Original Article

Healthc Inform Res. 2020 January;26(1):3-12.
https://doi.org/10.4258/hir.2020.26.1.3
pISSN 2093-3681 • eISSN 2093-369X

HIR
Healthcare Informatics Research

Application of Blockchain to Maintaining Patient Records in Electronic Health Record for Enhanced Privacy, Scalability, and Availability

Dara Tith, Joong-Sun Lee, Hiroyuki Suzuki, W. M. A. B. Wijesundara, Naoko Taira, Takashi Obi, Nagaaki Ohyama

Institute of Innovative Research, Tokyo Institute of Technology, Yokohama, Japan

Objectives: Electronic Health Record (EHR) systems are increasingly used as an effective method to share patients' records among different hospitals. However, it is still a challenge to access scattered patient data through multiple EHRs. Our goal is to build a system to access patient records easily among EHRs without relying on a centralized supervisory system. **Methods:** We apply consortium blockchain to compose a distributed system using Hyperledger Fabric incorporating existent EHRs. Peer nodes hold the same ledger on which the address of a patient record in an EHR is written. Individual patients are identified by unique certificates issued by a local certificate authorities that collaborate with each other in a channel of the network. To protect a patient's privacy, we use a proxy re-encryption scheme when the data are transferred. We designed and implemented various chaincodes to handle business logic agreed by member organizations of the network. **Results:** We developed a prototype system to implement our concept and tested its performance including chaincode logic. The results demonstrated that our system can be used by doctors to find patient's records and verify patient's consent on access to the data. Patients also can seamlessly receive their past records from other hospitals. The access log is stored transparently and immutably in the ledger that is used for auditing purpose. **Conclusions:** Our system is feasible and flexible with scalability and availability in adapting to existing EHRs for strengthening security and privacy in managing patient records. Our research is expected to provide an effective method to integrate dispersed patient records among medical institutions.

Keywords: Health Information Exchanges, Electronic Health Records, Patient Data Privacy, Computer Security, Decentralization

Submitted: September 13, 2019
Revised: November 8, 2019
Accepted: November 29, 2019

Corresponding Author
Joong-Sun Lee
R2-60 Tokyo Institute of Technology, 4259 Nagatsuta, Midori-ku, Yokohama, Kanagawa 226-8503, Japan. Tel: +81-0459245482, E-mail: j-lee@isl.titech.ac.jp (https://orcid.org/0000-0002-6976-6472)

I. Introduction

Electronic Health Record (EHR) systems [1] have been increasingly used as an effective method to share patients' records among different hospitals. However, it is still a challenge to access scattered patient data through multiple EHRs because existing EHRs are regionally limited or belong to affiliated hospitals. Based on the report published by the Office of the National Coordinator for Health Information Technology (ONC) [2], the main barrier to access patient records lies in the difficulty to find provider's addresses. So far, there have been several projects to overcome these problems; however, the solutions they have produced are difficult and

involve redesigning or upgrading of existing EHR systems, which would require substantial expenses. Among them, one of the most actively ongoing programs is run by CommonWell Health Alliance [3] in the United States, a nonprofit association. They support EHRs, care providers, and healthcare information technology (HIT) vendors to connect to their nationwide interoperability network via certified integration platforms and intermediaries. They use a centralized system that allows patients and doctors to search for a patient's scattered medical records [4]. Such a centralized architecture has some drawbacks that it may face the risk of single-point-of-failure and bottleneck of data flow when the system becomes larger.

In an EHR system, when patient records are accessed for some reason, the history of all such events must be recorded in a log file for later audit on access histories. The log file is used for reconstructing the past state of medical records, and it can be represented as a legal document [5-7]. Thus, we should firmly protect the log file from illegal access and make it immutable if possible.

In this paper, we propose a decentralized system to address problems in sharing patient records among EHRs without relying on a high-end centralized system. Our system has three major features: (1) a trusted directory of patient data in EHRs which guarantees access as well as the integrity of the data itself, (2) strengthened security in dealing with patient data by utilizing a particular encryption scheme and providing a transparent and undeniable audit trail based on an immutable access log, and (3) providing scalability to cover multiple existing EHRs of regional or core hospitals with the least modification and availability of the system without relying on a centralized supervisory system.

We design the system following the Health Insurance Portability and Accountability Act (HIPAA) technical safeguard [8] and ISO/TS 18308 [5] for the interoperability, data integrity, auditability, and availability of the system. To accomplish our goals, we adopt blockchain technology, especially the permissioned consortium type [9], using the Hyperledger Fabric (HLF) platform. Multiple hospitals gather to form a consortium having a private peer-to-peer network, and permission to join it is determined based on consensus among the members.

HLF is an open-source platform that has many essential components available in some programming languages. In addition, it provides the Byzantine fault tolerant consensus protocol [10] for ordering transactions to a block. Moreover, it allows end-to-end [11] throughput of more than 3,500 transactions per second. It is a project [12] hosted by the

Table 1. Components of the Hyperledger Fabric

	Description
Ledger	It consists of a blockchain and state database [12] (a.k.a world state). The former is a transaction log, while the latter holds current values of ledger states. Due to the state database, the program readily obtains values without traversing the entire transaction log. Transactions [13-15] are collected to form a block that is appended sequentially to the last block of the blockchain, which is immutable once it is made.
User roles	There are three main types of user roles: client, peer (endorsing and committing one), and orderer. A peer is a network node, and endorsing peers, simply called endorsers, conduct endorsement with simulating a client's transaction proposal. The proposal is a tentative transaction before being accepted into new block of the ledger. An orderer runs an ordering service for creating a new block with transactions and then broadcasts the block to all peers. A committing peer, also called a committer, updates the ledger by appending the new block to it and revising the state database with the write-sets of valid transactions.
Chaincode	It is an application program run by peers to facilitate, verify, or enforce negotiation and agreement between users. A chaincode is otherwise known as a smart contract in other blockchain platforms like Ethereum. A chaincode [16] has many programming functions in it, and it usually reads and updates the ledger state with all the business logic contained inside functions.
Membership service provider (MSP)	MSP [17] aims to abstract all cryptographic mechanisms and protocols behind issuing and validating certificates, and user authentication.There are two types of MSP, channel and local. A channel MSP provides a method to validate enrolment certificates (ECerts) among different organizations in the channel, while a local MSP offers a method to verify a user's identity in one organization. Thus, each organization has their local MSP having unique a MSP ID, and they issue ECerts, X.509 certificates, to all the local participants with enrolment IDs (eID) through their certificate authority (CA).

 https://doi.org/10.4258/hir.2020.26.1.3

Linux Foundation, and contributions to the project are made by Digital Asset and IBM.

II. Methods

1. Hyperledger Fabric

In HLF, there are several key components (Table 1) that play pivotal roles in the system. In addition, it provides three phases of consensus (Table 2) to validate transactions before uploading them to the ledger. HLF provides a variety of special designated chaincodes called system chaincodes to perform certain privileged tasks. Examples of system chaincodes are Configuration, Life Cycle, Query, Endorser, and Validator system chaincodes. In our study, we designed several prerequisite chaincodes and implemented them in our prototype system.

2. System Conceptual Design

We built a private subnet of an HLF network where the same ledger is shared among the hospital members (Figure 1), which is called a channel. Organizations or departments within them can constitute independent channels with relevant ledgers according to their needs. In practice, medical data is usually too big to handle directly in a ledger; therefore, data is kept in an EHR, and only the address is recorded in the ledger. Such storage type is called on-chain or off-chain according to whether the data is in a ledger or not [15]. A ledger also contains the hash values of data. This guarantees data integrity because once a piece of data is written in a ledger, it becomes immutable, and this allows the user to check whether the data has been altered or not.

In our system, we assume that a client of HLF (Table 1) is a doctor, nurse, or clerk who helps patients to upload or share their medical records. Clients from medical institutions issue various types of transactions and store them in a ledger. The ledger consists of patient metadata, including demographics, and these data are used for retrieval requests to find transactions related to a specific patient during a specified period of timestamps of blocks in the ledger. Thus, the ledger functions as a registry of patient IDs for doctors to search for their patient's records stored in other EHRs. In addition, each transaction contains the client's request metadata, chaincode execution results, and medical record metadata, such as hospital ID, hash of medical records stored in an EHR, and so forth. In consequence, these data will be used for auditing purpose.

For an individual patient, the enrolment ID (eID) issued by a membership service provider (MSP) is used as the channel patient ID in the system. Each transaction in the ledger contains an eID, which is hashed after being concatenated with a random data so called salt [19] in the format as shown below:

$n $salt $hash (salt + eID).

This format is nearly the same as how the Linux system stores its user's hashed passwords with salts. Here, "$" is used as a delimiter between neighboring fields; "n" represents

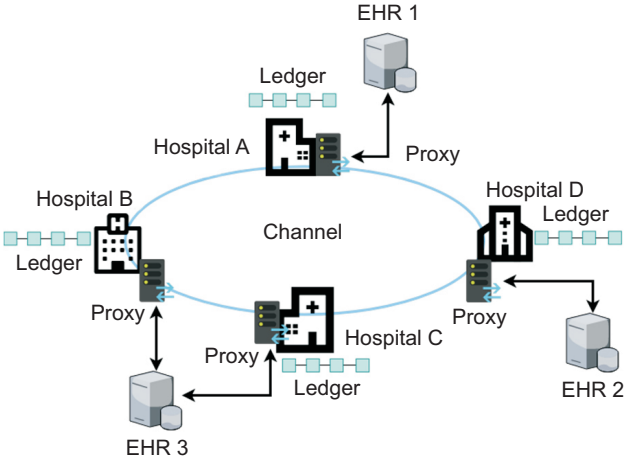

Figure 1. Channel of network among medical institutions with the same ledger. EHR: Electronic Health Record.

Table 2. Three phases of consensus to validate transactions in the Hyperledger Fabric

	Description
Endorsement	Endorsers simulate a client's transaction proposal and provide it with their digital signatures after validating the proposal format and executing the request chaincode [18] successfully.
Ordering	An orderer orders transaction proposals as transactions from different clients to create a new block. It signs the block with its digital signature and then broadcasts it to all peers, both endorsing and committing peers on a channel.
Validation & commit	After receiving a new block from the orderer, peers check whether it meets the requirements of the endorsement policy and then validate it by confirming the data integrity compared with the simulation result in the endorsement phase. If no mismatch is found, a committer appends the new block to the ledger, and updates the state database with the write-sets of the transactions.

hash algorithm type; and 1, 5, and 6 correspond to MD5, SHA-256, and SHA-512, respectively. Salt is a string of random alphanumeric characters up to 16 letters.

3. Cryptographic Scheme

Before patient data is uploaded to the EHR system with the patient's consent, the data is encrypted using an adequate symmetric key. Then the symmetric key is asymmetrically encrypted using the patient's public key and attached to the encrypted data. This hybrid encryption makes the procedure efficient in terms of both speed and convenience because the encryption of large data can be done faster by symmetric-key than asymmetric-key, while the latter is more convenient in the encryption of small-size cryptographic key.

To read patient data, a proxy downloads it from the relevant EHR and sends it to the receiver. However, in case the receiver is different from the patient, the encrypted symmetric key at the data should be transformed, so that it can be decrypted by the receiver's private key. To do this, we use a proxy re-encryption scheme (Figure 2) in which the patient generates the proxy re-encryption key by mathematically combining their private key and the receiver's public key using the AFGH algorithm [20,21]. After receiving the newly made re-encryption key, the proxy re-encrypts the symmetric key for the receiver. In that process, the symmetric key is not disclosed to the proxy. Otherwise, the proxy must send the data to the patient to make it encrypted using the receiver's public key.

4. Web-Based Application

Our system provides web-based application for clients in each hospital to make access requests to the ledger or EHR. Web-based application is the front-end side application program available in a hospital or clinic. A hospital can have a single peer or many peers according to their scale, while a small clinic functions as a client without peer. For identifying participants across the system, doctors in each hospital are assumed to have their ECerts.

Web-based application offers web-based user interfaces and essential interactive functions in communication between participants in the system. Patients use it to generate key pairs to register and enrol their identities to the system

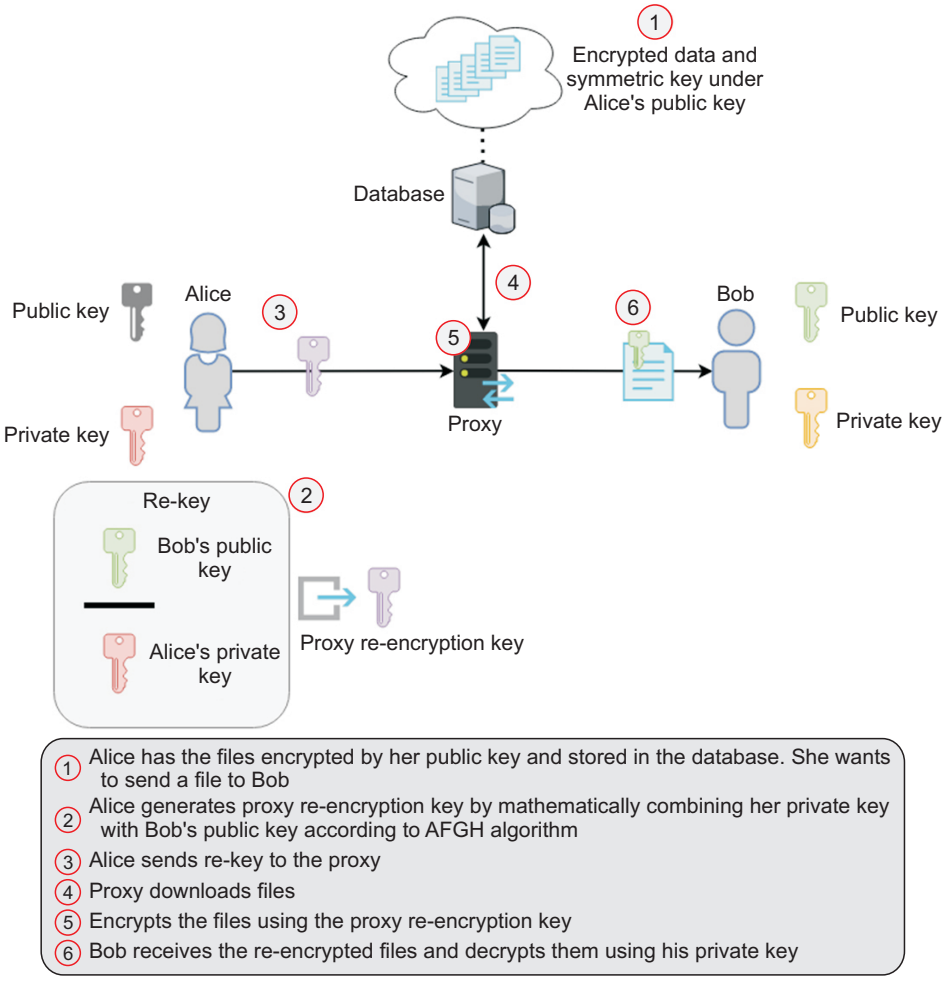

Figure 2. Proxy re-encryption scheme.

https://doi.org/10.4258/hir.2020.26.1.3

to obtain ECerts. In addition, they can generate proxy re-encryption keys and send them to the proxy. On the other hand, the client uses this web-based application to create a transaction proposal and submit it to the blockchain system for the tasks such as identifying a patient's identity and creating, uploading, and sharing medical records, metadata and so forth.

III. Results

1. Developed Chaincodes

In our prototype system, we installed five chaincodes with which business logics are performed. Each chaincode has many programming functions in it, and they usually read and update the ledger state with all the business logic contained inside functions. In an actual system, each chaincode needs to get agreement among all the member hospitals before being deployed in the system. Table 3 presents details of the proposed chaincodes.

2. Use Case Scenarios

We simulated use cases using the prototype system. In Figures 3–5, which describe a practical situation, we assume that a patient, let's call her Alice, visits Hospital_A for the first time. There, Alice is diagnosed with cancer, and her doctor, Dr. Bob, recommends her to go to the central hospital to see a cancer specialist. Dr. Bob uploads Alice's records with her consent to the hospital's EHR. Then Alice moves to the central hospital, and the cancer specialist accesses Alice's data in the EHR that belongs to Hospital_A.

1) First visit to a hospital

Alice makes a first visit to Hospital_A (Figure 3). To enrol in the hospital, she provides her demographic information or the national insurance number to a clerk. This information will be used for registering her in the patient identity source of the hospital and issuing an ECert for her. The ECert and private key need to be stored in a secure storage device, for instance, an IC card or USB memory. After issuing the ECert by local certificate authority (CA), the clerk must store the hash value of Alice's eID and individual patient ID in the ledger.

Table 3. Description of chaincodes installed in the prototype system

Chaincodes	Description
Record manager chaincode	This is the core chaincode of the system, which is involved in other chaincodes' execution, to simulate transaction proposals for validation and endorsement of a proposal. This chaincode helps a client in preparing, uploading, and sharing a patient's records.
Patient identity chaincode	This is called by clients to register and query a patient's identity from the ledger. Patients can find a list of identity transactions containing their previous hospital visits. In addition, if patients lose their ECerts, they can provide identifiable attributes to clients for searching and recovering them. Hash values of eIDs and demographics can be stored in the ledger for identifying patients. Since patients would be given different patient IDs from the hospitals they visited, this type of chaincode also stores and makes queries for patient IDs based on eIDs.
Permission manager chaincode	This works to authorize a third party's access to patient records based on patient consent. Patient consent contains a list of eIDs who are permitted to access, or conditions of comprehensive prior consent, which a patient puts into a transaction as metadata when the data is recorded in the ledger. For instance, a patient can share a specific part of his or her records with an insurance company who is also a participant in the network by putting its eID in the transaction.
Personal folder chaincode	This helps doctors collect all of a patient's transactions. It provides special query functions for searching for the transactions based on multiple keywords, such as the hash value of an eID with Salt, hospital ID, or timestamp.
Audition chaincode	This is for designated peers to audit the access histories of patient records by analyzing the access log in the ledger. Thus, patients can realize how their data has traversed among medical institutions and monitor whether each data transfer was made adequately in compliance with their consent. This chaincode can also produce statistics based on doctors' activities, timestamps of transactions, and patient metadata with demographics.

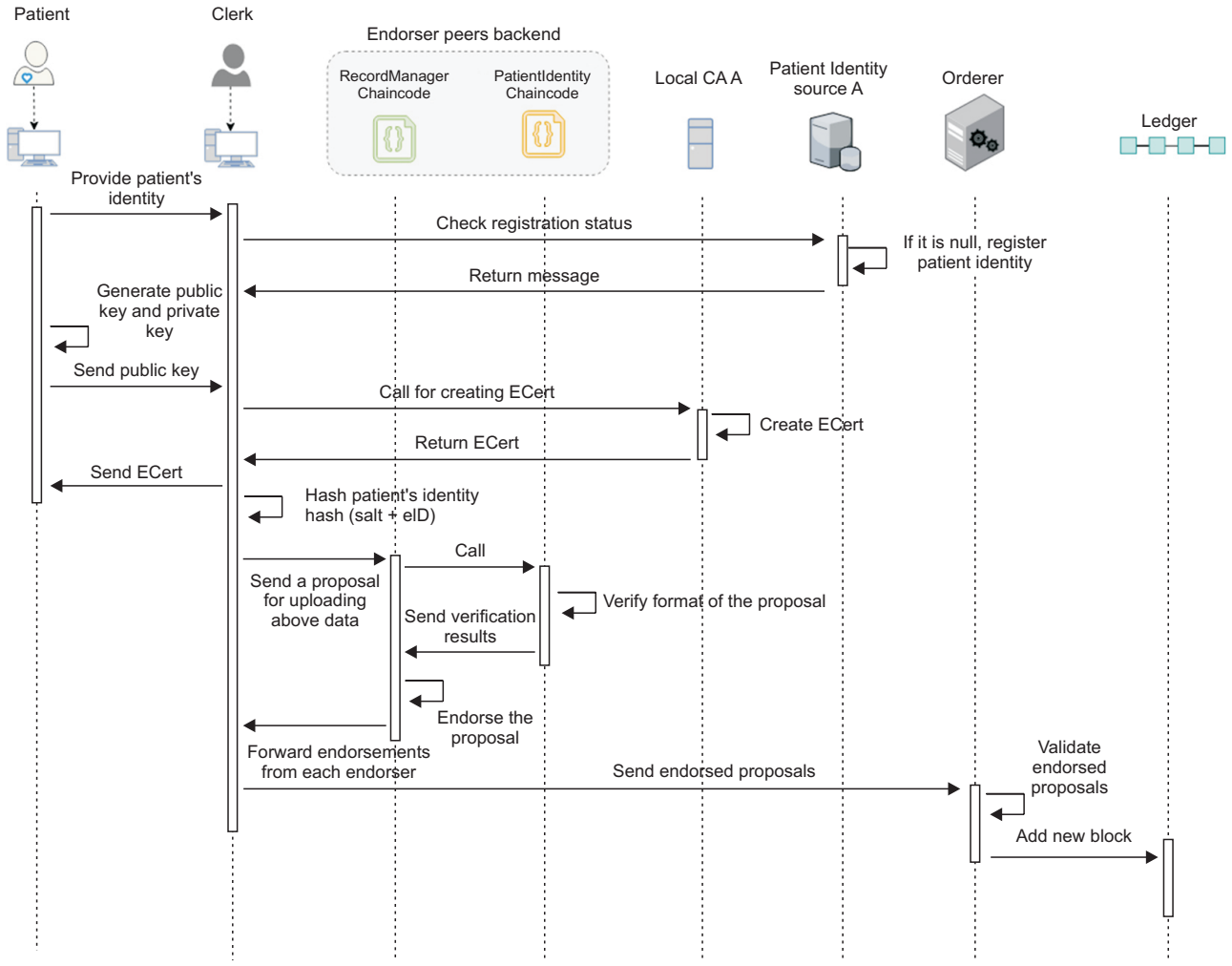

Figure 3. First visit to a hospital.

2) Uploading patient's record with metadata and consent

When a patient's records are uploaded to the EHR system (Figure 4), Alice provides the doctor her consent with conditions for sharing her records with other third parties or her relatives. Then, the doctor encrypts Alice's record using an adequate symmetric key and encrypts the key this time using Alice's public key to attach it with the record. Finally, the doctor uploads Alice's record to Hospital_A's EHR system and writes the record's consent and the address of the data location to the ledger.

3) Requesting patient's record

Alice goes to see a specialist in the central hospital (Figure 5), where she registers as a new patient, if needed, and provides her ECert previously issued in Hospital_A. When treating Alice, the doctor wants to get Alice's previous records, so he sends a transaction proposal of a request to obtain Alice's records metadata during a certain period and the previous hospital's ID. Then, each endorsing peer simulates the transaction proposal executing chaincodes and returns each result of the chaincode to the proxy of the hospital where the client application is run by the doctor. The application compares the query results, and if they are all matched, it lets the doctor select the necessary records from them to make a list of the patient's records that he wants to obtain. After receiving the list, the proxy asks Alice to generate the proxy re-encryption key. Then, the proxy downloads Alice's records in the list from relevant EHRs and re-encrypts every encrypted symmetric key at each record using the re-encryption. After that, the proxy sends Alice's records to the doctor.

3. Prototype System

A prototype system was built on a small scale for testing on a local network with four Window PCs for patients to use the patient web application, four Linux PCs for doctors to use the doctor web application, and four proxies for four hospitals. In addition, we used two Window PCs as EHRs. The HLF platform was run on Docker for executing chaincodes.

https://doi.org/10.4258/hir.2020.26.1.3

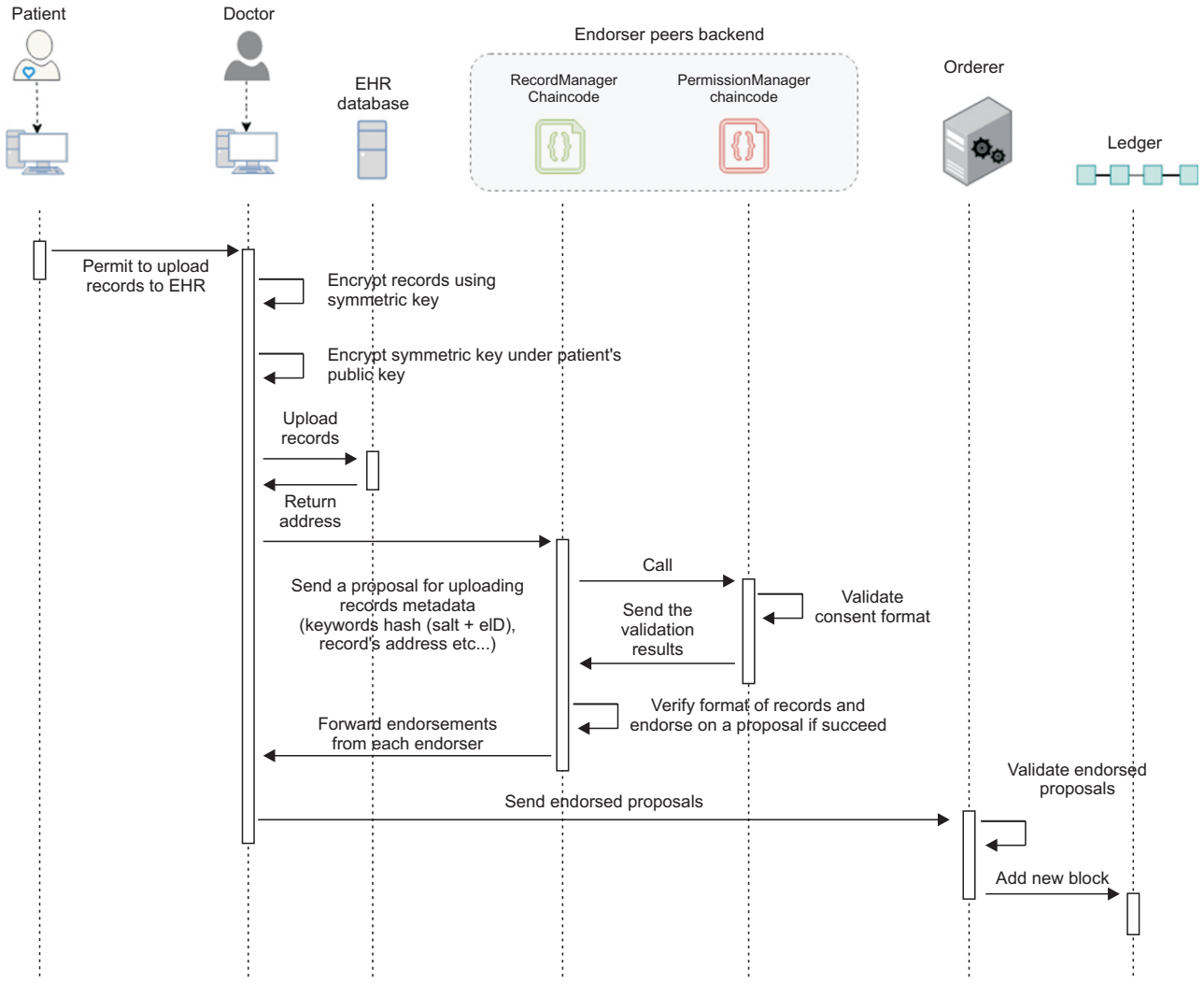

Figure 4. Uploading record with metadata and consent access. EHR: Electronic Health Record.

For EHR records, we dealt with standardized data, such as HL7/CDA and DICOM image data. We changed the system configuration with various numbers of PCs to assess the performance including chaincode logic. As a result, it took a little more time with an increasing number of PCs when querying data in a blockchain as well as encrypting and decrypting the records and transferring files.

The above prototype is not the same as an actual working environment. The system and chaincode functionality may require specific modification to suit consortium privacy policies and the legal requirements set by the governing authority.

IV. Discussion

In implementation of the system, all the verification steps are essential for security purposes. To protect patient privacy, we adopted the Advanced Encryption Standard (AES) algorithm for symmetric-key encryption of patient data and the Elliptic Curve ElGamal (EC-ElGamal) algorithm for asymmetric-key encryption of the symmetric key in the proxy re-encryption scheme. The asymmetric-key pair is also used for the signature on the transaction proposal. However, for the purpose of further strengthening security, a patient can have another key pair for a signature different from the one of the encryptions. The former is generated by using the HLF function, the latter by importing a function of EC-ElGamal encryption using EC cryptography. When a patient chooses to have two pairs of keys, he or she bears a greater burden to keep them secret. In the case that a patient loses these private key, a key escrow system is assumed to be used for retrieving the lost keys or symmetric keys from the ECert issuer or the hospital only for decryption of the patient data. After all, retrieved keys must be used temporarily before new keys and a new ECert are issued for the patient.

We hash eID with salt to avoid transactions of the records

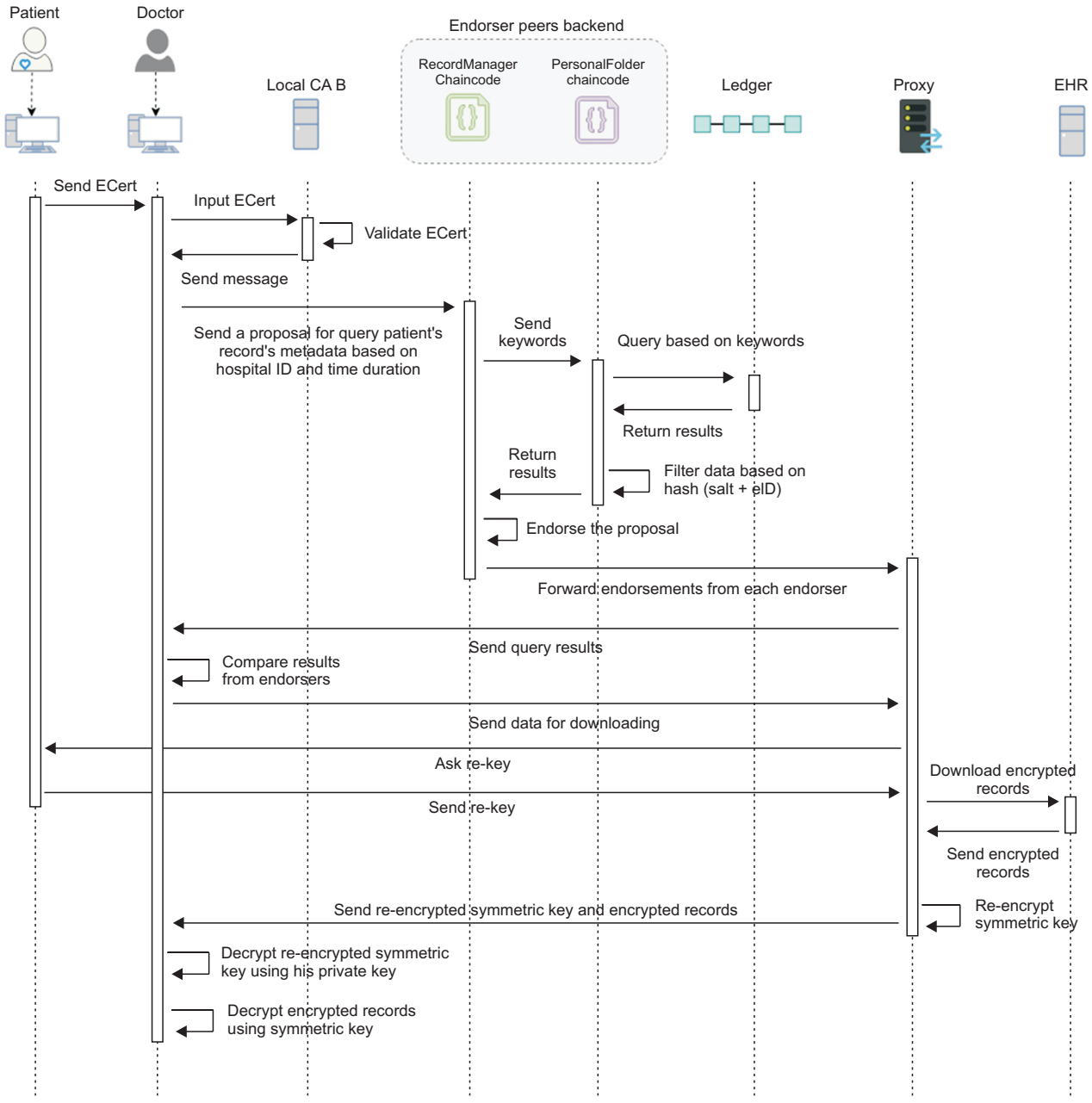

Figure 5. Requesting patient's records. CA: certificate authority, EHR: Electronic Health Record.

related with a patient having the same hash value of eID with which the patient records might be traced undesirably along the ledger. Meanwhile, this technique causes longer processing time to find out a patient in query of the data. To make the process faster, doctors can input many relevant query keywords for obtaining the data. These keywords include not only eIDs but timestamps and hospital IDs.

The proxy's roles are to connect different EHRs through a secured communication network, download the medical records and re-encrypt the patient's data. This scheme makes the processing time shorter in transferring a patient's data securely; otherwise, the data must be sent to the patient to decrypt using the patient's private key and encrypt again using the receiver's public key before it is sent back to the proxy and then to the receiver. For proxy re-encryption, we adopt the AFGH algorithm because it uses the receiver's public key rather than the private key as in BBS algorithm [22], where the receiver's private key is created and used transiently only for receiving the data.

To strengthen the privacy in access to records, patients can give consent with conditions in the transaction of records for sharing them to third party. Furthermore, the ledger retains events of sharing data and the relevant person's information, which facilitates the auditing procedure.

https://doi.org/10.4258/hir.2020.26.1.3

There have been several projects to establish a medical information-sharing system based on the blockchain. Among them, MedRec [23] is an early study applying the private Ethereum platform to EMRs. In Ethereum, an executable program run in the network is called a smart contract instead of a chaincode. Ethereum requires mining mechanisms to sustain the distributed ledger, which is a time-delayed process with miners competing in proof of work, although it is not difficult to make a private platform have a short block time less 10 seconds. Medical stakeholders, such as researchers, public health authorities, and so forth, need to be incentivized to participate actively as miners. To address these issues, MedRec 2.0 is currently under development [24].

Ancile [13] is another blockchain-based system using the private Ethereum platform, which applies a technique that is similar to ours for medical record management, adopting the on-chain and off-chain concept. Ancile uses distributed proxies for re-encryption, called blinding re-encryption, by splitting the ciphertext for re-encryption between multiple nodes.

On the other hand, Dubovitskaya et al. [25] uses HLF in the cloud system. In this system, the data structure consists of key and value pair. The key is a hash of a combination of the symmetric key and uniquely identifiable information (UII) of the patient, and the value is the record metadata. To reduce the vulnerability of the system, patients encrypt each piece of their data using different symmetric keys. However, this incurs a heavy burden of key management such that patients need to choose the corresponding symmetric key for generating a key number every time they query for the data.

Our system is a consortium network. If other medical institutions want to access this network, they must make a request to register as a member of this network. Otherwise, a non-member institution can communicate through the member institutions. Peers are the trusted elements from each medical institution. They need to strengthen their own security to protect peers from illegal access. At the same time, every medical institution needs to agree on the chaincode logic before deploying them in the system. Thus, our blockchain system also can be run effectively in the cloud system even though its fundamental standpoint is opposite in terms of decentralization. Cloud computing can provide a solution to the blockchain size problem that ledger size gets gradually bigger with time and peers will have difficulty to keep and process it.

In conclusion, our system can be used to constitute a large-scale EHR system. It is flexibly configurable to be a top layer of existing EHR systems to strengthen security in the management and exchange of medical records. Our system takes on the roles of a patient identifier, a trustee access log, and registry of patient records. Even though our system does not offer explicit incentives to participants as other blockchain-based systems do by issuing a cryptocurrency, it will benefit users and stakeholders too, including healthcare service providers and the government. We expect that our research can help patients to find their medical histories more easily when they visit other hospitals. As future work, we are going to test our system in a real hospital environment. We will prepare to deal with non-standardized data in a real-world field test.

Conflict of Interest

No potential conflict of interest relevant to this article was reported.

ORCID

Dara Tith (http://orcid.org/0000-0003-4372-7640)
Joong-Sun Lee (http://orcid.org/0000-0002-6976-6472)
Hiroyuki Suzuki (http://orcid.org/0000-0002-5028-5388)
W. M. A. B. Wijesundara (http://orcid.org/0000-0002-7228-524X)
Naoko Taira (http://orcid.org/0000-0001-6169-8957)
Takashi Obi (http://orcid.org/0000-0001-9430-2728)
Nagaaki Ohyama (http://orcid.org/0000-0002-4297-2575)

References

1. Greenhalgh T, Hinder S, Stramer K, Bratan T, Russell J. Adoption, non-adoption, and abandonment of a personal electronic health record: case study of HealthSpace. BMJ 2010;341:c5814.
2. Pylypchuk Y, Johnson C, Henry J, Ciricean D. Variation in Interoperability among US non-federal acute care hospitals in 2017. ONC Data Brief 2018;(42):1-15.
3. CommonWell Health Alliance. About CommonWell [Internet]. [place unknown]: CommonWell Health Alliance; c2020 [cited at 2020 Jan 10]. Available from: https://www.commonwellalliance.org/about/.
4. CommonWell Health Alliance. Use cases and specifications [Internet]. [place unknown]: CommonWell Health Alliance; c2020 [cited at 2020 Jan 10]. Available from: https://www.commonwellalliance.org/connect-to-the-network/use-cases-and-specifications/.
5. van der Linden H, Kalra D, Hasman A, Talmon J. Inter-organizational future proof EHR systems: a review of

the security and privacy related issues. Int J Med Inform 2009;78(3):141-60.
6. Keris MP. A Pandora's Box: the EMR's audit trail [Internet]. [place unknown]: EMR Discovery Blog; 2017 [cited at 2020 Jan 10]. Available from: https://www.emrdiscoveryintel.com/single-post/A-Pandoras-Box.
7. Walsh T, Miaoulis W. Privacy and security audits of electronic health information. J AHIMA 2014;85(3):54-9.
8. HIPAA compliance guide [Internet]. [place unknown]: The HIPAA Guide; c2017 [cited at 2020 Jan 10]. Available from: https://www.hipaaguide.net/hipaa-compliance-guide/.
9. Xu X, Weber I, Staples M, Zhu L, Bosch J, Bass L, et al. A taxonomy of blockchain-based systems for architecture design. Proceedings of 2017 IEEE International Conference on Software Architecture (ICSA); 2017 Apr 3-7; Gothenburg, Sweden. p. 243-52.
10. Sousa J, Bessani A, Vukolic M. A byzantine fault-tolerant ordering service for the hyperledger fabric blockchain platform. Proceedings of the 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN); 2018 Jun 25-28; Luxembourg City, Luxembourg. p. 51-8.
11. Androulaki E, Barger A, Bortnikov V, Cachin C, Christidis K, De Caro A, et al. Hyperledger fabric: a distributed operating system for permissioned blockchains. Proceedings of the 13th EuroSys Conference; 2018 Apr 23-26; Porto, Portugal.
12. Hyperledger Fabric. What is a blockchain [Internet]. [place unknown]: Hyperledger; c2019 [cited at 2020 Jan 10]. Available from: https://hyperledger-fabric.readthedocs.io/en/latest/blockchain.html.
13. Dagher GG, Mohler J, Milojkovic M, Marella PB. Ancile: privacy-preserving framework for access control and interoperability of electronic health records using blockchain technology. Sustain Cities Soc 2018;39:283-97.
14. Roehrs A, da Costa CA, da Rosa Righi R. OmniPHR: a distributed architecture model to integrate personal health records. J Biomed Inform 2017;71:70-81.
15. Kuo TT, Kim HE, Ohno-Machado L. Blockchain distributed ledger technologies for biomedical and health care applications. J Am Med Inform Assoc 2017;24(6):1211-20.
16. Manzoor A, Liyanage M, Braeke A, Kanhere SS, Ylianttila M. Blockchain based proxy re-encryption scheme for secure IoT data sharing. Proceedings of 2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC); 2019 May 14-17; Seoul, Korea. p. 99-103.
17. Thakkar P, Nathan S, Viswanathan B. Performance benchmarking and optimizing hyperledger fabric blockchain platform. Proceedings of IEEE 26th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS); 2018 Sep 25-28; Milwaukee, WI. p. 264-76.
18. Wang S, Ouyang L, Yuan Y, Ni X, Han X, Wang FY. Blockchain-enabled smart contracts: architecture, applications, and future trends. IEEE Trans Syst Man Cybern Syst 2019;49(11):2266-77.
19. Preneel B. Cryptographic hash functions: theory and practice. In: Gong G, Gupta KC, editors. Progress in cryptology – INDOCRYPT 2010. Heidelberg, Germany: Springer; 2010. p. 115-7.
20. Thangam V, Chandrasekaran K. Elliptic curve based proxy re-encryption. Proceedings of the 2nd International Conference on Information and Communication Technology for Competitive Strategies (ICTCS); 2016 Mar 4-5; Udaipur, India. p. 1-6.
21. Chow SS, Weng J, Yang Y, Deng RH. Efficient unidirectional proxy re-encryption. In: Bernstein DJ, Lange T, editors. Progress in cryptology – AFRICACRYPT 2010. Heidelberg, Germany: Springer; 2010. p. 316-32.
22. Ateniese G, Fu K, Green M, Hohenberger S. Improved proxy re-encryption schemes with applications to secure distributed storage. ACM Trans Inf Syst Secur 2006;9(1):1-30.
23. Azaria A, Ekblaw A, Vieira T, Lippman A. Medrec: using blockchain for medical data access and permission management. Proceedings of the 2nd International Conference on Open and Big Data (OBD); 2016 Aug 22-24; Vienna, Austria. p. 25-30.
24. MedRec [Internet]. Cambridge (MA): MIT Media Lab; c2019 [cited at 2020 Jan 10]. Available from: https://medrec.media.mit.edu/technical/.
25. Dubovitskaya A, Xu Z, Ryu S, Schumacher M, Wang F. Secure and trustable electronic medical records sharing using Blockchain. AMIA Annu Symp Proc 2018;2017:650-9.

 https://doi.org/10.4258/hir.2020.26.1.3