage can be a solution to the problem of centralized storage that utilizes a distributed ledger and allows data to be stored independently on multiple nodes in the network. The issue of distributed storage is the limitation of storage capacity and processing of network nodes. To overcome these limitations, this paper combines distributed storage and Inter Planetary File System (IPFS), which is a peer-to-peer architecture [3].

The Secure-Cloud CDM (SC-CDM), designed by us, generates the encrypted CDM with a random key and initial vector and signs it with the public key of the CDM consumer to securely deliver the data before they are uploaded to IPFS. The encrypted CDM is lightweight via IPFS, but is maintained in transactions on a distributed ledger. The proposed system employs an immutable ledger on a message channel basis. It can manipulate and modify the current state of CDM. Finally, we evaluate the fragmented CDM algorithm in terms of elapsed time depending on the number and size of fragments while writing and reading the CDM.

The main contributions of our work can be summarized as follows:

1. The SC-CDM is the first approach of demultiplexing with the data confidentiality proof of CDM based on asymmetric key cryptography. We analyze and prove the security of SC-CDM.
2. Compared to previous solutions, we adopt a byte-level challenge with encryption to ensure efficiency while handling a large CDM.
3. The comparison results show that our scheme has the higher performance compared to other schemes.
4. By using X.509 asymmetric key for DID (Decentralized Identifier)-based identity management (signing and using the DID operating, autonomous generation, updating, and removal functions), the risk of personal information exposure would be reduced.

2. Materials and Methods

2.1. Materials

2.1.1. Distributed Ledger

A distributed ledger is a new trend in the field of information technology. A lot of financial and non-financial applications has made the worth of distributed ledger. Blockchain is the latest leading technology. It brings a solution for challenges like traceability, transparency, trust, and accountability [4,5]. Distributed ledgers are multi-purpose technology in the digital world that are specifically designed to be shared across network of multiple sites, geographies, or institutions.

A blockchain is a linear form of a distributed ledger composed of immutable blocks of data, with each block containing a list of transactions and a unique reference to its predecessor block [6]. It is commonly considered as a specialized form of a distributed database [7]. Blockchain offers the ability to create unique (sometimes scarce) digital assets. Blockchain can solve the problem of Single Point Failure (SPF) [8]. Bitcoin's presenter proposed a mechanism to ensure digital data rights management [9].

Blockchain has features that can be utilized for transparent distribution of CDM data. The CDM provider functions as a peer and performs sequencing and validity determination of transactions on the blockchain. It also guarantees high availability through a peer-to-peer (P2P)-based file sharing system.

2.1.2. Decentralized Identifier (DID)

Identifiers uniquely identify an item of datum and give the datum a unique name to express its characteristics. DID, which is unique worldwide, is a new type of identifier that can be operated with high availability and can be cryptographically verified [4]. DIDs also support the principles of self-sovereignty [10], that is, they allow the identity owner to create, manage, and discard identifiers that appears to be suitable. It supports the use of multiple simultaneous identifiers, thus meeting the privacy enhancing properties. CDM can increase the interoperability of medical data between hospitals, but it is required to provide a secure medical data access environment for medical data analysis by various researchers, including disease prevention.

Identity management for the CDM operation process of hospital medical and prescription data is classified into four types: isolated, centralized, federated, and distributed ID. The isolated type is a form in which the user's identity is managed for each service, and the user needs the identity registration (member registration) and verification (authentication) procedures for each service, and the identity management system is securely built and operated for user authentication and access control of the cloud CDM service. In order to do that, it is expensive. In terms of security system, cloud CDM users need to register for identity verification and store or manage identity verification information (ID/PW, etc.), separate from hospital ID.

The centralized type is a method of centrally managing the identities of trusted users, and the construction and operation efficiency of the identity management system is higher than that of the isolated type. Users can register their identities only in the centralized management system and use the SSO (Single Sign On) function to connect multiple services through one authentication. With this technical feature, the centralized type is suitable for providing multiple services through a single cloud CDM. However, when a failure occurs in the central management system, the entire service cannot be used, and the interoperability and scalability are limited due to the closed structure.

The federated type is a method in which different service providers form a trust relationship for the convenience of users, and jointly manage the user's identity [11]. The federated type is suitable for SSO support between the hospital and the cloud CDM network. However, federated ID management should be preceded by establishing a trust relationship between the hospital and the cloud CDM, and if identity management is concentrated on a specific service provider, it has the same limitations as the centralized type.

The distributed ID type is operated in a way that users create their own identification information and share it through DLT (Distributed Ledger Technology) [12], and use DLT to share information to identify each service provider. The convenience of identity management can be improved by eliminating the need to register verification information. In terms of security system, since the user's identity is not stored in the central server, DB, etc., there is little risk of ID leakage, and theft and has the advantage of ensuring the integrity of the identification information through DLT technology.

2.1.3. Blockchain

Blockchain technology is a technology that transparently records transaction history on a ledger that anyone can read and copies and stores it on multiple computers. Therefore, since this technique is based on individual transactions, there is no need for a central administrator. In addition, because multiple computers verify the records, it is difficult to forge or alter them by third parties. The most representative applications based on blockchain technology are smart contracts, virtual currencies such as bitcoin, etc. Among them, decentralization, a characteristic of bitcoin, facilitates the sharing of control between users and ID providers, allowing users to manage their identities [13].

The NEXTLEAP system was developed for the purpose of alternative decentralization and enhanced privacy protection. Built as a federated ID system, this system uses algebraic MAC-based blind signatures to enhance privacy protection. It was also verified by the stability of messages through distributed IDs for messages allowed between different servers [14]. The blockchain protocol

“Ouroboros Praos” was developed to demonstrate the stability of security by delaying transmission in a semi-synchronous setting for complete conversion [15].

The chainAchor system is introduced to utilize identities that can be provided for privacy protection purposes. This system can read/verify transactions on the authorized blockchain, create a read-only list of public keys of anonymous members, and perform simple lookups on entities on the blockchain [16]. It also provides a protocol for an identity management system built into Bitcoin. This facilitates the sharing of control between users and ID providers, and verifies the relationship between the rights and responsibilities of users and ID issuers [13].

In addition, various studies are being conducted to improve security issues and operate more efficient systems through blockchain. A new practical paradigm named Thunderella was proposed through the replication of the system state by combining the asynchronous path and the synchronous alternative path [17]. It also provides the first official analysis of Bitcoin’s target (re)calculation capabilities in crypto settings. It extends the q-bounded motive model of the Bitcoin backbone protocol, which presented the properties of Bitcoin’s basic blockchain data structure, and shows how to build a strong public transaction ledger on top of it [18]. In order to reinforce the security issues of the developed system, abstract blockchain protocols can be defined and appropriate security properties of these protocols can be identified. To this end, it meets Nakamoto’s blockchain protocol and proves that these properties are sufficient for general applications [19]. For proof of security, Bitcoin has presented a ledger of (G) UC models such as Canetti, and a process has been proposed with the aim of achieving function [20]. A blockchain-based secure data sharing mechanism for VN (Vehicular Networks) is proposed. In order to efficiently manage service provisioning, we have introduced an edge service provider with regular nodes [21]. The dual decentralized network blockchain utilizes a licensed blockchain for increased processing speed and security, and allows the system to be used openly and without permission to application developers while dividing the system into multiple layers [22].

The public blockchain is based on a distributed ledger and is classified as a public ledger limited to P2P-based financial transactions in terms of opening and managing the transaction ledger [23]. A private blockchain to solve these limitations can be used to share CDM data by authorized users such as the medical sector. In addition, this technology method can provide functions as it becomes easier and safer to participate in data services by using digital identifiers [24].

2.1.4. Common Data Model (CDM)

In medical and pharmacy research, the management of sensitive medical information such as patient information and safety issues are emerging domestically and abroad, and the medical community is paying attention to the possibility of utilizing a highly objective and safe research through the use of a CDM. Therefore, the introduction of CDM is increasing in domestic and overseas hospitals. The analysis of a lot of real data utilizes the CDM system. For example, it facilitates analysis across data sources by standardizing terms for drug use, medical events and procedures, data structure, and data interpretation [25].

Electronic Health Records (EHR), which are widely used in hospitals, are rapidly ubiquitous in the medical field, but their use in biomedical and clinical research is limited due to interoperability issues and technical requirements. Therefore, software that directly interacts with previously constructed EHR data utilizing the extracted CDM data can make the data more accessible to users with limited computing expertise or domain knowledge, or EHR-based users, and accelerate related research. Moreover, we introduce PatientExploreR, a scalable application built on the R/Shiny framework that interfaces with a relational database of EHR data in Observational Medical Outcomes Partnership CDM (OMOP-CDM) format, which generates patient-level interactive dynamic reports and provides clinical data without programming [26].

Based on the EHR system, CDM has been expanded and built to enable the analysis and comparison of Adverse Drug Reactions (ADRs), whose data structures are integrated with other

external organizations. It provides a basis for collaborative research, analysis, and comparison between institutions in which the same type of CDM has been built, and provides an environment in which the same research can be conducted simultaneously on a variety of data sources [27].

The use of standard Health Information Exchange (HIE) data is increasing due to operability. However, the integration of HIE data into the EHR system has not yet been studied. To ascertain the quality of these data, referral documents from the Health Level 7 (HL7) Clinical Document Architecture (CDA) are converted to a CDM, to facilitate HIE data availability for longitudinal data analysis and to identify data quality levels. The CDA document was used to define the mapping rules for CDA-CDM conversion, confirming that CDM conversion was possible [28]. In addition, it has succeeded in transforming EHR into a common data model for Observational Health Data Science and Informatics (OHDSI), and this method can provide various opportunities for researchers and data holders [29]. Data integration is an important task in healthcare information and has a huge impact on what you get from existing health information data. Research using the OMOP-CDM, a common data model used to unify the data structure of each institution, is continuously being conducted in several countries. In France, a feasibility assessment was conducted for the implementation of the national electronic health record, which standardizes data and facilitates data exchange, sharing, and storage, using a CDM. This is especially the case when data are collected through different heterogeneous systems. CDM also provides tools for data-quality assessment, model integration, visualization, and analysis. An extract-transform-load process was implemented to feed data from the French healthcare system to the OMOP-CDM, and, as a result, 17 vocabularies corresponding to the French context were added to the concept of the OMOP-CDM [30].

2.1.5. Blockchain Application in CDM Environment

Medical data is defined as personal sensitive information by law, so its use is limited. In order to use the various healthcare data that have been established, it is limited to de-identification of personal information data, or individual consent is required, resulting in restrictions on data linkage/utilization. In addition, for the safe use of medical data, problem-solving for security threats must be preceded.

Medical data have conflicting action plans to protect and utilize personal data. As the openness of public data is activated, related personal information leakage and re-identification cases are increasing, and data provision is limited due to issues such as information security and personal information protection. Due to the nature of medical data, the scope is only medical information, not non-medical data, and such data are differentiated from the existing traditional database. Rather than implementing high performance, it is necessary to develop a technology and build an environment that requires a high level of security, considering the characteristics of medical information. As a result, it is necessary to propose a more reinforced system that can solve the information security system problems such as medical data leakage and re-identification, and this is presented in this paper.

In this paper, we present a study on blockchain technology for handling authority and management of medical data and a data exchange model for data linkage between institutions. In the case of blockchain, it is possible to protect personal information and strengthen self-determination through secure data exchange or smart contract. As for the data-exchange model, there is a CDM method developed for the purpose of exchange and analysis of distributed and stored data. Research that applied blockchain technology to the CDM environment is in the early stage of research, and specific implementation cases have been rarely published.

2.2. *Methods for Adapting to CDM Model*

2.2.1. The Baseline CDM Model

Data extracted from EMRs tend to be stored in different relational database schemas. Figure 1 illustrates the conventional concept of CDM and its operation scheme derived from several sources of EMR in hospitals.

Figure 1. Conventional concept of Common Data Model (CDM) and operation scheme.

In order to collect and integrate clinical data of multiple hospitals, it is required to solve the heterogeneity of data structure and format, differences in quality and quantity of data, technical limitations of interoperability, and security issues. CDM should support linking of common analysis codes for EMR resource linkage to support integrated data analysis of research institutions, without leaking sensitive personal information. Unitary CDM construction is an essential process for supporting a distributed research network. Observational Health Data Sciences and Informatics (OHDSI) are also pursuing the goal of developing open-source tools based on the OMOP-CDM and establishing a distributed research network [31]. OMOP participates in more than 160 institutions around the world and supports joint research using distributed research networks. OHDSI aims to build a system that can support interoperability in order to utilize open research results [32].

2.2.2. The Secure Operation Scheme for the SC-CDM

The SC-CDM aims to provide a distributed platform for CDM cloud, from transmission to utilization, including the workflow process communications for benefiting health professionals and patients.

The use of CDM format is important for efficient data exchange between organizations in the network, given the diversity of organizations in the SC-CDM. It maximizes the use of existing data resources. The CDM has been constructed and supplied by several providers, thus resulting in separated and disseminated its data by clinical researchers. Figure 2 illustrates the concept of the SC-CDM. As you seen, in Figure 2, the trust manager of hospital A signs a CDM that is access by hospital B. More specifically, the CDM in Node_A is signed by the private key (PrivateKey_A) corresponding to the public key (PublicKey_A) contained in Node_B. Then we represent it as CDM_{A→B}. The signed CDM is verified with PublicKey_A by the trust manager of hospital B operated in Node_B.

The proposed CDM reference model is constructed by several CDM providers and CDM consumers that consist of hospitals and research institutes. Using this reference model, clinical researchers separate and disseminate the CDM data. The following describes the details of the CDM reference model.

- Cryptography can be used for protecting information, using a hash value to maintain management of large-capacity CDMs.
- DID is applied to solve the risk of central ID management caused by SPF and privacy leakage problems.
- Distributed ledger is used to provide data integrity and share information through CDM signature.

Figure 2. Concept of the Secure-Cloud Common Data Model (SC-CDM) for building $CDM_{A \rightarrow B}$.

2.2.3. Applied Security and Privacy Method in SC-CDM

This section presents a method for handling the security and privacy concerns in the SC-CDM, including the model for secure access of CDM data. CDM is represented as a digital asset concerned in a blockchain. The participants of the blockchain are the stakeholders of CDM, which consists of a CDM consumer, a CDM provider, and a service broker. The service broker plays a role as a mediator that preserves a trustworthy delivery of CDM between the CDM consumer and CDM provider. In the CDM blockchain, the transaction can be explained as three operations related with CDM. To access the permissioned CDM blockchain (represented as a node in Figure 3), X.509 is used for defining the format of public key certificates. Node A represents a CDM provider, and Node B represents a CDM consumer. Two trust managers located in the service broker play role as agents of the CDM provider operated in Node A and the CDM consumer operated in Node B for delivering the CDM trustily (represented as $CDM_{A \rightarrow B}$ in Figure 3).

Figure 3. CDM delivery process.

When a certificate is signed by a trusted certificate authority in SC-CDM, the CDM consumer accesses the public key of the CDM provider. The SC-CDM can establish secure communications with

the components and validate $CDM_{A \rightarrow B}$, which is digitally signed by the corresponding private key owned by the CDM provider. Table 1 shows the basic definition of the components of SC-cloud.

Table 1. The basic definition of the components of CDM blockchain.

	Component	CDM Blockchain
1	Digital asset	CDM with generated digest
2	Participant	CDM provider CDM consumer Service broker
3	Transaction	Write the digest of CDM Read the signed CDM Read the digest associated with CDM
4	Access control	Read a public key by CDM provider Read a public key by CDM consumer Write a public key by Service broker
5	ID management	X.509 public/private key

Table 2 summarizes the qualitative comparison of the characteristics of the baseline CDM model and the proposed cloud CDM model, SC-CDM. In terms of the fault tolerance, data sharing, privacy, and data integrity, SC-CDM is an improvement over the baseline CDM model.

Table 2. Comparison of qualitative factors in the CDM model.

	The Baseline CDM Model	The Proposed Secure-Cloud CDM Model (SC-CDM)
Fault tolerance	Limitation of single point of failure	High availability with Replication
Data Sharing	High cost of sharing large CDM data	Low cost of data sharing using P2P repository and distributed ledger
Privacy	The exposure of CDM data ownership	Enhanced secure DID ownership
Data integrity	Vulnerability of CDM Data integrity	Ensuring data integrity with cryptography
DID = Decentralized Identifier; P2P = peer-to-peer.		

2.2.4. CDM Separate Block Encryption/Decryption Algorithm

Both an asymmetric key and a symmetric key are valid on the blockchain. The symmetric key is applied for CDM data encryption according to performance characteristics of encryption and decryption of Cloud CDM service and its application. The symmetric protocol is used to ensure data confidentiality with CDM data resilience. When the multi-channel method is used, a large amount of data transmission between devices is distributed to other channels, thus increasing the efficiency of communication and security and reducing the occurrence of information exposures problems that may occur during the data transmission process. Figure 4 shows how the encrypted CDM between Node_A and Node_B in SC-CDM is shared. In order for the CDM data owner to transmit CDM data, each of the original CDM data are divided into two pieces, and then the divided parts are encrypted, using a symmetric key and a random Initialization Vector (IV).

Figure 4. Sharing the encrypted CDM between nodes in SC-CDM.

The encryption scheme is described by the key-generation algorithm (Gen), having a key-space (K), the encryption (Enc), and decryption (Dec) algorithms, with the data-space (D).

- Gen generates a random key (32 byte) and an IV (16 byte);
- Enc/Dec is based on Advanced Encryption Standard (AES) and Cipher Block Chaining with IV;
- D: CDM derived by EMR.

It is used to ensure data confidentiality and integrity of SC-CDM. Every CDM has different key and IV to increase the security by exploring randomness. A blockchain is used for exchanging information of encryption and decryption of CDM between a CDM producer and a CDM consumer through transaction.

The transaction has four components:

- IV;
- Shared key (generated by symmetric key protocol);
- Two hash values generated by SHA-256.

The IV adds randomness at the start point of the encryption process of chained block encryption mod, and the random IV creates key file for each encrypted file. A cipher block is generated from encrypting a XOR output of the previous cipher block and forwards plaintext block.

In Cipher-Block Chaining (CBC) mode, the previous cipher block is given as input to the next encryption algorithm after XOR with original plaintext block [33]. If each key is only ever used to encrypt a single CDM, one can get away with using a fixed IV. This random IV ensures that each CDM encrypts differently, such that seeing multiple CDM encrypted with the same key does not give the attacker any more information than just seeing a single long message.

3. Results

3.1. Experimental Setup

In this experiment, we built a prototype called SC-Blockchain to evaluate the designed the SC-CDM, which runs on Linux operating system (Cent OS 7.x, IPFS 0.42). We employed a lightweight blockchain/SC-Blockchain, which shows advanced performance on timestamping data with limitation, allowing very small amounts of data to be handled at a given time. Since the hash was added to the blockchain at the time of upload to the IPFS network, the signed document was timestamped in a

tamperproof way. IPFS, a web protocol for data decentralization, uses content-addressing to uniquely identify each file in a global namespace connecting all devices. For IPFS to provide tamper-proof content retrieval, it uses hash points as its contents, which cannot be altered in any way. SC-Blockchain is also used for the management of the distributed ledger, persisting a separate storage structure in IPFS.

To deliver a request for CDM from clients to SC-CDM, an agent instead of the client needs to interact with tasks via a message broker. The SC-CDM participants generate an ID based on X.509 and store its public key in SC-Blockchain. The process of requesting and responding as well as delivering CDM data is done through a separate channel. The CDM processing related with requesting and responding is performed through a trust channel SC-Blockchain. The user's CDM signing and verification is configured to be performed through celery for asynchronous queue processing. Message queues are processed using REDIS, which supports NoSQL to improve performance.

3.2. Construction of Storage Structure for SC-CDM

IPFS is a P2P distributed storage system for storing large-capacity files or data [34]. By storing the data in an IPFS network file system, the IPFS hash for that data is obtained. Figure 5 shows the network bandwidth of IPFS in SC-CDM.

Figure 5. CDM concept and operation scheme.

If a large amount of CDM data is shared Care Delivery Organizations (CDOs) via the IPFS network, the actual CDM data are encrypted and stored separately in IPFS, and the IPFS hash is stored and obtained in the blockchain [35]. Figure 6 shows how IPFS is starting as a daemon.

```

[~/CDM/cBlock]$ ipfs daemon
Initializing daemon...
go-ipfs version: 0.4.23-
Repo version: 7
System version: amd64/darwin
Golang version: go1.13.7
Swarm listening on /ip4/127.0.0.1/tcp/4001
Swarm listening on /ip4/192.168.0.12/tcp/4001
Swarm listening on /ip6:::1/tcp/4001
Swarm listening on /p2p-circuit
Swarm announcing /ip4/1.247.66.125/tcp/4001
Swarm announcing /ip4/127.0.0.1/tcp/4001
Swarm announcing /ip4/192.168.0.12/tcp/4001
Swarm announcing /ip6:::1/tcp/4001
API server listening on /ip4/127.0.0.1/tcp/5001
WebUI: http://127.0.0.1:5001/webui
Gateway (readonly) server listening on /ip4/127.0.0.1/tcp/8080
Daemon is ready
=

```

Figure 6. Starting the Inter Planetary File System (IPFS) as daemon.

3.3. X.509 Based Key Management

In this experiment, the ID of the administrator of the hospital was generated by using the OpenSSL library in python, and the X.509 certificate was generated and maintained in its own wallet. The public key associated with the generated private key was stored in the distributed ledger [36]. The following is part of the public key. The public key is used to verify the CDM delivered and is stored in IPFS.

—BEGIN PUBLIC KEY—

```

MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAsZhhPfQsP3MTrCnn7wsJ
wLX/VWxg1/Qf4Usv/uIfjGta2hcg2hg+TzSqV3bHjFssKBCSp7QnxC1dITt1QbAK
oifDT0dEOoOyYZ7o1xTZBfjUZBa6w7vAnMmOnKPdf/QXgOZjHXr+c4mPjOB5OBVA
wIaiO3ASSACkdeGZ2lZxrZe3tXYeX3Ng8a/nJ8OSdxWuoMKTQ3roooi8I4iC2SXU
5sgb9TxAF/nnm8q4vEiMPBi8d7FqsVPfh4tA2PoESqO/4g4c295Awfm2FT7ut8Z5
n1YyS3PZP7AcGnInWcZ6K5NAsUzsA60+WAgXluUb4lzuCX9ZhhHM+/utu50in0Zd
xQIDAQAB

```

—END PUBLIC KEY—

3.4. Our Encryption and Decryption Scheme for Delivering in SC-CDM

3.4.1. Integrity Auditing with the Digital Signature Based on Asymmetric Key Cryptography

Figure 7 presents an example of how to encrypt and decrypt data in a file named a.out by using the encrypt program implemented the block encryption and decryption algorithm, as well as how to sign and verify it by asymmetric key algorithm. It can describe the state of data and the process of ensuring and preserving the validity and accuracy of data between two parties, the CDM provider and CDM consumer. One (CDM provider) encrypts data by using the public key in the file yrp pubkey.pem, which is distributed from SC-CDM; the decryption key is derived from DID. To audit the integrity of CDM delivered from the outside to the SC-CDM, the other (CDM consumer) verifies the signature (in signfile.txt) of the CDM by yrprikey.pem, or its parts to be used later over encrypted data, and uploads the expected responses to the trust manager of SC-CDM.

```

$ ./a.out
hello world
$ ls a.out
a.out
$ python encrypt.py -c 1 -s 24 -k yrpubkey.pem -i a.out
[Encrypt] size = 12608, elaped time = 0.0003712177276611328
[Add to IPFS]size = 12608, elaped time = 0.14368486404418945
$ rm a.out
$ ls a.out
ls: a.out: No such file or directory
$ python decrypt.py -i sendfile.txt -o a.out -s signfile.txt -k yrprikey.pem
size = 12640, elaped time = 0.004257917404174805
$ ls a.out
a.out
$ chmod +x a.out
$ ./a.out
hello world
$

```

Figure 7. Testing of encryption and decryption of a file in SC-CDM.

Our integrity auditing protocol is an asymmetric key based integrity auditing scheme. When applying our integrity auditing protocol, an asymmetric key derived from DID of a client can audit the integrity of CDM without generating additional metadata.

3.4.2. Secure Proof of Ownership over Secure Transaction

Proof of ownership is similar to the integrity auditing protocol [37], but with the role of prover and verifier changed. In our construction, only clients with intact data can pass the proof-of-ownership protocol. Moreover, since the proof of ownership protocol is performed over encrypted data [38], the protocol does not expose any information, since CDM delivered is persisted in the IPFS as blockchain.

The SC-Cloud handles CDM data in IPFS in order to support only secure temporary storage of the transferred CDM data, thereby making file information of the data confidential. Currently, IPFS does not provide separate access control for external access, so it needs to be resolved.

In order to handle the problem of keeping data of a large volume of CDM in the distributed ledger of the SC-CDM, IPFS is applied to perform effective and secure management of CDM data. The following is a transaction for delivering the CDM. The transaction for CDM is signed with SHA-256. Table 3 shows the contents of the transaction for CDM.

Table 3. The contents of the transaction for CDM.

No.	Item	Contents
1	Sender	fK5q+LBx9DXQm9vxAJPaw1zBWA9UT60V5Ma12uxXGDQ=
2	Recipient	HhhAXOj+iMqRRNJjgDaRD/xK78 × 1248F7WNR9cEjxHk=
3	Key	KMGFK4HR3IFSHYMYUIYZUYZRCZZPR444B6K7D3BNZCGJ7D6C3OXA=====
4	IV	F9A90A83DA5E986ACEE3EE4F7AEA8343
5	Hash1	QmPs2BsjB9VBuV6pHxdVYJP8fsnbLu8BTY93b9anwwW6SM
6	Hash2	QmfHCn3qnnWobcmmWrVZTioWW6UdWxMrUqkBbD1BiTihiv
IV = Initialization Vector.		

3.4.3. Privacy Leakage Resilience of SC-CDM

In constructing SC-CDM, since we are the scheme based on asymmetric key encryption, we prevent leakage of personal information in the integrity audit process. In the secure operation scheme, the public key of CDM consumer is maintained in SC-Blockchain.

Figure 8 shows the result of the transaction log having a note CDM for registering public keys of CDM provider and CDM consumer in SC-Blockchain. CDM Separate Block Encryption/Decryption Algorithm (in Section 2.2.4) is used for building the encrypted CDM. The encrypted CDM is stored in IPFS with its metadata for signing and verifying. The transactions of new blocks can be verified by its hash value locally and the transactions from the IPFS network also can be verified. Transaction should be stored into IPFS. In the public-key-based solution, subsequent clients need to know the public key generated by the CDM uploader to audit the integrity. That is why subsequent clients can immediately learn who has a record. Figure 9 shows the transaction log related with consensus.

```

ctl : lAhNyduhvnuzr1cWcdJIOAVT5qMCIIR5UbuhiQhiCV4=
ctl : 1597649614.304564
ctl : 4vSTWwH_Y0l1emdbX_qEtIwhzbZIm3BL4Wcu3VYPZWl=
ctl : 1597649621.232658
sender: 47hLwlm/ZhJqib6LFurLEANJRLazquQBpSkXrBRhH7A=
recipient: G5PL+9DfGwbW9n1+RDbSKE0BMqW7noZ5G6J50qwNm5c=
pubKey: 47hLwlm/ZhJqib6LFurLEANJRLazquQBpSkXrBRhH7A=
sender: G5PL+9DfGwbW9n1+RDbSKE0BMqW7noZ5G6J50qwNm5c=
recipient: Cv2xoUiseMP70/2aQLTU7F9yvgJ16lRhSJHNmnaLnA=
pubKey: G5PL+9DfGwbW9n1+RDbSKE0BMqW7noZ5G6J50qwNm5c=
sender: 47hLwlm/ZhJqib6LFurLEANJRLazquQBpSkXrBRhH7A=
recipient: G5PL+9DfGwbW9n1+RDbSKE0BMqW7noZ5G6J50qwNm5c=
key : KMGFK4HR3IFSHYMYUIYZUYZRCZZPR444B6K7D3BNZCGJ7D6C30XA====
IV : F9A90A83DA5E986ACEE3EE4F7AEA8343
Hash 1 : QmPs2BsjB9VBuV6pHxdVYJP8fsnbLu8BTY93b9anwwW6SM
Hash 2 : QmfHCn3qnnWobcmmWrVZTioWW6UdWxMrUqkBbD1BiTihiv
pledge : 41757894681
pledge : 22565293942
pledge : 37926009743
ctl : dhs16k-Fb1Fc8XojtW2iMai3HGkMWLoycGQldwJTKZw=
pledge : 1
pledge : 1
pledge : 1
ctl : 1597649622.245917

```

CDM provider

CDM Consumer

Figure 8. The transaction log having a note CDM for registering public keys in SC-Blockchain.

```

ctl : VoU1cu01jVuq0vIWP7W3DewcdpVSsN5GNkBK8oc8Yv0=
ctl : 1597632805.06129
sender: TTJb/CupZryjGvVfp1jF5lCTAfvTGoQnNBmdM0ftxvA=
recipient: amvoFfwZtiuhvy2/gKGMiZYFzjA3Lbt+Pv0kw6AL+Vc=
key : KMGFK4HR3IFSHYMYUIYZUYZRCZZPR444B6K7D3BNZCGJ7D6C30XA====
IV : F9A90A83DA5E986ACEE3EE4F7AEA8343
Hash 1 : QmPs2BsjB9VBuV6pHxdVYJP8fsnbLu8BTY93b9anwwW6SM
Hash 2 : QmfHCn3qnnWobcmmWrVZTioWW6UdWxMrUqkBbD1BiTihiv
pledge : 29627965151
pledge : 21874816322
pledge : 2792546283
ctl : kZ5-MI70m7DS6CbwtANyRfvcijMMxR2uJoJ2HX0w9yw=
pledge : 1
pledge : 1
pledge : 1
ctl : 1597632806.075222
pledge : 42219012
pledge : 36400312541
pledge : 24556925213
ctl : 9-KiVVMjattbC5Xpgm33v1iRd0zEBZ-QVlZg30btRys=

```

Figure 9. The transaction log related with consensus.

Figure 10 shows the contents of the blockchain that consists of three blocks (0, 1, and 2) that are generated chronologically. The genesis block having is the first block in any blockchain-based protocol [39]. As can be seen in Figure 10, the block number 0 represents it. The block number 1

consists of a set of four CDM transactions, t_1 to t_3 , where t_1 is happened before t_2 and t_3 occurs lastly in the causal relationship. The output of the hash function, H , is the digest, a kind of hash value. For the CDM, the transaction t_1 is issued for registering the public key owned by the CDM provider. The transaction t_2 is issued for registering the public key owned by the CDM consumer. The transaction t_4 in the block b_2 represents the delivery of a signed CDM by the CDM provider. It is accessed by the CDM consumer for handling the encrypted CDM. Eventually a signed CDM is verified by a private key of the CDM consumer in SC-CDM that is used as a part of the transaction t_5 . One block can be added to the blockchain at a time [40]. Each block is verified to ensure that it follows in sequence from the previous block. All CDM transaction records are kept in the blockchain and are shared with peers, including some members of a CDM service system who have joined in a blockchain network. The block of transactions is committed by the leader.

Figure 10. The block consists of a set of transactions in SC-Blockchain.

3.5. Analysis of SC-CDM

In the cloud CDM model presented in this study, in order to increase security, CDM data are divided, and a multi-channel method is used. In the cloud CDM model, the multi-channel method is used by dividing each CDM datum into two pieces of data and storing each piece in IPFS, to receive two IPFS hashes.

When a CDM data owner transmits CDM data, each original CDM datum is divided into two pieces of data, and the divided data are encrypted by using a symmetric key and a random vector. Then, the encrypted segmented data are added to the IPFS network and received in the form of an IPFS hash. Finally, the symmetric key, random vectors, and IPFS hashes are all grouped into a single structure and sent to the requester of the CDM data, along with the signature, through the blockchain. After verifying the owner of the CDM data with a signature, the requestor retrieves the file divided by the IPFS hash of the received structure. After that, the encrypted file is decrypted by using the symmetric key and random vectors to obtain the divided CDM data. The requestor can normally receive the actual CDM data by recombining the CDM data divided by the same rule as the owner to obtain the original CDM data. Even if the requestor receives the IPFS file through the IPFS hash, the data undergo a process of decryption once more, so the information is not exposed except for users authorized by the CDM data owner. Figure 11 shows the overall process for building a secure CDM in SC-Blockchain. As shown Figure 11, H_1 and H_2 represent hash values of CDM_1 and CDM_2 . CDM_1

and CDM_2 are encrypted by the CDM Separate Block Encryption/Decryption Algorithm described in Section 2.2.4.

Figure 11. The overall process for building a secure CDM in SC-Blockchain.

In the IPFS network, an unchanging file is stored or downloaded, using an encryption hash function that returns a fixed value for the received data, and the user who will share CDM data is identified and signed through the encryption and decryption process, using the CDM partitioning algorithm. You can prove it through. Therefore, user access control is possible, and CDM data can be safely shared based on trust, because blockchain is used.

3.6. Performance Evaluation

By using IPFS to store files, the CDM cloud can provide a repository for CDM files with a method of persisting their CDM. When a CDM file is added to an IPFS node, the node returns a hash value of Content-Identifier (CID) for the CDM. The CID is the direct result of running the file through a cryptographic hash function. Every unique file will have a unique CID. This allows someone to retrieve a particular piece of content by providing the CID of the desired files. When a node is connected to the IPFS network, the content of that node is frequently broadcasted over the IPFS Distributed Hash Table (DHT). IPFS nodes treat stored data like a cache, meaning there is no guarantee that the data will continue to be stored, unless the CID is pinned.

In this experiment, CDM files imported into the SC-CDM are stored and shared in IPFS. Therefore, in order to evaluate the IPFS storage performance according to the size of the CDM file, the storage processing performance is examined according to the file size. When the IPFS node retrieves data from the network, it keeps a local cache of the data for future usage and takes up space on that particular

IPFS node. The used files for writing the CDM are designated to have the size of 1 KB, 10 KB, 100 KB, 1 MB, and 10 MB. Figure 12 presents the elapsed time for writing to IPFS that is linear with the file size.

Figure 12. Evaluation of the elapsed time writing to IPFS.

The fact that the elapsed time is greatly increased in a 10 MB file is explained by the characteristic that appears because IPFS stores blocks in units of 256 KB. In the case of 1 MB, four blocks are written, but in the case of 10 MB, the size of the block increases by 10 times to 40 blocks. In other words, since 1, 10, and 100 KB are stored in one block, it can be understood that there is little difference in write execution time.

The SC-CDM's data analysis is processed by using the analysis tools inside the SC-CDM. Registered researchers are required to access CDM materials for this purpose, and researchers using CDM perform IPFS node access that operates inside the SC-CDM. This differs from the way the current CDM delivers and operates data to remote researchers. In this case, researchers perform this experiment in terms of the performance of SC-CDM, which analyzes data through access to data within the SC-CDM. This experiment composes a swarm cluster with private network function after registering two IPFS nodes as peers, to show the benefits of CDM data access of SC-CDM. To evaluate the performance of the configured swarm cluster, the elapsed time of operation execution is measured by accessing the CDM data of the local IPFS node inside the SC-CDM. In addition, to evaluate the data-access elapsed time of the existing CDM operation method, the execution time of the CDM data-access operation in the remote IPFS node was compared. The elapsed data-access time of the remote IPFS node was to ignore network delays for objective performance evaluation. To this end, after passing the hash value to read data to another peer of the swarm cluster, the elapsed time of CDM data for data access was obtained.

The file is used for file accessing and is designated as the size of 1 MB, 10 MB, 100 MB, 200 MB, 500 MB, 750 MB, and 1 GB. After writing 20 consecutive times, leverage the elapsed time average and standard deviation of the execution time are determined, obtained as an experimental result. Table 4 presents the evaluation of accessing CDM data from IPFS in a swam cluster.

Table 4. Evaluation of storing CDM into a private IPFS network.

No.	The Size of CDM	SC-CDM (Local IPFS)		Conventional Method (Remote IPFS)		Comparison of Elapse Time
		Elapsed Time	Standard Deviation	Elapsed Time	Standard Deviation	
1	1 MB	0.263663	0.010216	0.234719	0.010720	1.079860
2	10 MB	0.289340	0.010536	1.056073	0.006692	3.649938
3	100 MB	1.102761	0.019368	9.671313	0.012963	8.770090
4	200 MB	2.066382	0.021744	19.249866	0.026164	9.315734
5	500 MB	4.934743	0.045454	48.105810	0.028492	9.748392
6	750 MB	7.296134	0.029448	72.207809	0.036717	9.896722
7	1 GB	9.931975	0.058510	98.620600	0.049551	9.929606

Figure 13 shows the standard deviation of elapsed time and elapsed time (blue line) when CDM clients access the SC-CDM local IPFS node. In the traditional CDM environment, CDM clients showed standard deviation of elapsed time and elapsed time (yellow line) when accessing remote IPFS nodes. As shown in Figure 13, the elapsed time of data access of local IPFS and remote IPFS increases relatively as the size of CDM data increases. Table 4 shows the performance comparison of SC-CDM (local access) and conventional method (remote access), and it can be observed that the execution time of remote access increases as the CDM file increases. The CDM file showed 1.08 times the elapsed time for 1 MB and 3.65 to 8.77 times for 100 MB from 10 MBs. From 200 MB to 1 GB, access to remote files increased from 9.32 to 9.93 times. This experiment confirmed that the method is excellent for maintaining the CDM data inside the SC-CDM.

Figure 13. Evaluation of the elapsed time.

4. Discussion

In this paper, the basic model of CDM in Figure 1 for common operation of data of EMR data from various hospitals was extended to the cloud-based CDM model to design the architecture. Moreover, SC-CDM was proposed to provide safe operation of CDM data. To ensure data integrity, CDM data are

encrypted and stored in IPFS and accessed from IPFS, to perform decryption. In terms of performance, the write performance of blocks in CDM data increased linearly with the creation time of the block ID of P2P in IPFS. In order to confirm the superiority of SC-CDM access to CDM data, a private network swarm cluster was constructed. Compared to the existing SC-CDM, the SC-CDM approach has been confirmed to be superior because the access time is about 10 times less when accessing 1 GB CDM.

Increasing demands for co-use of data in hospitals are driving CDM deployments, and SC-CDMs proposed in building networks involving multiple hospitals can be used as a major reference model.

5. Conclusions

CDM for data sharing and utilization of medical institutions requires access to patient medical information. It is used for disease research and customized medical care. In order to obtain meaningful results in research-network-based clinical studies on various patient data in hospitals, information assets need to be shared and be utilized. The cloud CDM provides interoperability for the participation of multiple hospitals and serves as an information-based research for customized and user-centered healthcare. However, reliable management of safe and transparent medical information of personal information is required. The cloud CDM model proposed in this paper, SC-CDM, applies distributed ID and blockchain technology for secure access control that occurs during CDM conversion and medical records access for joint use of hospital data. In this paper, experiments were conducted with storing and maintaining CDM data in IPFS. This confirmed that the data were fragmented and maintained, ensuring their safety. The result of the transaction is stored in the blockchain. Through this, it was found that the data-storage performance of the CDM composed of pieces was improved.

Author Contributions: Conceptualization, Y.B.P. and Y.K.; methodology, Y.B.P.; software, Y.K.; validation, Y.K., Y.B.P. and J.H.C.; formal analysis, Y.K.; investigation, J.H.C.; resources, Y.B.P.; data curation, Y.B.P.; writing—original draft preparation, Y.K.; writing—review and editing, J.H.C.; visualization, Y.K.; supervision, J.H.C.; project administration, J.H.C.; funding acquisition, J.H.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Institute of Korea Health Industry Development Institute (KHIDI), grant number HI19C1032, and The APC was funded by Ministry of Health and Welfare (MOHW).

Acknowledgments: This work was supported by Institute of Korea Health Industry Development Institute (KHIDI) grant funded by the Korea government (Ministry of Health and Welfare, MOHW) (No. HI19C1032, Development of autonomous defense-type security technology and management system for strengthening cloud-based CDM security).

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

References

- Shivade, C.; Raghavan, P.; Fosler-Lussier, E.; Embi, P.J.; Elhadad, N.; Johnson, S.B.; Lai, A.M. A review of approaches to identifying patient phenotype cohorts using electronic health records. *J. Am. Med. Inform. Assoc.* **2014**, *21*, 221–230. [[CrossRef](#)] [[PubMed](#)]
- You, S.C.; Lee, S.; Cho, S.-Y.; Park, H.; Jung, S.; Cho, J.; Yoon, D.; Park, R.W. Conversion of National Health Insurance Service-National Sample Cohort (NHIS-NSC) Database into Observational Medical Outcomes Partnership-Common Data Model (OMOP-CDM). *Stud. Health Technol. Inf.* **2017**, *245*, 467–470.
- Muralidharan, S.; Ko, H. An InterPlanetary File System (IPFS) based IoT framework. In Proceedings of the 2019 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 11–13 January 2019; pp. 1–2.
- Pănescu, A.-T.; Manta, V. Smart Contracts for Research Data Rights Management over the Ethereum Blockchain Network. *Sci. Technol. Libr.* **2018**, *37*, 235–245. [[CrossRef](#)]
- Androulaki, E.; Barger, A.; Bortnikov, V.; Cachin, C.; Christidis, K.; De Caro, A.; Enyeart, D.; Ferris, C.; Laventman, G.; Manevich, Y.; et al. Hyperledger fabric: A distributed operating system for permissioned blockchains. In Proceedings of the Thirteenth EuroSys Conference, Porto, Portugal, 23–26 April 2018; ACM: New York, NY, USA, 2018; p. 30.

6. Dagher, G.G.; Mohler, J.; Milojkovic, M.; Marella, P.B. Ancile: Privacy-preserving framework for access control and interoperability of electronic health records using blockchain technology. *Sustain. Cities Soc.* **2018**, *39*, 283–297. [CrossRef]
7. Silberschatz, A.; Korth, H.F.; Sudarshan, S. *Database System Concepts*; McGraw-Hill: New York, NY, USA, 1997.
8. Xiong, L.; Li, F.-G.; Zeng, S.-K.; Peng, T.; Liu, Z.-C. A Blockchain-Based Privacy-Awareness Authentication Scheme with Efficient Revocation for Multi-Server Architectures. *IEEE Access* **2019**, *7*, 125840–125853. [CrossRef]
9. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. May 2009. Available online: <http://www.bitcoin.org/bitcoin.pdf> (accessed on 27 August 2020).
10. Shrestha, A.K.; Vassileva, J. Blockchain-Based Research Data Sharing Framework for Incentivizing the Data Owners. In Proceedings of the International Conference on Blockchain, Seattle, WA, USA, 25–30 June 2018; Lecture Notes in Computer Science. Springer: Berlin/Heidelberg, Germany; Volume 10974, pp. 259–266.
11. Allen, C. The Path to Self-Sovereign Identity. 2016. Available online: <http://www.lifewithalacrity.com/2016/04/the-path-to-self-sovereign-identity.html> (accessed on 27 August 2020).
12. UK Government. *Distributed Ledger Technology: Beyond Block Chain (PDF) (Report)*; UK Government, Office for Science: London, UK, January 2016.
13. Augot, D.; Chabanne, H.; Chenevier, T.; George, W.; Lambert, L. A User-Centric System for Verified Identities on the Bitcoin Blockchain. In *Data Privacy Management, Cryptocurrencies and Blockchain Technology*; Springer: Oslo, Norway, 2017; Volume 10436, pp. 390–407.
14. Halpin, H. NEXTLEAP: Decentralizing Identity with Privacy for Secure Messaging. In Proceedings of the 12th International Conference on Availability, Reliability and Security, Reggio Calabria, Italy, 29 August–1 September 2017; pp. 1–10.
15. David, B.; Gaži, P.; Kiayias, A.; Russell, A. Ouroboros Praos: An Adaptively-Secure, Semi-synchronous Proof-of-Stake Blockchain. In Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques, Dan Panorama Tel-Aviv, Tel Aviv, Israel, 29 April–3 May 2018; pp. 66–98.
16. Hardjono, T.; Pentland, A. Verifiable Anonymous Identities and Access Control in Permissioned Blockchains. *arXiv* **2019**, arXiv:1903.04584.
17. Pass, R.; Shi, E. Thunderella: Blockchains with Optimistic Instant Confirmation. In Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques, Dan Panorama Tel-Aviv, Tel Aviv, Israel, 29 April–3 May 2018; Springer: Cham, Switzerland, 2018; pp. 3–33.
18. Garay, J.; Kiayias, A.; Leonardos, N. The Bitcoin Backbone Protocol with Chains of Variable Difficulty. In Proceedings of the 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, 20–24 August 2017; Springer: Cham, Switzerland, 2017; pp. 291–323.
19. Pass, R.; Seeman, L.; Shelat, A. Analysis of the Blockchain Protocol in Asynchronous Networks. In Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, 30 April–4 May 2017; Springer: Cham, Switzerland, 2017; pp. 643–673.
20. Badertscher, C.; Maurer, U.; Tschudi, D.; Zikas, V. Bitcoin as a Transaction Ledger: A Composable Treatment. In Proceedings of the 37th Annual International Cryptology Conference, Santa Barbara, CA, USA, 20–24 August 2017; Springer: Cham, Switzerland, 2017; pp. 324–356.
21. Javed, M.U.; Rehman, M.; Javaid, N.; Aldegheishem, A.; Alrajeh, N.; Tahir, M. Blockchain-Based Secure Data Storage for Distributed Vehicular Networks. *Appl. Sci.* **2020**, *10*, 2011. [CrossRef]
22. Seol, Y.; Ahn, J.; Park, S.; Ji, M.; Chae, H.; Yi, J.; Paek, J. DDNB—Doubly Decentralized Network Blockchain Architecture for Application Services. *Appl. Sci.* **2020**, *10*, 5212. [CrossRef]
23. Vukolić, M. *The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication*. In *International Workshop on Open Problems in Network Security*; Springer: Cham, Switzerland, 2015; pp. 112–125.
24. Tschorsch, F.; Scheuermann, B. Bitcoin and Beyond: A Technical Survey on Decentralized Digital Currencies. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 2084–2123. [CrossRef]
25. Schneeweiss, S.; Brown, J.S.; Bate, A.; Trifirò, G.; Bartels, D.B. Choosing Among Common Data Models for Real-World Data Analyses Fit for Making Decisions About the Effectiveness of Medical Products. *Clin. Pharmacol. Ther.* **2020**, *107*, 827–833. [CrossRef] [PubMed]

26. Glicksberg, B.S.; Oskotsky, B.; Thangaraj, P.M.; Giangreco, N.; Badgeley, M.A.; Johnson, K.W.; Datta, D.; Rudrapatna, V.A.; Rappoport, N.; Shervey, M.M.; et al. PatientExploreR: An extensible application for dynamic visualization of patient clinical history from electronic health records in the OMOP common data model. *Bioinformatics* **2019**, *35*, 4515–4518. [CrossRef] [PubMed]
27. Choi, W.; Ko, S.J.; Jung, H.J.; Kim, T.M.; Choi, I. Expansion of EHR-Based Common Data Model (CDM). *Stud. Health Technol. Inf.* **2019**, *264*, 1443–1444.
28. Ji, H.; Kim, S.; Yi, S.; Hwang, H.; Kim, J.-W.; Yoo, S. Converting clinical document architecture documents to the common data model for incorporating health information exchange data in observational health studies: CDA to CDM. *J. Biomed. Inf.* **2020**, *107*, 103459. [CrossRef] [PubMed]
29. Yoon, D.; Ahn, E.K.; Park, M.Y.; Cho, S.Y.; Ryan, P.; Schuemie, M.J.; Shin, D.; Park, H.; Park, R.W. Conversion and Data Quality Assessment of Electronic Health Record Data at a Korean Tertiary Teaching Hospital to a Common Data Model for Distributed Network Research. *Health Inf. Res.* **2016**, *22*, 54–58. [CrossRef] [PubMed]
30. Lamer, A.; DePas, N.; Doutreligne, M.; Parrot, A.; Verloop, D.; Defebvre, M.-M.; Ficheur, G.; Chazard, E.; Beuscart, J.-B. Transforming French Electronic Health Records into the Observational Medical Outcome Partnership's Common Data Model: A Feasibility Study. *Appl. Clin. Inf.* **2020**, *11*, 013–022.
31. Kim, H.K.; Jeong, E.Y.; Park, D.K. Study on HIPAA PHI application method to protect personal medical information in OMOP CDM construction. *J. KINGComput.* **2017**, *13*, 66–76.
32. Shin, S.J.; You, S.C.; Park, Y.R.; Roh, J.; Kim, J.-H.; Haam, S.J.; Reich, C.G.; Blacketer, C.; Son, D.-S.; Oh, S.; et al. Genomic Common Data Model for Seamless Interoperation of Biomedical Data in Clinical Practice: Retrospective Study. *J. Med. Internet Res.* **2019**, *21*, e13249. [CrossRef]
33. Kattamuri, M. Block Cipher Modes of Operation. Available online: <https://www.geeksforgeeks.org/block-cipher-modes-of-operation> (accessed on 27 August 2020).
34. Benet, J. IPFS—Content Addressed, Versioned, P2P File System. *arXiv* **2017**, arXiv:1407.3561.
35. Khatal, S.; Rane, J.; Patel, D.; Patel, P.; Busnel, Y. FileShare: A Blockchain and IPFS Framework for Secure File Sharing and Data Provenance. In Proceedings of the Computing Algorithms with Applications in Engineering; Springer: Cham, Switzerland, 2020; pp. 825–833.
36. Aydar, M.; Cetin, S.C.; Ayvaz, S.; Aygun, B. Private key encryption and recovery in blockchain. *arXiv* **2019**, arXiv:1907.04156.
37. Jayaraman, I.; Mohammed, M. Secure Privacy Conserving Provable Data Possession (SPC-PDP) framework. *Inf. Syst. e-Bus. Manag.* **2019**, 1–27. [CrossRef]
38. Park, C.; Kim, H.; Hong, D.; Seo, C. A Symmetric Key Based Deduplicatable Proof of Storage for Encrypted Data in Cloud Storage Environments. *Secur. Commun. Netw.* **2018**, *2018*, 2193897. [CrossRef]
39. TetraCoin. What Is Genesis Block and Why Genesis Block Is Needed? Available online: <https://medium.com/@tecracoin/what-is-genesis-block-and-why-genesis-block-is-needed-1b37d4b75e43> (accessed on 27 August 2020).
40. Zhang, R.; Xue, R.; Liu, L. Security and Privacy on Blockchain. *ACM Comput. Surv.* **2019**, *52*, 1–34. [CrossRef]