applied sciences

MDPI

Article

Secure Outsourced Blockchain-Based Medical Data Sharing System Using Proxy Re-Encryption †

Young-Hoon Park [1,*,‡], Yejin Kim [1], Shin-Ok Lee [1] and Kwangman Ko [2,*]

[1] Division of Computer Science, Sookmyung Women's University, 100 Cheongpa-ro 47-gil, Yongsan-gu, Seoul 04310, Korea; dpwls7753@sookmyung.ac.kr (Y.K.); shinok.lee@sookmyung.ac.kr (S.-O.L.)
[2] School of Computer and Information Engineering, Sangji University, 83 Sangjidae-gil, Wonju-si 26339, Korea
* Correspondence: yh.park@sookmyung.ac.kr (Y.-H.P.); kkman@sangji.ac.kr (K.K.); Tel.: +82-2-2077-7326 (Y.-H.P.); +82-33-730-0486 (K.K.)
† This paper is an extended version of our paper published in International Conference on Green and Human Information Technology 2021.
‡ Current address: 101, Saehim-Hall, 100, Cheongpa-ro 47-gil, Yongsan-gu, Seoul 04310, Korea.

Abstract: The security and privacy of electronic health records (EHRs) have received considerable attention from healthcare workers and researchers. To ensure security, various encryption and decryption schemes as well as key management protocols have been developed. However, owing to sharing and scalability issues, additional security technologies have been proposed. Nonetheless, these technologies cause other problems, such as efficiency issues. Blockchain-based EHR management systems have been proposed to overcome computational overhead. However, because most blockchain systems are installed by outsourcing companies, EHRs may be leaked to the company. Hence, we herein propose a blockchain-based EHR management scheme with proxy re-encryption. In this scheme, we set a proxy server that re-encrypts the ciphertext between file servers, thereby solving EHR sharing issues. Furthermore, because the server is separated from the blockchain system, the outsourcing company cannot manipulate the server or access the records. In addition, the blockchain assists in access control by using smart contracts, thereby enabling secure and efficient EHR sharing. By performing security analysis, we prove that our proposed scheme solves the aforementioned security problems. In addition, we experimentally demonstrate the efficient operation of the proposed system.

Keywords: EHR; blockchain; proxy re-encryption

Citation: Park, Y.-H.; Kim, Y.; Lee, S.-O.; Ko, K. Secure Outsourced Blockchain-Based Medical Data Sharing System Using Proxy Re-Encryption. *Appl. Sci.* **2021**, *11*, 9422. https://doi.org/10.3390/app11209422

Academic Editor: Byung-Seo Kim

Received: 29 August 2021
Accepted: 3 October 2021
Published: 11 October 2021

1. Introduction

Preserving the security and privacy of massive medical data is a key issue in most healthcare institutes [1–3]. A significant amount of medical data are being created daily at an increasing speed [4]. Therefore, the system that efficiently manages the significant volume of medical data must be focused on. However, both the efficiency and security of such medical data cannot be overcome using only existing cryptography schemes simultaneously [5].

To maintain confidentiality, all medical data are encrypted using a symmetric encryption scheme, such as the advanced encryption standard (AES), and the symmetric key is encrypted via an asymmetric encryption scheme [6,7]. In addition, access control is employed for the efficient management of authority [8]. Moreover, security issues in healthcare systems with mobile devices are addressed, and possible solutions are provided in [9,10]. However, when using these classical cryptography schemes, issues arising from massive medical data and the efficient sharing of problems cannot be solved [8]. To overcome such problems, blockchain has been proposed recently [5].

Blockchain is a type of data structure that guarantees the integrity of stored data. Blockchain was first used for managing transaction data, and this is known as cryptocur-

rency, e.g., Bitcoin [11] and Ethereum [12]. Blockchain is used to store not only the data of currency transactions, but also various types of historical data such as the distribution process, status of shared items, security protocols [13], and electronic healthcare data [14,15]. In particular, many blockchain-based medical data management systems have been proposed, such as MedRec [16] and MedShare [17].

However, the proposed systems exhibit several vulnerabilities. MedRec [16] and its extended version [18] employ a public blockchain; therefore, efficiency may be a concern. In addition, MedShare operates access control only with the blockchain; hence, encrypted medical data can be delivered to unauthorized users, and efficiency problems due to decryption and encryption may occur [17,19].

Other solutions for overcoming both privacy and sharing issues have been proposed. Li et al. provided a framework that preserves personal health records based on attribute-based encryption (ABE) [20], and Liu et al. proposed an extended version that employs attribute-based signcryption (ABSC) [21]. Furthermore, Li et al. developed a homomorphic encryption-based electronic health record (EHR) management system [22]. However, advanced encryption schemes such as ABE and homomorphic encryption involve computational overhead; therefore, they are not suitable for managing large volumes of medical data.

The solutions presented above for secure medical data management systems cannot overcome the computational overhead problem owing to the outsourcing of blockchain [23,24]. Most healthcare institutes are unable to develop systems that include blockchain and cryptography schemes owing to insufficient IT expertise [25]. However, when an outsourcing company is employed to implement a file server and a blockchain server, the healthcare data stored in the file server may be accessed by the company. In particular, this occurs when a decryption key is lost, or the stored data are deleted because of security attacks, natural disasters, etc. In this case, the data stored by the outsourcing company may be leaked. Therefore, a security solution that prevents the company from accessing privacy-sensitive healthcare data is required.

Proxy re-encryption (PRE) is a promising solution to this issue. It transfers a ciphertext into another ciphertext, enabling the receiver to decrypt the encrypted data with its own secret key [26]. This method does not reveal any sensitive data, including plaintext or secret keys of the sender or receiver, during the entire process. Unlike the conventional encryption method, where the ciphertext must be decrypted and re-encrypted to enable the receipt of a message, PRE does not require decryption. Therefore, it is a secure method for use in circumstances where data must be shared with third parties in an untrusted network [27].

PRE is a key solution to the problem of medical data leakage by outsourcing companies. In our proposed system, we set a proxy server between the data server and the end user and separated the proxy from the outsourcing company. Consequently, we were able to design a system that prevents the company from viewing EHRs in the system.

In this study, we employed a PRE scheme to encrypt the symmetric key and achieve efficient access control. Basically, all EHRs are encrypted with symmetric keys, and each key is encrypted using the public key cryptography system as a common secure communication protocol such as SSL/TLS. In addition to this system, we adopt the blockchain to protect the original EHR and to provide access control for the case of data sharing to third parties. Furthermore, to prevent exposure of the EHR to the blockchain outsourcing company, we add the PRE [28] to the blockchain-based EHR management. In our system, the proxy server is separated from the blockchain scope, so the outsourcing company cannot interfere the PRE. The symmetric key is secured by the encryption process of the PRE and would be re-encrypted by the proxy when the legal user requests to receive the EHR.

We first developed this blockchain-based model with PRE in [29], and in this work, we extend the proposed system. Detailed protocols and algorithms for managing healthcare data are presented herein. The main contributions of this study are as follows:

1. We first propose a blockchain-based EHR management model. In each distributed ledger, the access information that reveals the user that can read the corresponding EHR and the hash value of the record are stored. Hence, only authorized users can download and retrieve the encrypted EHR;
2. In addition to the blockchain, we adopt a PRE scheme to enable secure and efficient EHR sharing. Moreover, the proxy is separated from the blockchain system, and the re-encryption key is generated and maintained by the system manager, not the outsourcing company. Hence, the company cannot open the encrypted EHRs;
3. We provide essential cyptography schemes to secure the EHRs, such as encryption/decryption and digital signature schemes. Hence, the confidentiality and integrity of the EHRs are guaranteed.

The remainder of this paper is organized as follows: We first discuss the relevant studies in Section 2. In Section 3, we introduce the essential items for the proposed schemes, such as the blockchain and PRE. In Section 4, the blockchain-based EHR management scheme using PRE is described. In Section 5, the security improvements of the proposed scheme as well as the overall performance of the proposed scheme are discussed. Finally, the conclusions of this study and future studies are provided in Section 6.

2. Related Works

In this section, we present previous studies that are relevant to the current study. We first introduce studies pertaining to blockchain-based EHR management schemes, followed by those associated with EHR management with PRE.

2.1. Blockchain Implementations on Medical Data

To guarantee the integrity of healthcare data, many blockchain technologies have been employed in data management systems. MedRec provides safe storage and EHR sharing for patients served by different providers. MedRec employs Ethereum to manage access control, where only users who have access rights can access the healthcare data [16]. In addition, Yang et al. incorporated the ABSC scheme to strengthen the confidentiality and integrity of medical data [18]. The symmetric key used to encrypt the medical data was encrypted using ABE, and a signature for both the encrypted data and encrypted key was created. However, these schemes are disadvantageous in that they depend on the proof of work and cryptocurrency.

Chelladurai et al. employed the blockchain to guarantee the integrity of the EHR [30], and Ismail et al. proposed a healthcare record management framework to provide not only security, but also privacy. Moreover, there are several blockchain-based works for secure EHR sharing. Dubovitskaya et al. [31] presented a blockchain framework for managing and sharing EHR data between healthcare providers and researchers. Xiaodong et al. [32] proposed a method of sharing EHR by storing encrypted data in the cloud and saving information, including addresses, to the blockchain. Moreover, ABE and signatures were combined to achieve sharing data in many-to-many communications. Wang et al. [33] presented a protocol for securely sharing EHRs of data stored in the cloud based on a consortium blockchain with searchable encryption and PRE. MeDShare [17] monitors all actions, including data transition and sharing, and records them in the blockchain. This system uses smart contracts and access control to track the behavior of the data and enables the withdrawal of dishonest access. By enabling data owners to control the anonymization process, Yang et al. [34] proposed a method for sharing private data with blockchain in a cloud federation. However, problems caused by outsourcing blockchain have not been addressed.

2.2. Proxy Re-Encryption for Medical Data Using Blockchain

In addition to the aforementioned proposals, methods for implementing PRE using blockchain and personal medical data have been proposed to address security issues. Xiao et al. [35] proposed a blockchain-based scheme for sharing and protecting EHRs using

PRE. The participants of the system include a hospital, user, and system manager, where the hospital maintains its own private blockchain, and the system manager generates the master key and system parameters as a trusted third party. The user stores his/her medical data in the hospital's blockchain by joining the network, and the system manager re-encrypts the data when the data are requested by other doctors. Thwin et al. suggested a personal health record (PHR) system for secret data sharing that involved the blockchain and PRE [36]; furthermore, they developed an access-control model for PHR systems [37]. This model stores the encrypted PHR for cloud storage, and the PHR metadata are stored in the blockchain with access logs. The gateway server re-encrypts the data requests from the cloud storage and sends it to the user client. Dagher et al. proposed Ancile [38], which is an EHR access-control framework involving PRE and the blockchain. However, none of the previous works discussed possible security attacks due to the untrustworthy blockchain outsourcing companies.

3. Preliminaries

3.1. Blockchain

To guarantee the integrity of the EHR and to enable efficient and secure access control, we employed blockchain in an EHR management system. Blockchain was first explained by Satoshi Nakamoto in 2008 as a Bitcoin technique for an electronic cash system [11]. It is a decentralized peer-to-peer network operated by distributed nodes. In a blockchain network, every node contains ledgers with identical data. The data recorded in blocks cannot be modified and are resistant to external attacks. Hence, blockchain has garnered the attention of researchers in various fields [39,40].

Figure 1 illustrates the structure of a general blockchain. The blocks, which comprise transaction lists and block information, are linked in chronological order, and identical data are contained by the nodes participating in the network. After the data are saved in the blockchain, they cannot be altered. The hash of the previous block is stored in the next new block to guarantee the integrity of the previous data in the blockchain, and a digital signature generated via a public-secret key method enables the verification of nodes that are deploying the transaction. The time stamp and Merkle root enhance the data integrity and prevent forgery. The time stamp prevents the double-spending problem of blocks, whereas the Merkle root proves the integrity of transactions within each block [41]. Therefore, we can conclude that, unlike the conventional centralized database, blockchain provides data integrity by rendering information available to the public [42]. This transparency and distribution are vital to data integrity, rendering data forgery extremely difficult.

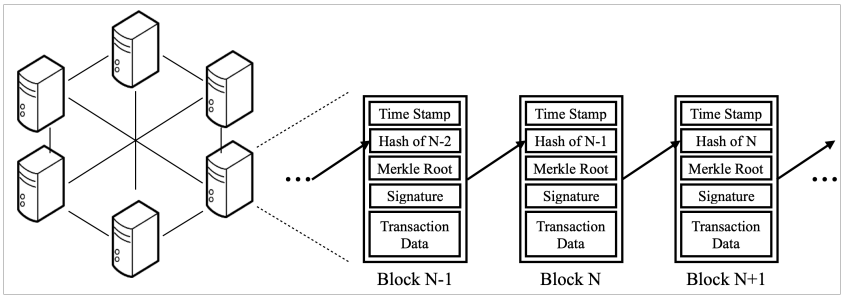


Figure 1. Blockchain structure.

Another critical function of the blockchain is smart contracts, which are digital contracts that are executed automatically once they fulfill the required condition [43]. This ideal contract obviates the requirement for a third party in transactions; furthermore, it is written and operated by all nodes in the distributed network without the control of a single entity once it is uploaded. We used smart contracts to upload and retrieve data from the blockchain and interplanetary file system.

A decentralized application (DApp) enables interactions between users and the blockchain network. It is similar to the conventional application, which is structured

with a front end and back end. However, compared with an application that executes the back end on a centralized server, the DApp is implemented on a distributed peer-to-peer network. Smart contracts are executed on a DApp, providing convenience to users who are willing to interact with the blockchain through smart contracts [44].

In this paper, we employ Ethereum [12] to realize a blockchain-based EHR management system which protects the distributed ledgers and enables smart contracts. In addition, to monitor the blockchain, we build a DApp using Web-based programming.

3.2. Proxy Re-Encryption

PRE is a scheme that enables a proxy to transform a ciphertext that can be decrypted by one user into another ciphertext that can be decrypted by another user without decryption. Since its introduction by Blaze et al. in 1998 [26], various versions of the PRE have been proposed. PRE is based on a public key cryptography system; as such, the encryption and decryption keys are dissimilar. After encryption, the data can be re-encrypted without decryption by a proxy; hence, the plaintext is not revealed during the PRE, and the confidentiality of the ciphertext is preserved. The basic concept of the PRE scheme is illustrated in Figure 2.

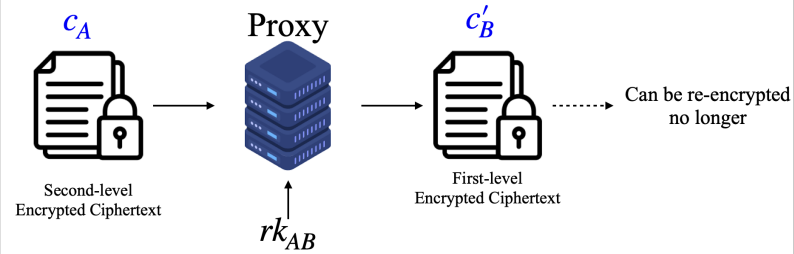


Figure 2. Process of unidirectional proxy re-encryption.

Let pk_A and sk_A be Alice's encryption and decryption keys, respectively, and let pk_B and sk_B be the encryption and decryption keys for Bob, respectively. It is noteworthy that sk_A and sk_B are maintained only by Alice and Bob, respectively. Moreover, for a message M, let c_X be an encrypted version of M that can be decrypted with sk_X, where $X \in \{A, B\}$.

Conventionally, to allow Bob to decrypt c_A, Alice first decrypts with sk_A and encrypts with pk_B, or may employ a third party, which is known as a proxy. In the latter case, Alice first segregates sk_A between Bob and the proxy, whereas c_A is partially decrypted by the proxy and then finally decrypted by Bob [45,46]. However, this approach requires Bob to possess an additional secret key for every delegation that he accepts [47]. Therefore, a PRE scheme was proposed.

The concept of the PRE scheme is as follows. First, four parties are involved: a delegator (Alice), a delegate (Bob), a key distributor, and a proxy. The aim of the key distributor is to generate and distribute keys that include the keys of various participants (pk_X and sk_X) and the re-encryption key (rk). As shown in Figure 2, let us consider the case in which Alice delegates the access authority of message M to Bob. The aim of the proxy is to re-encrypt c_A to c_B using only rk without decryption.

After its introduction by Blaze, Bleumer, and Strauss, the PRE scheme has received considerable attention, and more developed PRE schemes have been developed thereafter. Generally, the PRE scheme can be categorized into two types: unidirectional and bidirectional. The bidirectional PRE scheme allows users to re-encrypt the ciphertext repeatedly, whereas the unidirectional PRE allows only a one-time re-encryption. The first proposed method is bidirectional PRE. However, the bidirectional PRE scheme has a few shortcomings: the delegation in the scheme is transitive and the delegator's secret key can be completely recovered if the proxy and delegate collude. Hence, systems that treat privacy-sensitive data employ unidirectional PRE schemes [28]. In the proposed scheme, we used unidirectional PRE because the re-encrypted data can no longer be re-encrypted.

4. The Proposed Scheme

In this section, we discuss an EHR management system that uses a blockchain and the PRE algorithm. First, we briefly introduce the system model; subsequently, we describe the processes for data creation, storage, retrieval, and sharing.

4.1. System Model

Figure 3 shows the model of the proposed system. The proposed medical data management system is composed of a user, server, proxy server, and third party. In addition, as shown in Figure 3, we assume that all systems except the proxy server are installed by an outsourcing blockchain company. In other words, the outsourcing company's role is to install and manage the blockchain and file server for the EHRs.

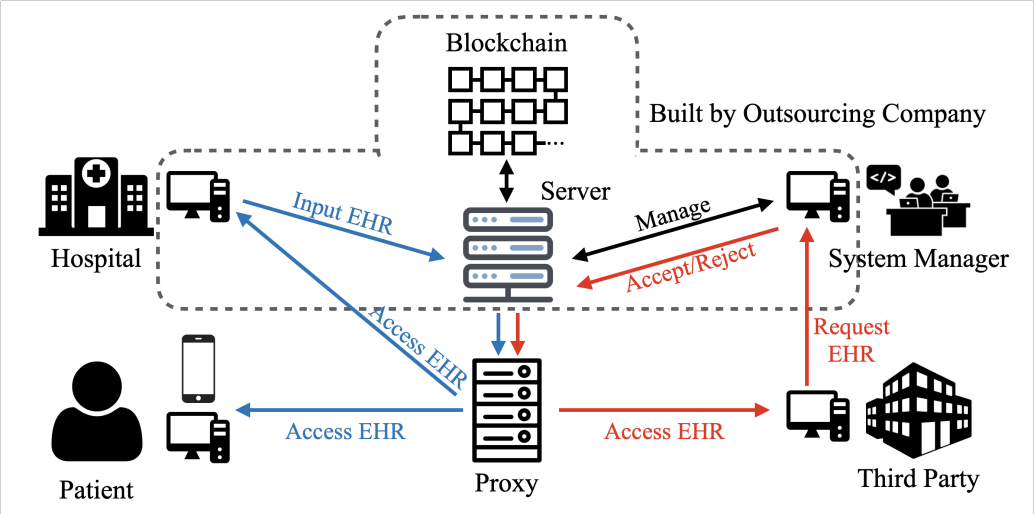
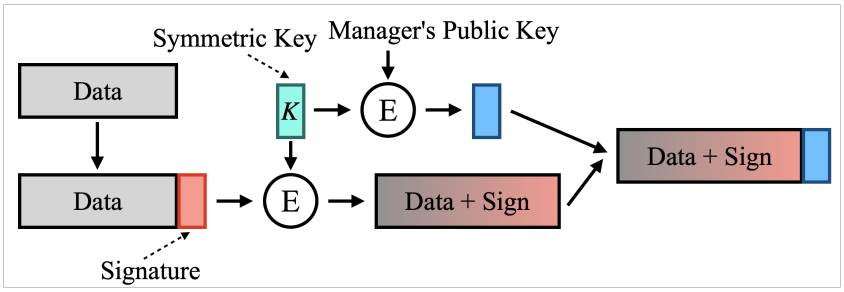


Figure 3. System model.

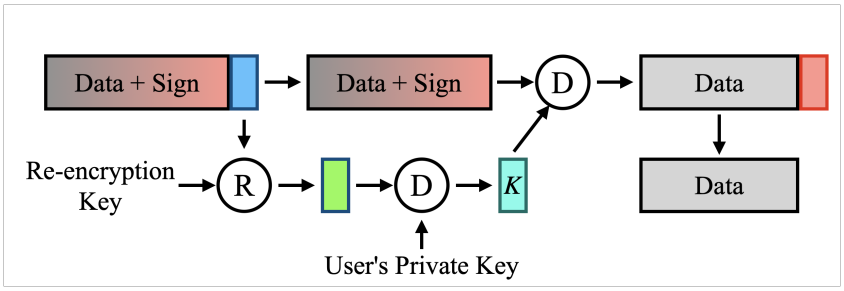
All EHRs are stored in the server. Before being saved, the signature is attached to the EHR, the combined data are encrypted using a symmetric encryption scheme, and the symmetric key is encrypted using a public key encryption scheme. Figure 4a shows the encryption process. Because this system employs the PRE, we selected the second level of the PRE as the public key encryption scheme. To obtain the EHR, the symmetric key should be decrypted using PRE decryption (Section 4.3.4) or re-encryption (Section 4.3.5) and another PRE decryption (Section 4.3.6). Figure 3 shows the digital signature, encryption, decryption, and re-encryption processes. The detailed algorithms for encrypting, re-encrypting, and decrypting the symmetric key via PRE are provided in Section 4.3.

In addition, when the EHR is stored, shared, and accessed, all of the log data for the EHRs are recorded in the distributed ledger of the blockchain. Moreover, information regarding participants that can access the EHR is stored in the distributed ledger. Because the blockchain guarantees the integrity of the entire managed data, no participants, except the system manager, can modify or delete the log data and access information.

However, because these servers and blockchain systems are installed by an outsourcing company, all keys that include the private key can be managed by the company. This may be necessary, for example, if one of the important participants loses a private key and requests a key to access the stored data. However, if the outsourcing company abuses this privilege, sensitive data may be leaked even if they are encrypted and managed by the blockchain system, because the company has all the information.



(**a**) Encryption.



(**b**) Re-encryption and decryption.

Figure 4. Encryption and decryption processes for EHR.

To prevent this threat, we added a proxy server to the system. As shown in Figure 3, we installed a proxy server between the data server and end users. The aim of the proxy server is to re-encrypt the symmetric key. Originally, the symmetric key is retrieved by the system manager; however, after the proxy, it is reformed to a ciphertext that can be decrypted by the end user. In this regard, a re-encrypted key is input to the proxy.

Figure 4b shows the re-encryption and decryption processes of the symmetric key. Originally, the symmetric key is encrypted with the manager's public key using the second-level encryption. When a user requests the healthcare data, the encrypted symmetric key is re-encrypted and modified to another ciphertext that can be decrypted with the user's private key using first-level decryption. Subsequently, the user retrieves the symmetric key with his/her private key and finally decrypts the data using the symmetric key.

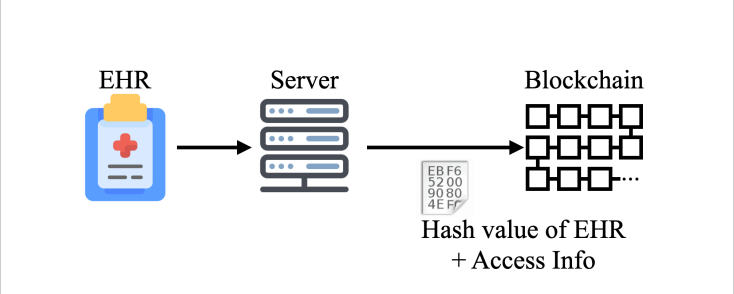
4.2. Blockchain in the System

In the proposed system, the blockchain is utilized for two purposes: to verify the integrity of the EHRs and for access control. In each block of the blockchain in our system, hash values of the records and access authority information are stored, and they are protected by the robust security of the blockchain system.

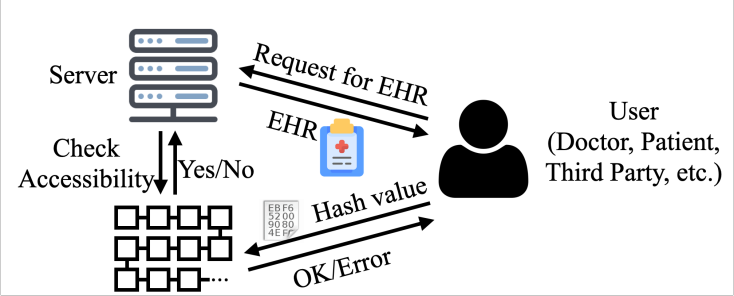
Figure 5 shows the interactions between the data server and the blockchain for cases in which a user saves and requests the EHRs, separately. In this figure, we assume that the user is permitted to use the record. In terms of the EHR storage process, as shown in Figure 5a, the hash value of the EHR is calculated and transmitted to the blockchain with other information, such as the access range, creation date, and saving date. When another user, such as a doctor, patient, or third party, enquires to use the EHR and receives the record as shown in Figure 5b, the user calculates the hash value of the record and sends it to the blockchain to verify the integrity. Subsequently, the blockchain server verifies whether the hash value of the corresponding data is consistent with the hash value in the blockchain and returns the result to the user.

In addition to integrity verification, the blockchain provides access control using a smart contract. As shown in Figure 5a, the information revealing the user that can access the corresponding record is transmitted to the blockchain. As shown in Figure 5b, when the server receives the data request from a user, it verifies the data accessibility from the

blockchain using the smart contract. Subsequently, when the blockchain returns a possible sign, the server delivers the EHR to the user.



(**a**) Saving EHR.



(**b**) Accessing EHR.

Figure 5. Interactions between server and blockchain.

4.3. The Re-Encryption Processes

Next, we present the processes for PRE in detail. We address the system initialization, user registration, data storage, data retrieval by the system manager, and data retrieval by an end user. The notations used in this subsection are listed in Table 1.

Table 1. Notations.

Symbols	Meaning
κ, ℓ_0, ℓ_1	Security parameters
p and q	Large primes satisfy $q\|(p-1)$, where length of q is κ
$\mathbb{G}$	Subgroup of $\mathbb{Z}_q^*$
H_1, H_2, H_3, and H_4	Hash functions
$g \in \mathbb{G}$	Generator
$\mathtt{sk}_M$, $\mathtt{pk}_M$	Manager's private and public keys
$\mathtt{sk}_i$, $\mathtt{pk}_i$	User u_i's private and public keys

4.3.1. System Initialization

Here, we describe the process for system initialization, which is performed after the system begins or is reset. In this process, all of the variables and functions are initialized, and the system is ready to receive users. As mentioned earlier, we employed Chow's model [28], which provides an efficient and secure unidirectional PRE, for our system.

1. The system manager selects two large primes p and q, where $q|(p-1)$. Let the number of digits in q be κ, and let $\mathbb{G}$ be a subgroup of $\mathbb{Z}_q^*$ with order q;

2. The manager selects four hash functions, $H_1 : \{0,1\}^{\ell_0} \times \{0,1\}^{\ell_1} \rightarrow \mathbb{Z}_q^*$, $H_2 : \mathbb{G} \rightarrow \{0,1\}^{\ell_0+\ell_1}$, $H_3 : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$, and $H_4 : \mathbb{G} \rightarrow \mathbb{Z}_q^*$. It is noteworthy that the security parameters ℓ_0 and ℓ_1 are determined by κ;
3. The manager sets `param` as $(q, \mathbb{G}, g, H_1, H_2, H_3, H_4, \ell_0, \ell_1)$.

4.3.2. User Registration

After system initialization, the system is ready to receive user registration. In this process, secret and public keys are generated, where each key is composed of two variables.

1. The system verifies the validity of user u_i, who tries to register;
2. If the validity test is passed, then the system assigns `param` to u_i;
3. User u_i selects $x_{i,1},\ x_{i,2} \in \mathbb{Z}_q^*$, and sets $\mathtt{sk}_i$ as $(x_{i,1},\ x_{i,2})$;
4. User u_i calculates $pk_{i,1} = g^{x_{i,1}} \mod q$ and $pk_{i,2} = g^{x_{i,2}} \mod q$, and sets public key $\mathtt{pk}_i$ as $(pk_{i,1}, pk_{i,2})$;
5. User u_i sends the public key to the system manager.

Specifically, we let $\mathtt{sk}_M = \{x_{M,1},\ x_{M,2}\}$ and $\mathtt{pk}_M = \{g^{x_{M,1}},\ g^{x_{M,2}}\}$ be the secret and public keys of the system manager, respectively.

4.3.3. Data Storage

When user u_i creates an EHR M and attempts to store it in the EHR management system, the processes involved are as follows:

1. User u_i signs M with one's private key. Let S be u_i's digital signature on M;
2. To encrypt both the message and the signature, user u_i randomly creates a symmetric key K for the AES and encrypts $(M\|S)$ with K. Let $\mathcal{C}$ be the encrypted data, i.e., $\mathcal{C} = Enc(K, (M\|S))$. In the next step, the second-level encryption of the PRE begins;
3. To encrypt K, user u_i selects $u \in \mathbb{Z}_q^*$ and $\omega \in \{0,\ 1\}^{\ell_1}$ randomly, and computes $r = H_1(K, \omega)$;
4. User u_i calculates the following:

$$\begin{aligned} D &= \left(pk_{M,1}^{H_4(pk_{M,2})} \cdot pk_{M,2}\right)^u, \\ E &= \left(pk_{M,1}^{H_4(pk_{M,2})} \cdot pk_{M,2}\right)^r, \\ F &= H_2(g^r) \oplus (K\|\omega); \end{aligned}$$

5. The user computes $s = (u + r \cdot H_3(D, E, F)) \mod q$, outputs the ciphertext $\mathcal{E} = (D,\ E,\ F,\ s)$, and sends $\mathcal{C}$ and $\mathcal{E}$ to the server;
6. After the server receives $\mathcal{C}$ and $\mathcal{E}$, it verifies if the following equation holds:

$$\left(pk_{M,1}^{H_4(pk_{M,2})} \cdot pk_{M,2}\right)^s = D \cdot E^{H_3(D,E,F)}.$$

If it does not hold, then the server rejects $\mathcal{C}$;
7. The server finally stores $\mathcal{C}$, and writes the access information on the blockchain.

4.3.4. Data Retrieval by System Manager

The system manager can retrieve the stored EHR. Because the manager decrypts the encrypted EHR using the second level of the PRE, the following process is performed. First, the manager decrypts the symmetric key and then decrypts the encrypted EHR using the symmetric key.

1. The system manager requests the server to send the encrypted data $\mathcal{C}$ and the encrypted key $\mathcal{E}$. Let $\mathcal{C}$ be $(D,\ E,\ F,\ s)$;
2. The manager calculates $F \oplus H_2(E^{\frac{1}{x_{M,1} \cdot H_4(pk_{i,2}) + x_{i,2}}})$. If no problem arises, then the result is two concatenated values; the first is the symmetric key K, and the second one is ω;

3. The manager verifies whether $E = \left(pk_{M,1}^{H_4(pk_{i,2})} \cdot pk_{M,2}\right)^{H_1(K,\omega)}$ is established. If it is not established, then the manager rejects $\mathcal{C}$ and $\mathcal{E}$, and terminates the process;
4. The manager decrypts $\mathcal{C}$ with K using the AES decryption algorithm. Subsequently, the manager obtains the original data M and its signature S;
5. The manager verifies the integrity of M via a verification procedure using the signature S. If the verification process is completed successfully, then data retrieval is terminated.

4.3.5. Data Sharing to a Third Party

The EHR may be requested by a third party, such as other healthcare institutes, pharmaceutical companies, and insurance institutes. During this process, re-encryption is performed because the encrypted symmetric key should be delivered to the third party. The processes involved are as follows:

1. When the server receives a request to access the stored EHR from the third party usr_t, the server verifies usr_t's accessibility from the blockchain. If usr_t does not have the right to access the EHR, then the server rejects usr_t;
2. The server obtains the requested pair of the encrypted data $\mathcal{C}$ and symmetric key $\mathcal{E} = (D,\ E,\ F,\ s)$, and sends $\mathcal{C}$ to the third party and $\mathcal{E}$ to the proxy server;
3. When the proxy server receives $\mathcal{E}$, it finds $\mathtt{rk}_{M\to t} = (rk_{M\to t}^{<1>},\ V,\ W)$ from its storage;
4. The proxy server calculates $E' = E^{rk_{M\to t}^{<1>}}$ and sends $(E',\ F,\ V,\ W)$ to the third party.

After this process, the third party receives the ciphertext of the EHR and the symmetric key that is encrypted with the first level of the PRE.

4.3.6. Data Retrieval by a Third Party

After the third party receives the requested pair $(\mathcal{C},\ \mathcal{E})$ via the process presented in Section 4.3.5, $\mathcal{E}$ is first decrypted to obtain the symmetric key K with secret key $\mathtt{sk}_t = (x_{t,1},\ x_{t,2})$. Subsequently, the third party decrypts $\mathcal{C}$ with K, and verifies the integrity using the verification algorithm of the digital signature. The processes involved are as follows:

1. The third party calculates $W \oplus H_2(V^{\frac{1}{x_{t,2}}})$. The following result can be obtained if no problem arises:

$$W \oplus H_2(V^{\frac{1}{x_{t,2}}}) = H_2(g^v) \oplus (h\|\pi) \oplus H_2(g^{x_{t,2}\cdot v\cdot\frac{1}{x_{t,2}}}) = (h\|\pi).$$

2. Using the result of Step 1, the third party derives K by calculating $F \oplus H_2(E'^{\frac{1}{h}})$. If no problem arises, the following result can be obtained:

$$\begin{aligned} F \oplus H_2(E'^{\frac{1}{h}}) &= H_2(g^r) \oplus (K\|\omega) \oplus H_2\left(\left(pk_{M,1}^{H_4(pk_{M,2})} \cdot pk_{M,2}\right)^{r\cdot rk_{M\to t}^{<1>}\cdot\frac{1}{h}}\right) \\ &= H_2(g^r) \oplus (K\|\omega) \oplus H_2\left(g^{(x_{M,1}\cdot H_4(pk_{M,2})+x_{M,2})\cdot\frac{r\cdot h}{\left(x_{M,1}\cdot H_4(pk_{M,2})+x_{M,2}\right)}\cdot\frac{1}{h}}\right) \\ &= H_2(g^r) \oplus (K\|\omega) \oplus H_2(g^r) = (K\|\omega); \end{aligned}$$

3. The third party verifies whether $V = pk_{t,2}^{H_1(h,\pi)}$ and $E' = g^{H(K,\omega)\cdot h}$ hold. If they do not hold, then the third party rejects the received data and requests them again from the server;
4. Using the decrypted symmetric key K, the third party decrypts $\mathcal{C}$ using the AES decryption scheme. Subsequently, $(M\|S)$ is obtained, where M is the plaintext of the requested EHR, and S is its signature;

5. Finally, the third party verifies the integrity of M via the process to verify the digital signature using S. If the verification is successful, then the third party accepts M and terminates the process.

4.4. Data Management Processes

In this subsection, we provide the processes for data creation and storage, data access by the user, and data access by the third party.

First, consider the case in which a new user, such as a health professional, general user, or third party participates in the system. When a new member participates in the proposed system, a private key should be created and a public key calculated for the PRE scheme, as mentioned in Section 4.3.2. In addition, to open the stored file, a re-encryption key that transforms the encrypted symmetric key to another encrypted key that can be decrypted by the user is generated, as mentioned in Section 4.3.5.

Next, we consider the case in which the data are created by the medical institute. For this case, the process described in Section 4.3.3 is executed. When a health professional, such as a doctor, creates an EHR, a symmetric key is created first, and then the data are encrypted with the key using a symmetric encryption scheme such as the AES. Subsequently, the symmetric key is encrypted using the second level of the PRE with the manager's public key. Next, a pair of data and encrypted keys are transferred to the data server, and the access authority for the data is recorded in the blockchain system.

When a manager wishes to view the stored file, he/she must first request the file server to send the corresponding file. Subsequently, the server verifies the requester's accessibility for the file using the blockchain and returns the encrypted file and encrypted key to the manager. Next, the manager retrieves the symmetric key using the second-level decryption, as mentioned in Section 4.3.4, and decrypts the file using the symmetric key. Finally, the manager obtains the data.

In addition to the manager, a general user, such as a patient or a third party, may require the stored data. Prior to this process, the third party should obtain the right to access the data and create a re-encryption key using the procedure detailed in Section 4.3.5. When the file server receives a data request, it verifies the accessibility of the blockchain server. If the verification is passed, then the file server sends the requested data to the proxy server, and the proxy re-encrypts the symmetric key. Finally, the encrypted data and the re-encrypted key are transmitted to the end user, and the user decrypts the key and data.

5. Discussion

In this section, we discuss the scheme proposed in Section 4. First, we analyze the security issues for managing and sharing data. Subsequently, we measure the operation times for various cases, such as encryption, re-encryption, and decryption, to demonstrate the availability.

5.1. Security Analysis

In this subsection, we analyze the proposed system with respect to security. When the EHR is created at the hospital, as shown in Figure 3, it is encrypted, and the key used for that encryption is encrypted with the manager's public key; hence, the path between the hospital and the server can be regarded as secure. Next, we discuss the case in which a patient and a third party require the EHR, where the privacy-sensitive record may be leaked to the outsourcing company.

First, we consider the case in which a patient or a third party requests an EHR to the server. Initially, when the EHR is stored on the server, information that allow access to the EHR is transmitted and saved in the blockchain. Moreover, the contents in the blockchain cannot be changed, and hence the access information cannot be modified. Therefore, the EHR is delivered to users who can access the record. In addition, owing to the proxy server, the encrypted symmetric key is transformed to one that can be decrypted by the

requested data without decryption; hence, the plaintext of the symmetric key or the EHR will not be exposed between the file server and the end user.

In the Introduction, we argued that the EHR can be leaked to the outsourcing company because the company may disregard the smart contract in the blockchain. To prevent this in our system, we adopted the PRE scheme; therefore, the symmetric key is not exposed to the outsourcing company because it can be decrypted by either the system manager or end user. Nonetheless, the system manager should be separated from the blockchain outsourcing company.

5.2. Experimental Results

In this subsection, we discuss the performance of the proposed system. Because the security level and system performance exhibit a tradeoff relationship, the operation costs for data storage and retrieval will increase when the proposed scheme is adopted. To solve this issue, we measured the performance time for each process and demonstrated that the processes would be performed within a reasonable duration. For the experiments, we use forced expiratory volume data from a spirometry device.

For the performance analysis, we used two computers with Intel(R) Core(TM) i5-8400 CPU @2.80 GHz, 8 GB of RAM, and Ubuntu Linux 18.04 LTS. These computers are used as proxy servers and client computers, respectively. In addition, pairing-based cryptography library 0.5.14 [48] and OpenSSL 1.1.1f library were used for PRE and AES cryptography, respectively. In this experimental analysis, we measured the operation times for the data storage and data retrieval processes.

First, we analyzed the results of the data storage process. This process involves a digital signature, data encryption using the AES, and symmetric key encryption using the PRE. For the digital signature, a digital signature algorithm is used. To operate AES encryption, we selected AES256. The sizes of the EHRs ranged from 128 kB to 64 MB, and 100 tests were conducted for each data size. The average time required for each data size is shown in the blue graph of Figure 6.

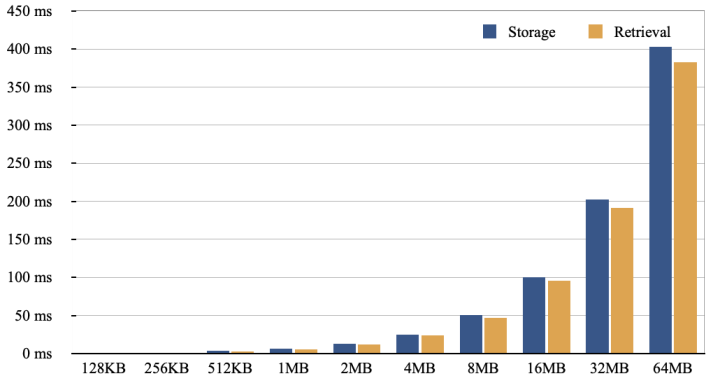


Figure 6. Operation times for data storage and data retrieval.

Next, we discuss the time required for data retrieval. This process involves symmetric key decryption using PRE, AES decryption, and the signature verification procedure. The experimental environment is the same as that for data storing. Moreover, to measure the operation time, we conducted 100 tests for each data size and calculated the average value. The experimental results for data retrieval are shown in the orange graph of Figure 6.

As shown by the graphs in Figure 6, we can confirm that the operation times for saving or loading the EHR are less than 1 s if the size of the record is not greater than 64 MB. Consequently, we conclude that, except for a case involving a large EHR, data storage and retrieval can be completed within a reasonable duration. In addition, we can verify whether the operation time is proportional to the EHR size, such that the computation costs for the PRE used for encrypting and decrypting the symmetric key will not affect the

entire operation duration, even though the PRE is a relatively heavy cryptography scheme compared with other security schemes.

6. Conclusions and Future Works

Herein, we propose a blockchain-based EHR management scheme with PRE. Owing to the security of the blockchain, the stored EHRs can be delivered only to authorized users, such that the integrity of the records is guaranteed. In addition, PRE ensures the confidentiality of the encrypted EHRs. In our system, the proxy server is separated from the blockchain system; therefore, the plaintexts of the EHRs will not be visible to the blockchain outsourcing company if the system manager does not reveal a user's private key. Based on security analysis and the experimental results, we confirmed the security and usability of the proposed scheme. Using the proposed scheme, privacy-sensitive information in the EHRs will be maintained by only trustable members; therefore, the security and privacy of the healthcare data are guaranteed.

A few shortcomings exist in our study. For example, when an authorized user obtains the EHR from the file server through the blockchain system and the proxy server, the user can own the record indefinitely, which may result in another privacy issue. If the user sends the obtained EHR to another person that is not granted access, then the privacy of the EHR owner will be invaded. To overcome this issue, additional privacy-preserving schemes are required. In future studies, we will research and develop a scheme that solves privacy and sharing problems simultaneously.

Author Contributions: Data curation, Y.K. and S.-O.L.; Project administration, Y.-H.P.; Writing—original draft, Y.-H.P.; Writing—review & editing, K.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was financially supported by the Ministry of Trade, Industry & Energy (MOTIE), the Korea Institute for Advancement of Technology (KIAT) through the Encouragement Program for The Industries of Economic Cooperation Region (P0014681), and by Seoul R&BD Program (CT200018, Seoul Campus Town Technology R&D Project).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Chen, H.Y.; Wu, Z.Y.; Chen, T.L.; Huang, Y.M.; Liu, C.H. Security Privacy and Policy for Cryptographic Based Electronic Medical Information System. *Sensors* **2021**, *21*, 713. [CrossRef] [PubMed]
2. Sharma, S.; Chen, K.; Sheth, A. Toward Practical Privacy-Preserving Analytics for IoT and Cloud-Based Healthcare Systems. *IEEE Internet Comput.* **2018**, *22*, 42–51. [CrossRef]
3. Abbas, A.; Khan, S.U. A Review on the State-of-the-Art Privacy-Preserving Approaches in the e-Health Clouds. *IEEE J. Biomed. Health Inform.* **2014**, *18*, 1431–1441. [CrossRef] [PubMed]
4. Jin, H.; Luo, Y.; Li, P.; Mathew, J. A Review of Secure and Privacy-Preserving Medical Data Sharing. *IEEE Access* **2019**, *7*, 61656–61669. [CrossRef]
5. Zyskind, G.; Nathan, O.; Pentland, A.S. Decentralizing Privacy: Using Blockchain to Protect Personal Data. In Proceedings of the 2015 IEEE Security and Privacy Workshops, San Jose, CA, USA, 21–22 May 2015; pp. 180–184. [CrossRef]
6. Zhang, J.; Sun, J.; Yang, Y.; Liang, C.; Yao, Y.; Cai, W.; Jin, J.; Zhang, G.; Sun, K. Image-Based Electronic Patient Records for Secured Collaborative Medical Applications. In Proceedings of the 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference, Shanghai, China, 17–18 January 2006; pp. 3218–3220. [CrossRef]
7. Lee, J.P.; Kim, Y.H.; Lee, J.K. SSL Application for Managed Security between the Mobile and HIS Biometric Information Collection Client. In Proceedings of the 2014 28th International Conference on Advanced Information Networking and Applications Workshops, Victoria, BC, Canada, 13–16 May 2014; pp. 55–60. [CrossRef]
8. Park, Y.H.; Je, D.H.; Park, M.H.; Seo, S.W. Efficient Rekeying Framework for Secure Multicast with Diverse-Subscription-Period Mobile Users. *IEEE Trans. Mob. Comput.* **2014**, *13*, 783–796. [CrossRef]

9. Korzun, D.G. Internet of Things Meets Mobile Health Systems in Smart Spaces: An Overview. In *Internet of Things and Big Data Technologies for Next Generation Healthcare*; Bhatt, C., Dey, N., Ashour, A.S., Eds.; Springer International Publishing: Cham, Switzerland, 2017; pp. 111–129. [CrossRef]
10. Korzun, D.; Meigal, A. Multi-Source Data Sensing in Mobile Personalized Healthcare Systems: Semantic Linking and Data Mining. In Proceedings of the 2019 24th Conference of Open Innovations Association (FRUCT), Moscow, Russia, 8–12 April 2019; pp. 187–192. [CrossRef]
11. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. Technical Report. Available online: https://bitcoin.org/bitcoin.pdf (accessed on 9 October 2021).
12. Wood, G. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum Proj. Yellow Pap.* **2014**, *151*, 1–32.
13. Piedrahita Castillo, D.; Regidor, F.M.; Higuera, J.B.; Higuera, J.R.B.; Montalvo, J.A.S. A New Mail System for Secure Data Transmission in Cyber Physical Systems. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* **2020**, *28*, 23–48. [CrossRef]
14. Sharma, A.; Sarishma; Tomar, R.; Chilamkurti, N.; Kim, B.G. Blockchain Based Smart Contracts for Internet of Medical Things in e-Healthcare. *Electronics* **2020**, *9*, 1609. [CrossRef]
15. Agbo, C.C.; Mahmoud, Q.H.; Eklund, J.M. Blockchain Technology in Healthcare: A Systematic Review. *Healthcare* **2019**, *7*, 56. [CrossRef]
16. 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]
17. 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]
18. Yang, H.; Yang, B. A Blockchain-Based Approach to the Secure Sharing of Healthcare Data. In Proceedings of the Norwegian Information Security Conference 2017 (NISK2017), Oslo, Norway, 27–29 November 2017; pp. 100–111.
19. Xia, Q.; Sifah, E.B.; Smahi, A.; Amofa, S.; Zhang, X. BBDS: Blockchain-Based Data Sharing for Electronic Medical Records in Cloud Environments. *Information* **2017**, *8*, 44. [CrossRef]
20. Li, M.; Yu, S.; Zheng, Y.; Ren, K.; Lou, W. Scalable and Secure Sharing of Personal Health Records in Cloud Computing Using Attribute-Based Encryption. *IEEE Trans. Parallel Distrib. Syst.* **2013**, *24*, 131–143. [CrossRef]
21. Liu, J.; Huang, X.; Liu, J.K. Secure sharing of Personal Health Records in cloud computing: Ciphertext-Policy Attribute-Based Signcryption. *Future Gener. Comput. Syst.* **2015**, *52*, 67–76. [CrossRef]
22. Li, F.; Liu, K.; Zhang, L.; Huang, S.; Wu, Q. EHRChain: A Blockchain-based EHR System Using Attribute-Based and Homomorphic Cryptosystem. *IEEE Trans. Serv. Comput.* **2021**, 1. [CrossRef]
23. Fu, J.; Wang, N.; Cai, Y. Privacy-Preserving in Healthcare Blockchain Systems Based on Lightweight Message Sharing. *Sensors* **2020**, *20*, 1898. [CrossRef]
24. Hao, K.; Xin, J.; Wang, Z.; Cao, K.; Wang, G. Blockchain-Based Outsourced Storage Schema in Untrusted Environment. *IEEE Access* **2019**, *7*, 122707–122721. [CrossRef]
25. Yang, Y.; Zheng, X.; Guo, W.; Liu, X.; Chang, V. Privacy-preserving smart IoT-based healthcare big data storage and self-adaptive access control system. *Inf. Sci.* **2019**, *479*, 567–592. [CrossRef]
26. Blaze, M.; Bleumer, G.; Strauss, M. Divertible protocols and atomic proxy cryptography. In *Advances in Cryptology—EUROCRYPT'98, Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques Espoo, Finland, 31 May–4 June 1998*; Nyberg, K., Ed.; Springer: Berlin/Heidelberg, Germany, 1998; pp. 127–144.
27. Lin, H.Y.; Jiang, Y.R. A Multi-User Ciphertext Policy Attribute-Based Encryption Scheme with Keyword Search for Medical Cloud System. *Appl. Sci.* **2021**, *11*, 63. [CrossRef]
28. Chow, S.S.M.; Weng, J.; Yang, Y.; Deng, R.H. Efficient Unidirectional Proxy Re-Encryption. In *Progress in Cryptology—AFRICACRYPT 2010, Proceedings of the International Conference on Cryptology in Africa, Stellenbosch, South Africa, 3–6 May 2010*; Bernstein, D.J., Lange, T., Eds.; Springer: Berlin/Heidelberg, Germany, 2010; pp. 316–332.
29. Lee, S.O.; Ko, K.; Park, Y.H. Blockchain-based Medical Data Management System Using Proxy Re-encryption. In Proceedings of the International Conference on Green and Human Information Technology 2021 (ICGHIT 2021), Jeju Island, Korea, 13–15 January 2021.
30. Chelladurai, M.U.; Pandian, D.S.; Ramasamy, D.K. A blockchain based patient centric electronic health record storage and integrity management for e-Health systems. *Health Policy Technol.* **2021**, 100513. [CrossRef]
31. Dubovitskaya, A.; Xu, Z.; Ryu, S.; Schumacher, M.; Wang, F. Secure and trustable electronic medical records sharing using blockchain. *AMIA Annu. Symp. Proc.* **2017**, *2017*, 650.
32. Yang, X.; Li, T.; Pei, X.; Wen, L.; Wang, C. Medical Data Sharing Scheme Based on Attribute Cryptosystem and Blockchain Technology. *IEEE Access* **2020**, *8*, 45468–45476. [CrossRef]
33. Wang, Y.; Zhang, A.; Zhang, P.; Wang, H. Cloud-assisted EHR sharing with security and privacy preservation via consortium blockchain. *IEEE Access* **2019**, *7*, 136704–136719. [CrossRef]
34. Yang, M.; Margheri, A.; Hu, R.; Sassone, V. Differentially private data sharing in a cloud federation with blockchain. *IEEE Cloud Comput.* **2018**, *5*, 69–79. [CrossRef]
35. Liu, X.; Wang, Z.; Jin, C.; Li, F.; Li, G. A blockchain-based medical data sharing and protection scheme. *IEEE Access* **2019**, *7*, 118943–118953. [CrossRef]

36. Thwin, T.T.; Vasupongayya, S. Blockchain based secret-data sharing model for personal health record system. In Proceedings of the 2018 5th International Conference on Advanced Informatics: Concept Theory and Applications (ICAICTA), Krabi, Thailand, 14–17 August 2018; pp. 196–201.
37. Thwin, T.T.; Vasupongayya, S. Blockchain-based access control model to preserve privacy for personal health record systems. *Secur. Commun. Netw.* **2019**, *2019*, 8315614. [CrossRef]
38. 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]
39. Gao, J.; Asamoah, K.O.; Sifah, E.B.; Smahi, A.; Xia, Q.; Xia, H.; Zhang, X.; Dong, G. Gridmonitoring: Secured sovereign blockchain based monitoring on smart grid. *IEEE Access* **2018**, *6*, 9917–9925. [CrossRef]
40. Xia, Q.; Sifah, E.B.; Huang, K.; Chen, R.; Du, X.; Gao, J. Secure Payment Routing Protocol for Economic Systems Based on Blockchain. In Proceedings of the 2018 International Conference on Computing, Networking and Communications (ICNC), Maui, HI, USA, 5–8 March 2018; pp. 177–181.
41. Lin, I.C.; Liao, T.C. A survey of blockchain security issues and challenges. *Int. J. Netw. Secur.* **2017**, *19*, 653–659.
42. Zhang, Y.; Xu, C.; Lin, X.; Shen, X.S. Blockchain-based public integrity verification for cloud storage against procrastinating auditors. *IEEE Trans. Cloud Comput.* **2019**, *9*, 923–937. [CrossRef]
43. Mohanta, B.K.; Panda, S.S.; Jena, D. An overview of smart contract and use cases in blockchain technology. In Proceedings of the 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Bengaluru, India, 10–12 July 2018; pp. 1–4.
44. Bahga, A.; Madisetti, V.K. Blockchain platform for industrial internet of things. *J. Softw. Eng. Appl.* **2016**, *9*, 533–546. [CrossRef]
45. Mambo, M.; Usuda, K.; Okamoto, E. Proxy Signatures for Delegating Signing Operation. In Proceedings of the 3rd ACM Conference on Computer and Communications Security, CCS '96, New Delhi, India, 14–16 March 1996; Association for Computing Machinery: New York, NY, USA, 1996; pp. 48–57. [CrossRef]
46. Zhou, L.; Marsh, M.; Schneider, F.; Redz, A. Distributed Blinding for Distributed ElGamal Re-Encryption. In Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS'05), Columbus, OH, USA, 6–10 June 2005; p. 824. [CrossRef]
47. Canetti, R.; Hohenberger, S. Chosen-Ciphertext Secure Proxy Re-Encryption. In Proceedings of the 14th ACM Conference on Computer and Communications Security, CCS '07, Alexandria, VA, USA, 29 October–2 November 2014; Association for Computing Machinery: New York, NY, USA, 2007; pp. 185–194. [CrossRef]
48. Applied Cryptography Group PBC Library—Pairing-Based Cryptography. Available online: https://crypto.stanford.edu/pbc/ (accessed on 9 October 2021).