International Journal of Control and Automation
Vol. 10, No. 11 (2017), pp.133-144
http//dx.doi.org/10.14257/ijca.2017.10.11.12

Blockchain-Based User-Centric Records Management System

Si-Wan Noh[1], Youngho Park[2], Chur Sur[3], Sang-Uk Shin[2]
and Kyung-Hyune Rhee[2]

[1]Interdisciplinary Program of Information Security
Graduate School Pukyong National University, Republic of Korea
[2]Department of IT Convergence and Application Engineering
Pukyong National University, Republic of Korea
[3]Department of Information Security, Busan University of Foreign Studies,
Republic of Korea
[1]nosiwan@pukyong.ac.kr, [2]pyhoya@pknu.ac.kr, [3]kahlil@bufs.ac.kr
[2]shinsu@pknu.ac.kr, [2]khrhee@pknu.ac.kr

Abstract

Sharing of medical records among institutions can present certain risks to patient's privacy. An exposure of personal medical records causes damage to the finance, etc. Sometimes, it could put patient's life in danger. To achieve secure data sharing between institutions, many studies use cloud storage to store medical records. However, many studies suppose that cloud service providers were trustworthy. These systems required a key establishment with the cloud service providers and all the access logs are governed by cloud service providers. Patients should ask cloud service providers to check these access logs. However, it can be forged by cloud service providers during the request process. In this paper, we propose blockchain based medical records management system for secure data sharing. In order to share securely, we use proxy re-encryption scheme. Also, we use blockchain technology to audit access logs without modification.

Keywords*: Medical Record Management, Blockchain, Healthcare, Cloud Storage, Proxy Re-encryption*

1. Introduction

In the past, medical records were kept in the form of a paper document by hospitals. Hence, many patients received medical care service from their family doctor and a lot of medical records are kept by their family doctor. However, paper documents required space for storing it. Also, to retrieve a record from document box is difficult. These days, with the development of the technology, patients receive healthcare service from specialists in each field. To efficiently share medical records among specialists, electronic medical record (EMR) was proposed [1]. Traditional paper documents are converted to an electronic format with the associated information as a collection of records. These records are stored in the database and provided it only if an access request of this data is valid.

When patients receive medical service at hospitals, they need several examinations (e.g. inspection result, X-ray image, *etc*.) to receive medical treatment. Even though the patient had an examination just recently from the other hospital, it might lead to additional costs for the patient and the hospital. Sharing of medical records between specialists belonging to the same hospital is relatively easy to achieve. However, cross-institutional sharing of medical record is complex. To

Received (July 5, 2017), Review Result (October 30, 2017), Accepted (November 6, 2017)

ISSN: 2005-4297 IJCA

protect patient's privacy from unauthorized accesses, a lot of researchers studied about cloud storage server based data sharing system [2][4][9][13].

To share medical records with other institution, patients upload their medical records to the cloud server. Cloud computing service enables patients to provide their medical record for multiple users, including hospitals, pharmacy, insurer, family, *etc*. However, the stored record contains patient's sensitive information. An exposure of this information will cause damage to the finance, social status of patients, *etc*. Also, medical records are stored on a semi-trusted third-party server. That means the service provider can access medical records on the server without permission.

To overcome this problem, patients encrypt their medical records under a secret key before storing it to the cloud server. The easiest way to sharing encrypted records between different users is to share the secret key with users. However, it could allow a malicious user to access unauthorized data. In [17], Blaze *et al.*, introduced proxy re-encryption (PRE) scheme. With the PRE scheme, a data is encrypted under patient's public key before storing it. The patient generates proxy re-encryption key and sends it to the cloud server. Using proxy re-encryption key, the cloud server transforms the encrypted medical record under patient's public key into an encrypted medical record under requester's public key on the same record. In [8], Chul *et al.*, proposed a certificateless proxy re-encryption scheme based on bilinear pairing. The schemes prevent the proxy from colluding attack and achieve chosen ciphertext attack security. Even if the requester colludes with the cloud server, they cannot reveal patient's secret key.

In this paper, we consider the following situation: patients want to provide their medical record for a doctor, pharmacist and much more. They store medical records to the cloud server. In this situation, patients grant an access right to the medical record for the doctor and he/she can access to the patient's medical record on the cloud server until his/her access right is revoked by the patient. If patients want to audit record access logs, these logs must be kept by someone. In the cloud computing environment, the cloud service provider plays the role of an access control administrator and an auditor. However, if the cloud service provider is compromised, the patient obtains an incorrect information in process of auditing. Also, these logs contain patient's sensitive information like medical records. To prevent an unauthorized access to these logs, only the record owner can obtain the access logs.

To achieve these requirements, blockchain based access control techniques are proposed [10][11][12]. In [11], Damiano *et al.*, proposed blockchain based access control system. Each transaction expresses an access right and policies, all transactions are publicly visible on the blockchain. In their proposed system, all users can efficiently exchange their right with other users and their idea motivated our work. However, most studies are focused on access control; they do not consider privacy and secure data sharing.

Based on the above consideration, in this paper, we propose a blockchain based medical records management system for secure data sharing. In order to achieve our goal, we present a system design for user-centric data sharing without the participation of fully trusted third party. The outline of the rest of this paper is as follows. In the next section, we first give a system design with our security goals, and then describe our protocol in Section III. We analyze the proposed system in Section IV, and finally conclude this paper in Section V.

2. System Design

In this section, we present our system model, assumptions and security requirements. We consider a system model shown in Figure 1 which consists of a certification authority, cloud service provider, requesters, and patients.

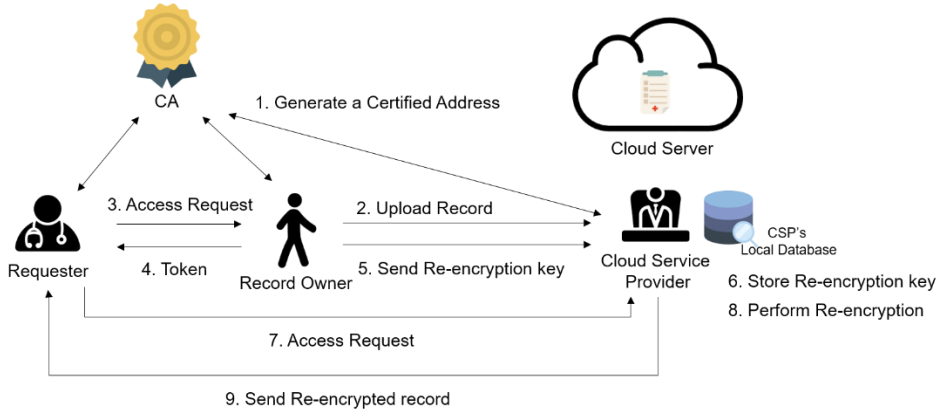


Figure 1. Proposed System Model

2.1. System Model

- **Patient.** A patient stores encrypted Personal Health Records(PHRs) onto the cloud server. When granting an access right to PHRs for requesters, the patient creates a token transaction and broadcasts it to the blockchain. At the same time, the patient generates a proxy re-encryption key to delegate decryption right to the grantor and sends it to the cloud service provider. In our system, all participants must be enrolled by the CA. In the enrollment phase, the patient needs e-mail authentication [7] without exposing his/her real-world identity. The patient's e-mail address is bounded to an address in the system.

- **Requester.** Requesters want to access patients medical record for research or medical treatment purposes. Unlike the patient, the requester must prove that he/she is qualified for the work to access medical records of patients (*e.g.*, national provider identifier). If the requester wants to access medical records that are stored on the cloud storage, he/she creates a request transaction. If the requester's qualifications suit the work, the patient creates a token transaction. It used to access records of the patient on the cloud storage.

- **Cloud Service Provider (CSP)**. The CSP is responsible for storing medical records of patients and transforms the encrypted medical record under patient's secret key into a new ciphertext under requester's secret key on the same record. The CSP stores re-encryption key for re-encryption to his/her local storage.

- **Certification Authority (CA).** In our System, we are using *a permissioned blockchain*. That means users need CA's permission to join the system. Only users authorized by CA can write on the Blockchain by creating transactions. To access medical records of different patients, the access requester must prove that the linkage between the address in the system and his identity in the real world. After that, the requester generates a certified address with the support of the CA. In our system, we consider the CA is a *functionally trusted* entity, *i.e.* the CA is assumed to be honest and fair, but it does not have access to the private keys of users.

We also make the following assumptions to clarify the proposed system.

- Public system parameters generated by the CA are already known to all the users.

- Efficient search methods that can allow users to search transactions on the blockchain are known to users.
- Consensus nodes for a block mining work use Bitcoin payment system. Users create a transaction and broadcast it to the Bitcoin blockchain network. The proposed system can be implemented by using the script language in Bitcoin.

2.2. Security Requirements

To design a medical record management for the data sharing in cloud storages, we consider the following security requirements as our design goals.

- **Integrity and confidentiality of the content of records.** Unauthorized users who do not have the access right should not be able to access the record of different users in the cloud storage. Even if the cloud service provider is compromised, medical records of the patient must be hidden from him/her. Additionally, authorized users should be able to verify the integrity of received records.
- **Auditability.** All actions of users in the system must be recorded on the blockchain. If patients want to audit all actions of users toward his/her medical record, he/she must be able to audit event records of the system without falsification.
- **Anonymity of patient identifiers on the blockchain.** Even though the medical record information of the patient is recorded on the public blockchain, no one can be related to the real-world identity of the patient.

3. Blockchain Based Medical Records Management

In this section, we describe the proposed secure medical record management system. The proposed system consists of setup, enrollment, grant access, get access, update and revocation phases. Table 1 shows the notations used to describe the proposed system.

Preliminaries. Before describing the proposed system, we provide a brief description of the properties of a bilinear pairing and proxy re-encryption. Let $\mathbb{G}$ is a finite group of size q and the additive group of integer residues modulo q is denoted by $\mathbb{Z}_q$. Also, $\mathbb{G}_1$and $\mathbb{G}_2$ be two multiplicative cyclic groups of prime order q. A bilinear map is a map e: $\mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ with the following properties:

- **Bilinear.** $e(g^a, g^b) = e(g,g)^{ab}$, for all a, b $\in \mathrm{Z}_q^*$ and $g \in \mathbb{G}_1$.
- **Non-degenerate.** If g is a generator of $\mathbb{G}_1$ then $e(g,g)$ is a generator of $\mathbb{G}_2$.
- **Computable.** $e(g,h)$ is efficiently computable for and $g, h \in \mathbb{G}_1$.

Setup. CA picks a random $\alpha_{CA} \in \mathbb{Z}_q$ as his/her master key and compute corresponding public key $\mathrm{PK_{CA}} = g^{\alpha_{CA}}$. CA chooses a random generator $g \in \mathbb{G}$ and function $\rho: \mathbb{G} \rightarrow \mathbb{Z}_q$, then publishes the system parameters $params_{CA} = (\mathbb{G}, q, g, PK_{CA}, \rho)$. CSP chooses bilinear map groups $(\mathbb{G}_1, \mathbb{G}_2)$ of order q and random generator $g_1, h \in \mathbb{G}_1$. CSP picks a random $\alpha_{CSP} \in \mathbb{Z}_q^*$ as his/her master key and compute a public key $PK_{CSP} = g^{\alpha} \in \mathbb{G}_1$. Also, CSP chooses hash functions $\mathrm{H}_1: \{0,1\}^* \rightarrow \mathbb{Z}_q^*, \mathrm{H}_2: \mathbb{G}_2^2 \rightarrow \{0,1\}^n, \mathrm{H}_3: \{0,1\}^* \rightarrow \mathbb{Z}_q^*, \mathrm{H}_4: \{0,1\}^* \rightarrow \mathbb{Z}_q^*, \mathrm{H}_5: \{0,1\}^* \rightarrow \mathbb{G}_1$ and computes the group elements $\mathrm{g}_2 = e(g,g), g_3 = e(g,h) \in \mathbb{G}_2$ then publishes the system parameters $params_{CSP} = (\mathbb{G}_1, \mathbb{G}_1, e, g_1, h, g_2, g_3, H_1, H_2, H_3, H_4, H_5)$.

Enrollment. In the enrollment phase, users generate a certified address with the support of CA [6] and he/she can attest to the involvement of the CA using the address.

After that, the user generates a private key for a proxy re-encryption using the certified address as an identity. The process is illustrated in Figure 2.

Patients (*i.e.* record owners) who want to enroll in the system create an *enrollment transaction* tx_{enroll} to generate a certified address and send a request message $m_{req,x}$ to the CA with enrollment information as follows:

(1) Select $k \leftarrow \mathbb{Z}_q$ uniformly at random and computes $h = g^k$.

(2) Send a request message $m_{req,PT_i} = \{email_{PT_I}, A_{PT_I}, pk_{PT_I}, \sigma_{sk_{PT_I}}(A_{PT_I}, email_{PT_I})\}$ to the CA.

(3) Create an enrollment transaction tx_{enroll} that contain the value h and $H(m_{req,PT_I})$.

(4) Broadcast the enrollment transaction tx_{enroll} to the blockchain network.

After receiving above message, when the transaction tx_{enroll} appears in the blockchain, the CA verifies signature and $H(m_{req,PT_I})$. The CA sends HTML link to verify ownership of an email address. After verifying it, the CA performs as follows:

(1) Select $k' \leftarrow \mathbb{Z}_q$ uniformly at random and computes a self-certified public key $c = h \cdot g^{k'}$.

(2) Compute $e = \rho(c)$ and $\bar{x} = k' + e \cdot \alpha_{CA}$.

(3) Create a response transaction tx_{resp} that contain the value e and $\bar{x}$.

(4) Broadcast the response transaction tx_{resp} to the blockchain network.

The patient computes his/her private key $x = \bar{x} + k$ and a certified address $cA_x = H(c)$. To verify this address, the verifier needs a signature signed by private key x. So the patient creates certification transaction tx_{cert} by sending transaction to himself.

After generating the address, the patient generates key pairs for a re-encryption process with the support of the CSP as follows:

(1) The patient creates *a key generation transaction* $tx_{key,1}$.

(2) The CSP computes $h_{PT_i} = H_1(cA_{PT_i}) \in \mathbb{Z}_q^*$ and extract patient's partial private key $d_{PT_i} = g^{1/(\alpha_{CSP}+h_{PT_i})} \in \mathbb{G}_1$.

(3) The CSP creates a *key generation transaction* $tx_{key,2}$ that contains a hash value of the partial private key $H(d_{PT_i})$ and sends the partial private key d_{PT_i} to the patient.

(4) The patient selects $a_1, a_2 \in \mathbb{Z}_q^*$ uniformly at random and sets $x_{PT_i} = (x_{PT_i,1}, x_{PT_i,2}) \in {\mathbb{Z}_q^*}^2$.

(5) The patient computes a private key for encryption $rSK_{PT_i} = (d_{PT_i}, x_{PT_i}) \in \mathbb{G}_1 \times {Z_q^*}^2$ and a public key $rPK_{PT_i} = \left(g_3^{x_{PT_i,1}}, g^{x_{PT_i,2}}\right) \in \mathbb{G}_2 \times \mathbb{G}_1$.

Also, requesters perform same processes (compute a certified address and key pair for re-encryption).

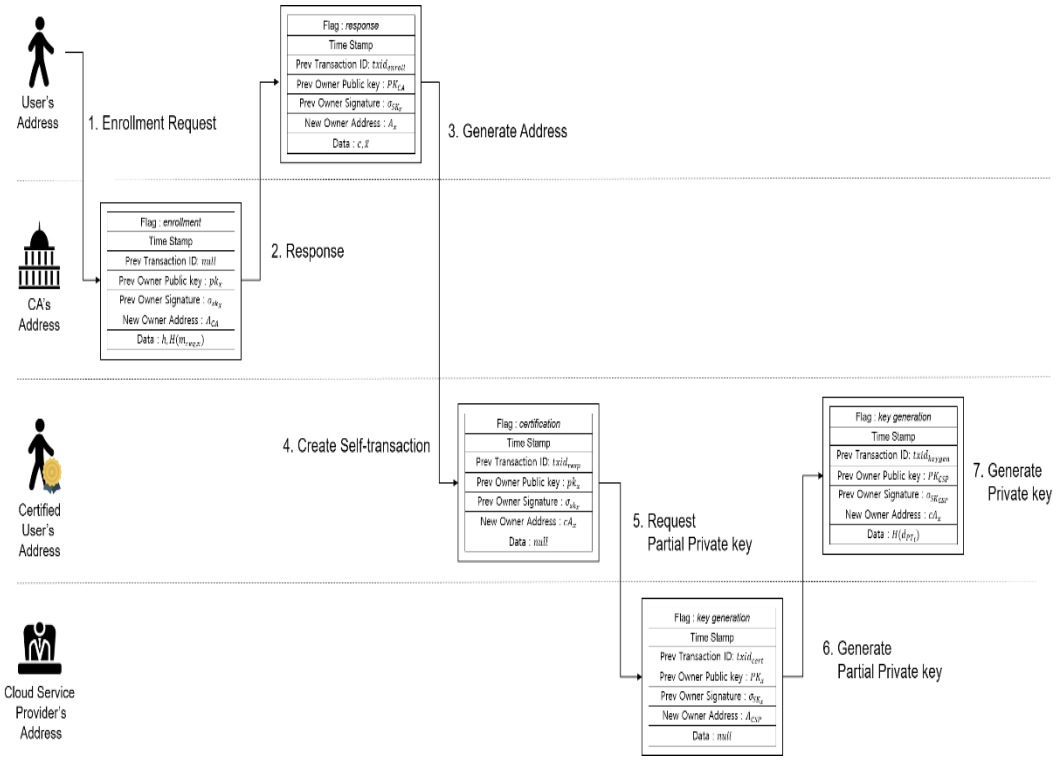


Figure 2. Enrollment Phase

Grant Access. After enrollment phase, the patient uploads his/her PHRs onto the cloud server with a record identifiable information and creates a *record transaction*. To protect the privacy of the patient, PHRs are encrypted by his/her public key. In this phase, the requester sends an access request message with a *request transaction*. If the patient wants to allow this request, he/she creates a *token transaction* and generates re-encryption key for the requester. Components of the stored record of the patient PT_i on the cloud server are as follows. The process is illustrated in Figure 3.

$$Recrod_{PT_i,j} = \{Enc_{rPK_{PT_i}}(data_j), H(data_j), cA_{PT_i}, txid_{rec_j}\sigma_{SK_{PT_i}}(H(data_j), cA_{PT_i})\}$$

If a doctor wants to access the patient's PHRs in the cloud server, the patient should inform his/her record transaction ID to the doctor. After acquiring it, the doctor creates a request transaction. After a moment, when the request transaction appears on the patient's blockchain, the patient creates a token transaction only if the requester's certification transaction is valid. In this phase, we consider two types of relationships between transactions.

The first is like an Unspent Transaction Output (UTXO) in Bitcoin [14]. In the Bitcoin payment system, only unspent outputs can be used as inputs to a new transaction. When a transaction is confirmed by consensus nodes, inputs are deleted from a UTXO pool and outputs are added to the pool as a new UTXO. In our system, similarly, the token transaction that is created by the record owner (patient) can be consumed as a token for the access record on the cloud server. Second is similar to the first one, however, the difference is allowed double spending (*i.e.* the single input is spent more than once). We call this relationship as a '*reference relationship*'. In our system, the record transaction ID of the patient is one of the input of the token transaction. If token transactions are created by the same patient, their input also uses the same transaction ID.

When the record owner creates a token transaction, he/she consumes the unspent output of the request transaction of the requester and refers to the record transaction of the record owner. If the token transaction is broadcast on the blockchain network, consensus nodes check that the signature in the inputs field is valid or not using the address in the

output filed of the previous transaction (both record transaction and request transaction). However, if the output of the record transaction is consumed by the record owner to update his/her record, this token transaction is invalid (If the requester wants to access the updated record, he/she needs to request the record again).

Before creating a token transaction, the record owner generates re-encryption key for the requester. The token transaction including a hash value that is based on the re-encryption key generated by the record owner and the address of the requester. The record owner sends the re-encryption key to the CSP with the transaction ID of the token transaction. The CSP stores it in a local storage with the transaction ID. The record owner computes the re-encryption key $rk_{PT_i \to req}$ the following steps:

(1) Choose a random $s \in \mathbb{Z}_q^*$ and compute $\mu = H_3(g_2^s \| cA_{PT_i} \| rPK_{PT_i} \| cA_{req} \| rPK_{req})$.

(2) Compute $h_{PT_i} = H_1(cA_{PT_i})$ and $h_{req} = H_1(cA_{req})$.

(3) Set the proxy re-encryption key $rk_{PT_i \to req} = \left(rk_{PT_i \to req}^{(1)}, rk_{PT_i \to req}^{(2)}, rk_{PT_i \to req}^{(3)}\right) = \left(g^{(\alpha_{CSP} + h_{PT_i})}\right), (g^{h_{req}} \cdot g_1), (g^{x_{req,2}})^{x^{PT_i,1}}$.

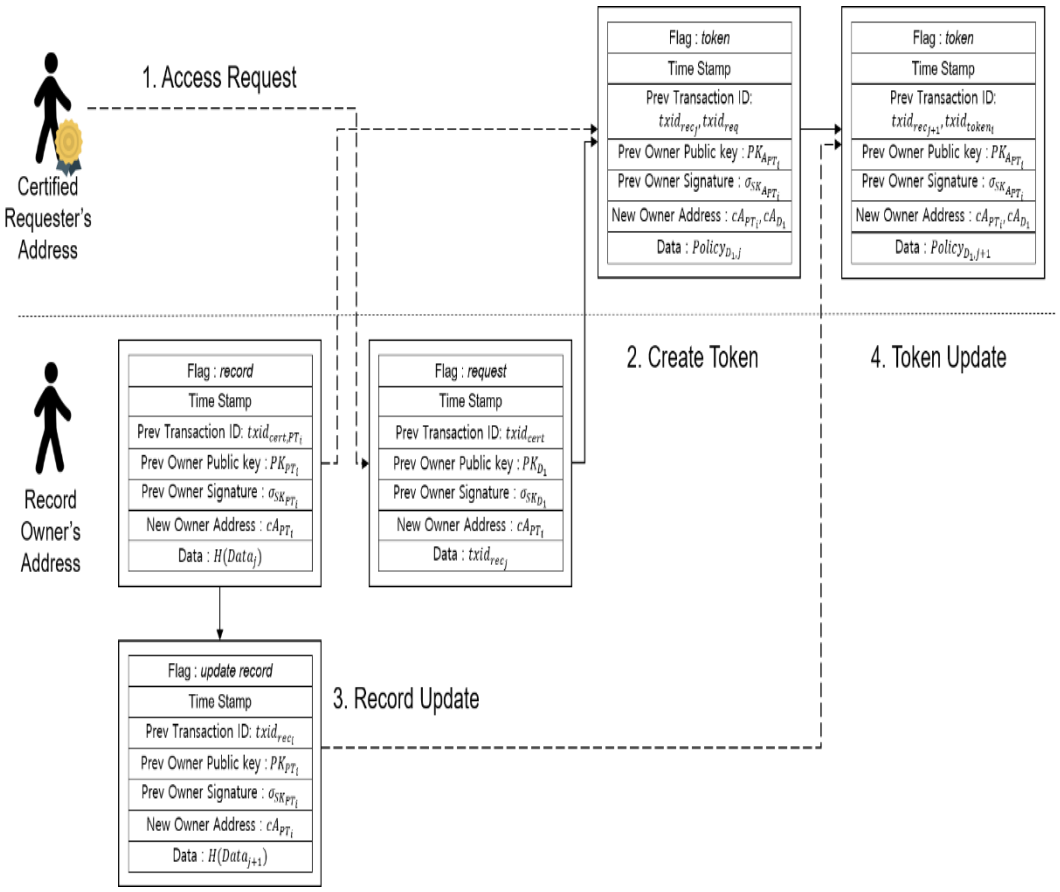


Figure 3. Grant Access Phase

Get Access. After gaining a token to the medical record on the cloud server, the requester creates *a get access transaction* tx_{get} to get the re-encrypted medical record. When the transaction is added to the CSP's blockchain, the CSP verifies requester's token transaction. Only when the token transaction is valid and the comparison of a hash value of the re-encryption key stored in local storage with the hash value in the token transaction is the same, the CSP performs re-encryption process as a proxy and sends a re-

encrypted medical record to the requester. After sending it, the CSP creates *a receipt transaction* for the record owner. In this paper, we skip detailed processes of the re-encryption without key generation. The process is illustrated in Figure 4.

1. The requester creates a get access transaction

2. CSP checks the validation of the request and performs re-encryption process using the corresponding proxy re-encryption key $rk_{PT_i \rightarrow req}$ in the local storage.

3. CSP sends the re-encrypted record $Recrod_{req,j}$ to the requester.

4. The requester decrypts received record using his/her private key rSK_{req}.

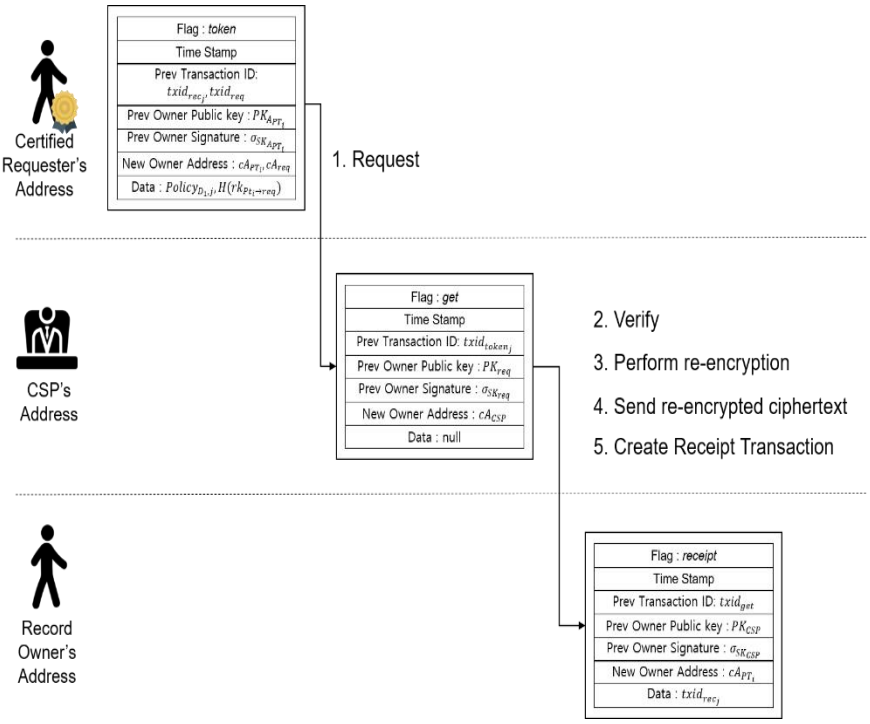


Figure 4. Get Access Phase

Revocation. The output filed of the token transaction contains the requester's address and the record owner's address in the system. That is, the record owner can consume the token transaction as well as the requester. The record owner could revoke the token transaction at any time by consuming its output. It can be implemented using Multisignature (multisig) in the Bitcoin payment system. It requires the signature of multiple users before the output of the transaction can be consumed. In our system, we use 1-of-2 multisig to achieve our requirement. After spending the token transaction by the record owner, it cannot be used for record access.

Update. The patient's PHRs are frequently updated. However, the record transaction contains a hash value of PHRs at some point in the past. In our system, the record owner can update his/her record transaction on the blockchain using *an update record transaction*. The record owner just consumes the output of the previous record transaction and creates a new record transaction with the hash value of the updated record. After creating the updated record transaction, unused token transactions are no longer available.

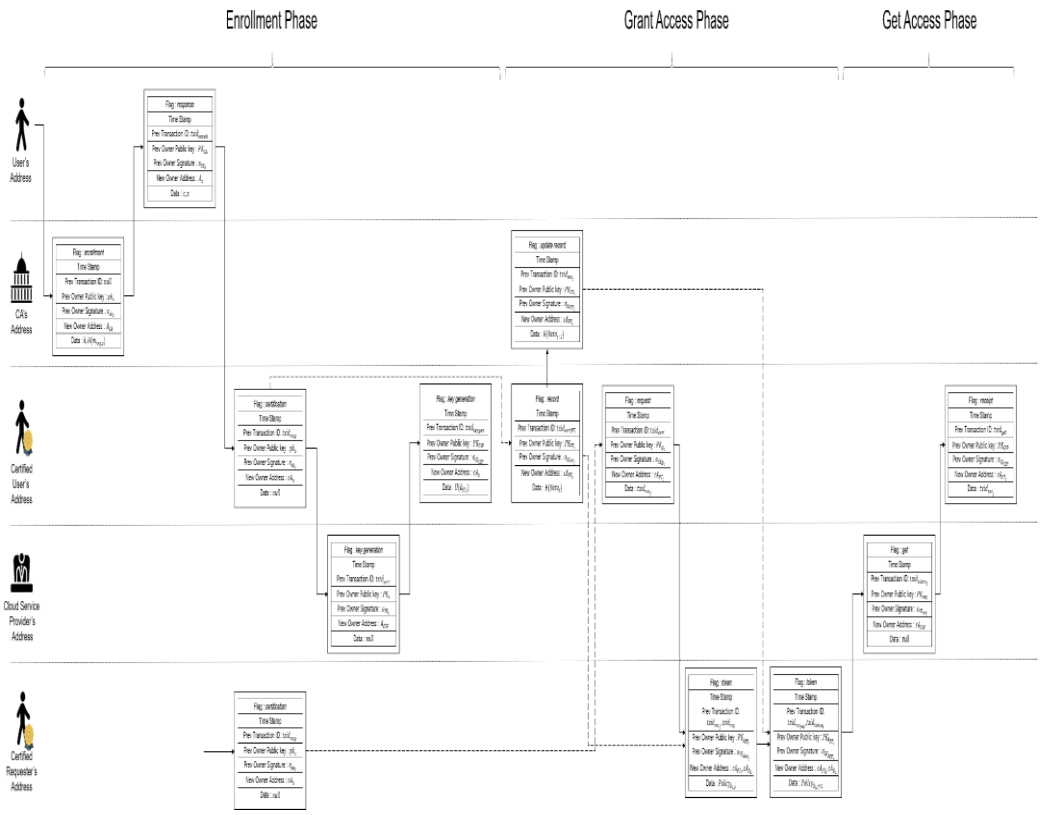


Figure 5. Transaction Transfer Process4. Security Analysis

4.1. Integrity and Confidentiality of the Content of Records

In our system, patients store their medical records to the cloud storage. Records in the cloud server are encrypted by their own secret key to protect their privacy from malicious cloud service provider. When patients upload their medical records to the cloud server, they create a record transaction that contains a hash value of records. It is like a timestamp of the record; the requester can verify the validity of the record when the requester gets it. If malicious cloud service provider wants to change some record in the cloud server for malicious, he/she must change a corresponding record transaction and the block header that containing the record transaction. However, it will be hard, because of an immutability of the blockchain. Also, a confidentiality of the content of records is can be guaranteed by a proof of the scheme in [8].

4.2. Auditability

After the requester using the token transaction (*i.e.* create get access transaction), the cloud service provider creates a receipt transaction for the record owner. The record owner can trace the access log back through connected transactions. All transactions in the blockchain include a timestamp of the user and an address in the system. It can be possible that the record owner obtains a timeline for auditing. Also, the record owner can obtain these logs without a communication with the cloud service provider.

4.3. Anonymity of Patient Identifiers on the Blockchain

In the enrollment phase, patients provide their email address to the CA. In our system, the CA does not need real-world identities of patients to register. Even if someone provides a hacked email-address to the CA in the enrollment phase, it does not affect our system. In the case of the Bitcoin, real-world identities are revealed in

the buying process. However, in our system, real-world identities of users are not revealed across all phases of system without exposing the relationship between the identity and the address themselves.

5. Conclusion

In cloud based data sharing system, cloud service providers can know all accesses to stored records. However, if the record owner wants to know access logs to the stored record, he/she asks them for an auditing. If the cloud service provider is compromised, the record owner obtains incorrect result. In this paper, we proposed a blockchain based secure data sharing system in the cloud storage. Due to the features of a blockchain technology, we can make record owners control who can access to their medical record in the cloud server without participant of the fully trusted third party.

Acknowledgments

This paper is a revised and expanded version of a paper entitled “Scalable Access Control Based on Blockchain for Healthcare Service System” presented at 13th International Conference on Multimedia Technology and Applications (MITA 2017) took place from July 9 to 11 in Kuala Lumpur, Malaysia.

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT) (No.2017-0-00156, The Development of a Secure Framework and Evaluation Method for Blockchain)

References

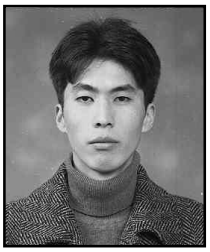
[1] T. J. Hannan, “Electronic medical records”, Health informatics: An overview, vol. 133, **(1996)**.
[2] T. Ermakova and B. Fabian, “Secret sharing for health data in multi-provider clouds”, Proceedings of the 15th Conference of Business Informatics (CBI), Vienna, Austria, **(2013)**.
[3] M. Li, S. Yu, Y. Zheng, K. Ren and W. Lou, “Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption”, IEEE transactions on parallel and distributed systems, vol. 24, no. 1, **(2013)**, pp. 131-143.
[4] S. Ruj, M. Stojmenovic and A. Nayak, “Decentralized access control with anonymous authentication of data stored in clouds”, IEEE transactions on parallel and distributed systems, vol. 25, no. 2, **(2014)**, pp. 384-394.
[5] W. S. Ng, B. C. Ooi, K. L. Tan and A. Zhou, “Peerdb: A p2p-based system for distributed data sharing”, Proceedings of the 19th International Conference on Data Engineering (ICDE03), **(2003)**.
[6] G. Ateniese, A. Faonio, B. Magri, and B. De Medeiros, “Certified bitcoins”, Proceedings of the 12th International Conference on Applied Cryptography and Network Security(ACNS), Lausanne, Switzerland, **(2014)**.
[7] S. L. Garfinkel, “Email-based identification and authentication: An alternative to PKI?”, IEEE Security & Privacy, vol. 99, no. 6, **(2003)**, pp. 20-26.
[8] C. Sur, C. Jung, Y. Park and K. Rhee, “Chosen-ciphertext secure certificateless proxy re-encryption”, Proceedings of the 11th International Conference on Communications and Multimedia Security, Linz, Austria, **(2010)**.
[9] Q. Xia, E. B. Sifah, K. O. Asamoah, J. Gao, X. Du and M. Guizani, “MeDShare: Trust-Less Medical Data Sharing Among Cloud Service Providers via Blockchain”, IEEE Access, vol. 5, **(2017)**, pp. 14757-14767.
[10] A. Ouaddah, A. Elkalam and A. A. Ouahman, “Towards a novel privacy-preserving access control model based on blockchain technology in IoT”, In Europe and MENA Cooperation Advances in Information and Communication Technologies, Springer International Publishers, **(2017)**, pp. 523-533.
[11] D. D. F. Maesa, P. Mori and L. Ricci, “Blockchain Based Access Control”, Proceedings of the 17th International Conference on Distributed Applications and Interoperable Systems, Neuchâtel, Switzerland, **(2017)**.
[12] Q. Xia, E. B. Sifah, A. Smahi, S. Amofa and X. Zhang, “BBDS: Blockchain-Based Data Sharing for Electronic Medical Records in Cloud Environments”, Information 2017, vol. 8, no. 2, **(2017)**, pp. 44-59.
[13] S. Yu, C. Wang, K. Ren and W. Lou, “Achieving secure, scalable, and fine-grained data access control in cloud computing”, Proceedings of the 29th Conference on Compute Communication, San Diego, CA, USA, **(2010)**.
[14] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system”, **(2008)**.

[15] G. Zyskind, and O. Nathan, "Decentralizing privacy: Using blockchain to protect personal data", Proceedings of the 36th IEEE Symposium on Security and Privacy Workshops (SPW), **(2015)**.
[16] A. Dubovitskaya, Z. Xu, S. Ryu, M. Schumacher and F. Wang, "How Blockchain Could Empower eHealth: An Application for Radiation Oncology", Proceedings of the 3rd International Workshop on Data Management and Analytics for Medicine and Healthcare, Munich, Germany, **(2017)**.
[17] M. Blaze, G. Bleumer and M. Strauss, "Divertible protocols and atomic proxy cryptography", Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques, Espoo, Finland, **(1998)**.

Authors

Si-wan Noh, he received his B.S. degree in Department of IT Convergence and Application Engineering from Pukyong University, Republic of Korea in 2016. He is currently a master course student of Pukyong National University. His research interests are related with blockchain, applied cryptography, and communication security.

Youngho Park, he received his M.S. and Ph.D. degrees in Department of Computer Science and Information Security from Pukyong National University, Republic of Korea in 2002 and 2006, respectively. He is currently a researcher of Electronics and Information Telecommunication Research Center, Pukyong National University. His research interests are related with applied cryptography, communications security, secure vehicular ad hoc network, authentication, and key management.



Chul Sur, he received his M.S. and Ph.D. degrees in Department of Computer Science from Pukyong National University, Republic of Korea in 2004 and 2010, respectively. He is currently an assistant professor in the Department of Information Security, Busan University of Foreign Studies, Republic of Korea. His research interests are related with applied cryptography and its applications, network security, and secure e-commerce.

Sang Uk Shin, he received his M.S. and Ph.D. degrees in Department of Computer Science from Pukyong National University, Republic of Korea in 1997 and 2000, respectively. He worked as a senior researcher in Electronic and Telecommunications Research Institute (ETRI), Republic of Korea from 2000 to 2003. He is currently a professor in the Department of IT Convergence and Application Engineering, Pukyong National University, Republic of Korea. His research interests are related with cryptographic protocol, mobile network security, digital forensic and e-discovery.

Kyung-Hyune Rhee, he received his M.S. and Ph.D. degrees from Korea Advanced Institute of Science and Technology (KAIST), Republic of Korea in 1985 and 1992, respectively. He worked as a senior researcher in Electronic and Telecommunications Research Institute (ETRI), Republic of Korea from 1985 to 1993. He is currently a professor in the Department of IT Convergence and Application Engineering, Pukyong National University, Republic of Korea. His research interests center on applied cryptography, communication and network security, and blockchain.