Original article

eISSN 2799-8010
J Yeungnam Med Sci 2022;39(2):98-107
https://doi.org/10.12701/yujm.2021.01368

JYMS
JOURNAL OF YEUNGNAM MEDICAL SCIENCE

Storing information of stroke rehabilitation patients using blockchain technology: a software study

Min Cheol Chang

Department of Physical Medicine and Rehabilitation, Yeungnam University College of Medicine, Daegu, Korea

Background: Stroke patients usually experience damage to multiple functions and a long rehabilitation period. Hence, there is a large volume of patient clinical information. It thus takes a long time for clinicians to identify the patient's information and essential pieces of information may be overlooked. To solve this, we stored the essential clinical information of stroke patients in a blockchain and implemented the blockchain technology using the Java programming language.
Methods: We created a mini blockchain to store the medical information of patients using the Java programming language.
Results: After generating a unique pair of public/private keys for identity verification, a patient's identity is verified by applying the Elliptic Curve Digital Signature Algorithm based on the generated keys. When the identity verification is complete, new medical data are stored in the transaction list and the generated transaction is verified. When verification is completed normally, the block hash value is derived using the transaction value and the hash value of the previous block. The hash value of the previous block is then stored in the generated block to interconnect the blocks.
Conclusion: We demonstrated that blockchain can be used to store and deliver the patient information of stroke patients. It may be difficult to directly implement the code that we developed in the medical field, but it can serve as a starting point for the creation of a blockchain system to be used in the field.

Keywords: Blockchain; Information services; Patients; Rehabilitation; Stroke

Introduction

A blockchain, also called a distributed or shared ledger, is a technology through which participants jointly verify, store, distribute, and interconnect data without an authorized third party by generating data in blocks [1,2]. It allows participants to jointly record data by distributing the data to a person-to-person (P2P) network rather than to the central server of a specific organization [1,2]. Those who are permitted to see the ledger read it according to an agreed method, and the transactions are also recorded according to an agreed method. In this way, information can be stored securely without a central server, and the content can be trusted without a third-party guarantee. The applications of blockchain have been expanding to various fields including government, finance, and public data, and its use is also being explored in the medical field [3-5].

Currently, patient medical information is stored on hospital servers, and hospitals cannot easily exchange the patient information stored on such servers with other hospitals because of the risk of a privacy breach [1]. Consequently, when patients are transferred to another hospital, they need to carry their medical information from one hospital to the next. Conversely, if blockchain were used in place of servers, the information would be generated

Received: July 28, 2021 • **Revised:** August 13, 2021 • **Accepted:** August 13, 2021
Corresponding author: Min Cheol Chang, MD
Department of Physical Medicine and Rehabilitation, Yeungnam University College of Medicine, 170 Hyeonchung-ro, Nam-gu, Daegu 42415, Korea
Tel: +82-53-620-4682 • Fax: +82-53-4231-8694 • E-mail: wheel633@gmail.com

and recorded in block units and then stored on multiple nodes in a distributed manner [6]. This would make hacking practically impossible while allowing hospitals to easily and freely share patient information with other hospitals.

During the rehabilitation of a stroke patient, a large amount of data related to the patient's condition is generated due to the long rehabilitation period [7]. Furthermore, many patients are transferred to different hospitals to receive rehabilitation treatments. When a patient moves to another hospital, the patient or their guardian must receive the printed patient care information from the old hospital and deliver it to the new hospital [1,7]. Doctors check the current and past conditions of transferred patients based on these paper records, which takes a considerable amount of time. Moreover, important pieces of information may be missed due to the large volume of information and because each hospital records patient information in its own way. We believe that blockchain can help to solve this problem.

Based on the previous study [7], we identified the essential information for stroke patients receiving rehabilitation treatment and then used private blockchain network technology to store the information of one patient through the medical information transaction process. Since the personal medical information of patients can be transferred on the network, only the participants who were authorized to use the private blockchain were allowed to write and read the patient medical information.

Methods

We created a mini blockchain to store the medical information of patients using the Java programming language. To actually run the source code written in Java, we installed the Java Development Kit and Eclipse, the most representative integrated Java development environment.

Results

The medical information transaction process is illustrated in Fig. 1.

1. Patient consent

A patient's consent is required before sharing their personal medical information on the network. The procedure for obtaining consent is conducted through an application. Once the patient's consent has been obtained, a unique pair of public/private keys is issued for the patient. The issued keys are stored as files: the private key is stored on the patient's smartphone, and the public key is stored by the medical institutions that participate in the network. In general, keys are stored in a byte format, which is difficult to read or process. However, the format in which the key data are encoded and managed can be conveniently viewed and transmitted using the Base64 algorithm (an encoding algorithm that converts binary data to text). An encoded file is called 'Privacy-Enhanced Mail' and generally has the file extension '.pem'. The keys are randomly generated using the chart number or resident registration number of the patient.

Code: Generation of private and public keys

A unique pair of private/public keys are generated for a given patient using the Elliptic Curve Digital Signature Algorithm (ECDSA) algorithm. The generated keys have the file extension '.pem' and are stored under a specific file name.

Generation of private and public keys using the Elliptic Curve Digital Signature Algorithm

```
// Private and public keys are generated using the ECDSA algorithm and then stored.
// For a detailed version of the ECDSA, we use sect163k1.
public void generate (String privateKeyName, String publicKeyName) throws Exception {
// The ECDSA algorithm of the Bouncy Castle is used.
KeyPairGenerator generator = KeyPairGenerator.getInstance("ECDSA", "BC");
// sect163k1 is the algorithm used to generate the elliptic curve.
ECGenParameterSpec ecsp;
ecsp = new ECGenParameterSpec("sect163k1");
generator.initialize(ecsp, new SecureRandom());
// A random pair of keys is generated using this algorithm.
KeyPair keyPair = generator.generateKeyPair();
System.out.println("A pair of elliptic curve encryption keys was generated.");
// The private and public keys are extracted from the generated keys.
PrivateKey priv = keyPair.getPrivate();
PublicKey pub = keyPair.getPublic();
// The private and public keys are stored under specific file names.
writePemFile(priv, "EC PRIVATE KEY", privateKeyName);
writePemFile(pub, "EC PUBLIC KEY", publicKeyName);
}

// The generated encryption key is save as a file in the .pem class.
private void writePemFile(Key key, String description, String filename) throws FileNotFoundException, IOException{
Pem pemFile = new Pem(key, description);
pemFile.write(filename);
```

(Continued to the next page)

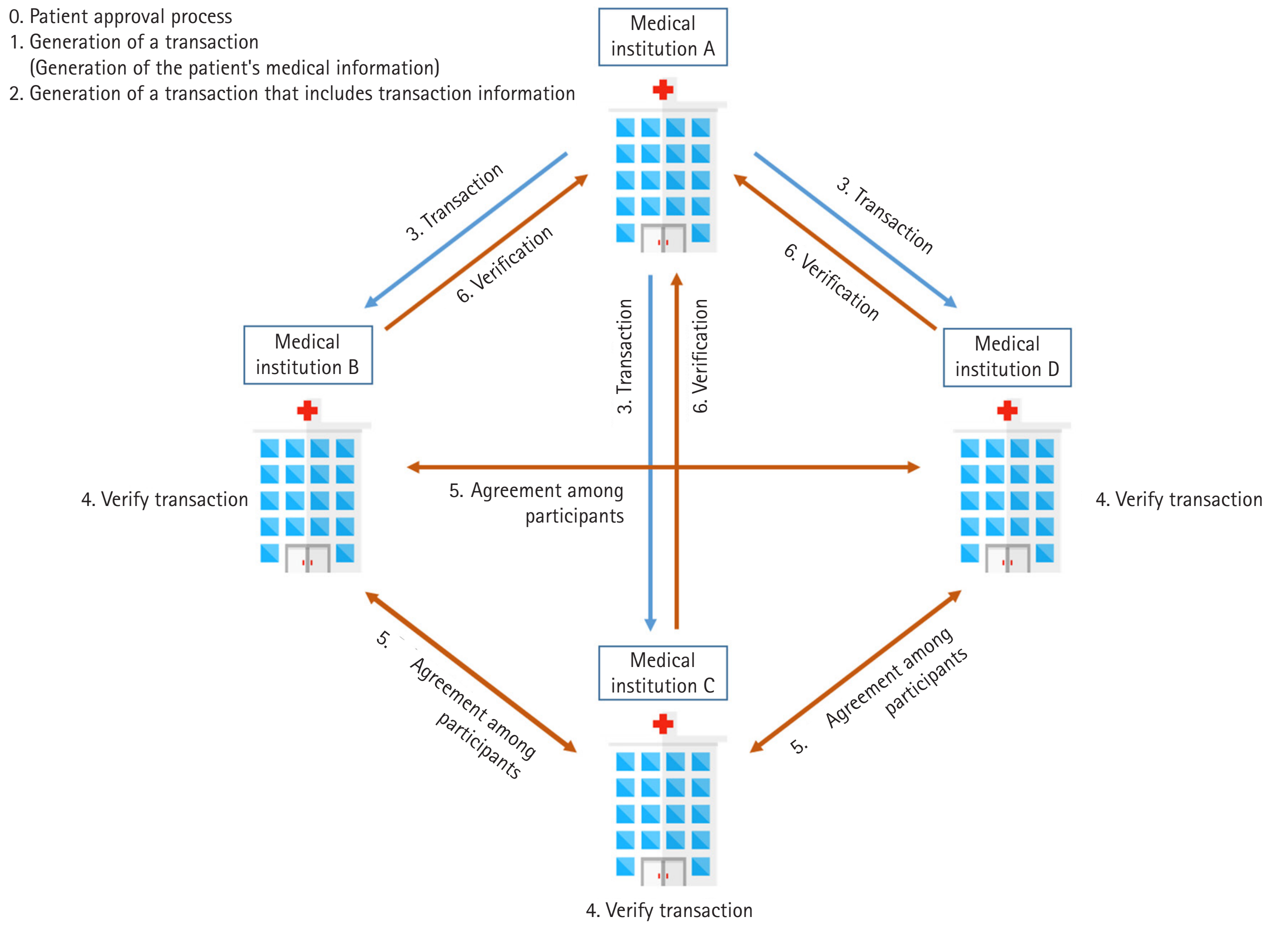
0. Patient approval process
1. Generation of a transaction
(Generation of the patient's medical information)
2. Generation of a transaction that includes transaction information
Medical institution A
Medical institution B
Medical institution C
Medical institution D
3. Transaction
6. Verification
3. Transaction
6. Verification
3. Transaction
6. Verification
4. Verify transaction
5. Agreement among participants
4. Verify transaction
5. Agreement among participants
5. Agreement among participants
4. Verify transaction

Fig. 1. Application of blockchain technology to medical information: patient identities are verified using the public key–based structure, and the medical records of the verified patients are connected as a chain through the hash value.

```
System.out.println(String.format("Encryption key %s was exported to the file %s.", description, filename));
}
```

When the private and public keys stored in the .pem files are opened, they have the following format.

Format of private and public keys

```
-----BEGIN EC PRIVATE KEY-----
MGwCAQAwEAYHKoZIzj0CAQYFK4EEAAEEVTBTAg
EBBBUCoeEILPFbMQIh3CRiHo+S
3++ka8egBwYFK4EEAAGhLgMsAAQCupGyQ46vQ9dG6w-
tab7xZtJMFdcIGu0fHQR+Z
OnNX8k/xYjkrRjGTBEU=
-----END EC PRIVATE KEY-----

-----BEGIN EC PUBLIC KEY-----
(Continued to)
```

```
MEAwEAYHKoZIzj0CAQYFK4EEAAEDLAAEArqRskOO
r0PXRusLWm+8WbSTBXXCBrtH
x0EfmTpzV/JP8WI5K0YxkwRF
-----END EC PUBLIC KEY-----
```

2. Identification

The purpose of the identification process is to verify the identity of patients using a public key-based structure (Fig. 2). The function of the public key-based structure is to manage passwords as pairs of private and public keys using the ECDSA algorithm. Data encrypted with a private key can only be decrypted with the public key with which it is paired. The private key is encrypted using the patient's chart number or resident registration number and then decrypted by reading the public key. If the patient's public key or private key does not exist or has been manipulated or damaged, identity verification fails, and it is impossible to access the patient's medical information.

Code: Identity verification process

The private and public keys of a given patient are read, and their va-

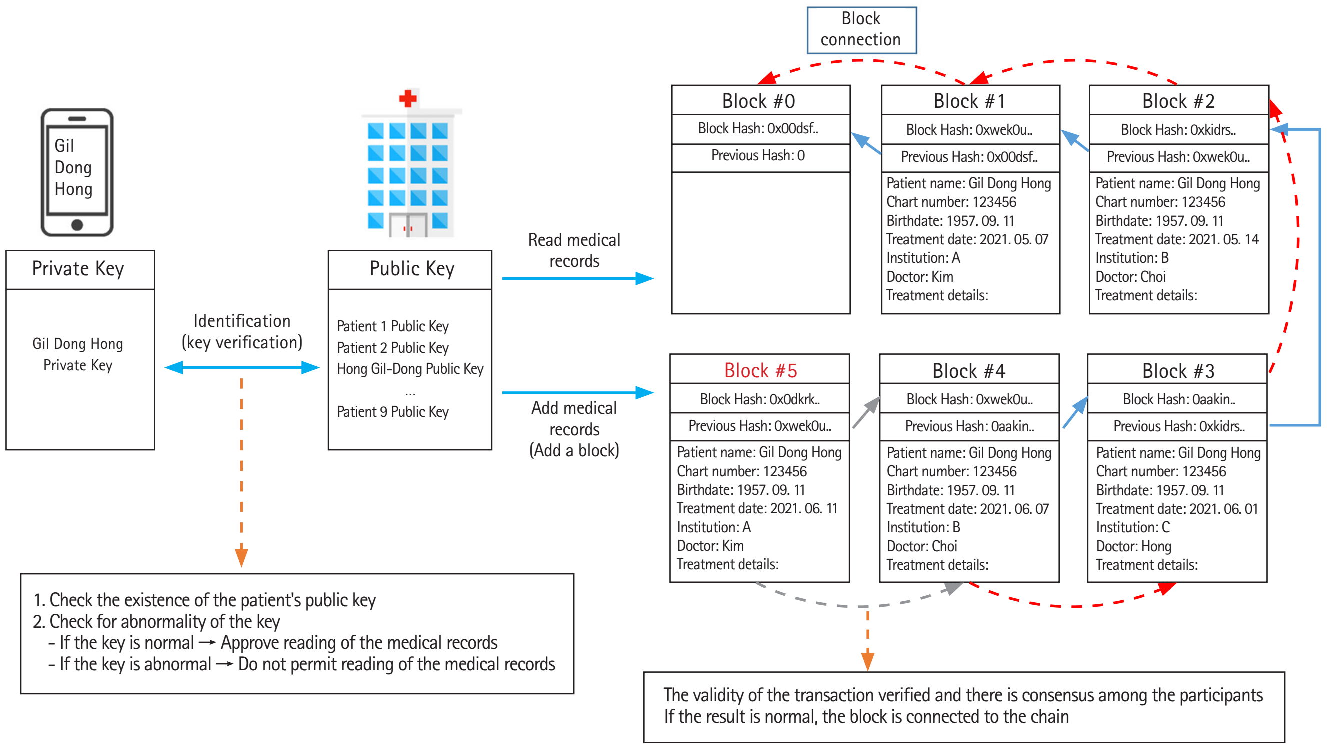

Fig. 2. Encryption process of the public key-based structure. After the consent to use and share personal medical information is provided, a unique pair of keys (private key, public key) is issued for the patient. An identity verification process is required to read or update medical records. The process is performed using the unique key pair assigned to the patient. Once the identity is verified normally, the patient's past medical records can be read, and new ones can be written. A new medical record is stored in a block, and if no abnormality is observed through validation verification, the new medical record is linked to the previous block and stored.

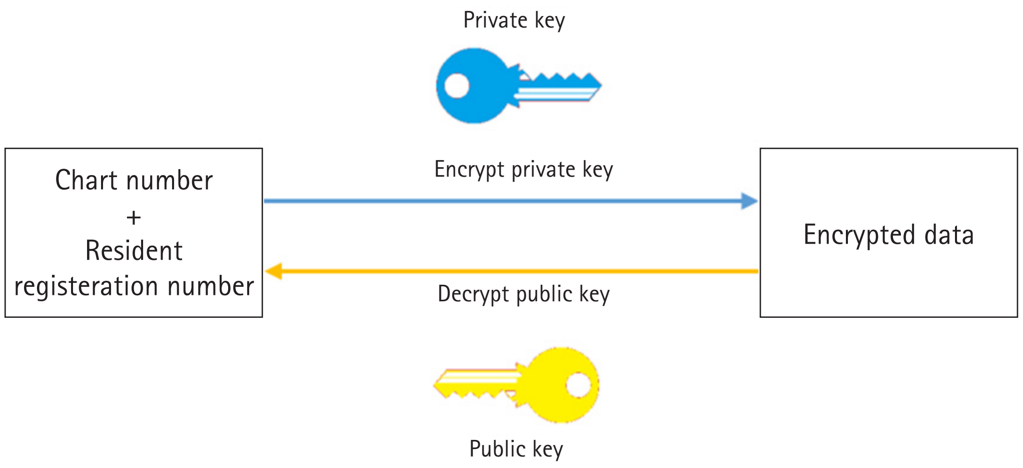

Fig. 3. Distribution and agreement process of medical information transaction. The verification and agreement process for medical information transaction is shown. The patient's identity verification process uses a public key-based structure that uses the Elliptic Curve Digital Signature Algorithm. The public key-based structure is a method of managing passwords with a pair of private and public keys, and facilitates the encryption of private keys using chart numbers or resident registration numbers. Data encrypted using the private key can be decrypted by only using the paired public key.

lidity is verified.

A text string file (.pem) is read and the private and public keys are extracted. The file is encrypted (signed) using the private key and decrypted using the public key.

Verification of the validity of private and public keys

```
//The private and public keys are read and initialized.
public void setFromFile(String privateKey, String publicKey) throws Exception {
//A function to extract the private key from the certificate of a text string format:
this.privateKey = new EC().readPrivateKeyFromPemFile(privateKey);
//A function to extract the public key from the certificate of a text string format:
this.publicKey = new EC().readPublicKeyFromPemFile(publicKey);
}

//The validity of the key is verified.
public static boolean isKeyVerify(String id, PrivateKey privateKey, PublicKey publicKey) throws Exception{
boolean result;
Signature ecdsa;
Signature signature;
String text;
byte[] baText;
byte[] baSignature;
//Encryption (signature) using the private key:
ecdsa = Signature.getInstance("SHA1withECDSA");
ecdsa.initSign(privateKey);
text = id;
baText = text.getBytes("UTF-8");
//The data from the original source that are encrypted and signed are output.
ecdsa.update(baText);
baSignature = ecdsa.sign();
//Decryption using the public key when verifying the data:
signature = Signature.getInstance("SHA1withECDSA");
signature.initVerify(publicKey);
signature.update(baText);
result = signature.verify(baSignature);
return result;
}
```

3. Reading and adding medical information

After successful identity verification, the medical institution can read the patient's past medical records and can update the record with new medical information (Fig. 3). The medical information is updated through the propagation of a transaction. Moreover, a medical record can be created, and the message by which it is created is expressed as a transaction. Transactions are propagated to participating medical institutions connected to the network. Each participating medical institution that receives a transaction updates its transaction information and propagates the transaction to another medical institution. When a medical institution receives a transaction, it needs a means to verify whether the transaction is correct. A digital or electronic signature is required for validity verification, and the original data (the patient's medical information) is therefore sent together with electronically signed data to verify that the transaction is correct and that the data has not been modified. Successfully verified transaction information is reflected in the hash value of the generated block.
Code: Validity test
A transaction object that records medical information is generated, and the validity of the transaction is verified.

Verification of the validity of transactions containing patients' medical information

```
transaction = new Transaction(key, patientsData.getPatientsData());

//The transaction information to be propagated is created.
//The transaction information includes the patient's medical information, key information, electronically signed data, and transaction generation time.
public Transaction(Key key, String patientsData) throws Exception {
this.patientsData = patientsData
this.sender = key.getPublicKey();
this.timestamp = Util.getDate();
this.signature = key.sign(getData()); //Electronic signature of original data
}

//The simple transaction information, excluding the signature value, is returned.
public String getData() {
return new Util().getHash(sender.toString()) + timestamp + this.patientsData
}

//The normality of the transaction is verified (a validity test).
public boolean verifyTransaction(Transaction transaction) throws Exception {
Signature signature;
```
(Continued to the next page)

```
signature = Signature.getInstance("SHA1withECDSA");
byte[] baText = transaction.getData().getBytes("UTF-8");
signature.initVerify(transaction.getSender());
signature.update(baText);
return signature.verify(new BigInteger(transaction.getSigna-
ture(), 16).toByteArray());
}
```

The process for adding medical information to a patient's medical record is as follows: (1) The original data (medical information) is encrypted with the patient's own private key and electronically signed. (2) The original data and the electronically signed data are propagated to the participating medical institutions. (3) To verify whether the transaction is valid, the participating medical institutions decrypt it using the patient's public key. (4) The decrypted data and original data are compared to verify the integrity of the data and whether there is any manipulated data. (5) If the received transaction is determined to be valid, the transaction is updated in the blockchain, and the transaction is propagated to the medical institutions participating in the blockchain network.

4. Block connection

Whenever a transaction of medical information is performed, a block that contains the transaction information is generated and connected continuously to other blocks, and the information is stored in a distributed manner at the medical institutions participating in the network. Many blocks are closely interconnected through the hash values. A hash value is data that is converted to a special text string of a fixed length in which the original data cannot be distinguished when the hash goes through the hash function. The transaction information is reflected in the hash value of the newly generated block. When the internal data of a specific block in the blockchain changes, the hash value automatically changes, which also affects other blocks. In this way, a blockchain allows data tampering to be easily detected. In the blockchain, the hash value is used to add the corresponding block to the chain, and the hash value of the previous block is recorded in the current block. As a result, a connected list is created in the form of chain. Therefore, in order to hack a specific block, it is necessary to tamper with other blocks connected to the block of interest, making forgery exceedingly difficult.
Code: Generation of block objects
To create a block object, a verified transaction is added to the chain, and the transaction information is reflected in the hash value of the block. A new block hash is then generated using the previous block hash. The hash value of the previous block is saved in the newly generated block that follows it.

Generation of blockchain lists using hash values that reflect transaction information

```
block = new Block(block.getBlockID()+1, block.getBlock-
Hash(), Util.getDate(), patientsData, new ArrayList < Transac-
tion > ());

// A transaction is generated to add to the medical record of pa-
tient 2673123.
transaction = new Transaction(key, patientsData.getPatientsDa-
ta());
block.addTransaction(transaction);

//The transaction information is reflected in the hash value of the
block.// If the transaction information in a block is changed, the
hash values of all subsequent blocks are also changed.
public String getBlockHash() {
//The block hash is created using the previous block hash.
return Util.getHash(getTransaction() + previousBlockHash);
}
// The SHA-256 hash value, which is returned as a text string,
passes through the function.
//When the value of the text string changes, the hash value also
changes.
//For the SHA-256 hash algorithm, the Avalanche Effect method
is applied.
public static String getHash(String input) {
StringBuffer result = new StringBuffer();
try {
MessageDigest md = MessageDigest.getInstance("SHA-256");
md.update(input.getBytes());
byte bytes[] = md.digest();
for(int i = 0;i < bytes.lengthi++) {
result.append(Integer.toString((bytes[i] & 0xff) + 0x100, 16).
substring(1));
}
} catch(Exception e) {
e.printStackTrace();
}
return result.toString();
}
```

SHA, secure hash algorithm.

The patient's medical information is stored in the generated block, and multiple blocks form a list that is connected through their hash values.

Blockchain list information

```
= = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
Block number: 0
Created time: 2021-06-14 01:51:36.640
Previous hash: null
Block hash: 74234e98afe7498fb5daf1f36ac2d78acc339464f-
950703b8c019892f982b90b
Medical record:
= = = = = = = = = = = = = = = = = = = = = = = = = = = = = =

A normal transaction was found.
= = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
Block number: 1
Created time: 2021-06-14 01:51:39.661
Previous hash: 74234e98afe7498fb5daf1f36ac2d78acc339464f-
950703b8c019892f982b90b
Block hash: 6345e8d4fe0b749dfe1862cbfd90817f38dd3692cd-
2cd918be58e5becd540944
Medical record:
NAME: Gil Dong Hong
AGE: 50
GENDER: M
STROKE: Lt. hemiplegia d/t Rt. CR infarct
ONSET: Jan/01/2021
MEP: Jan/12/2021: Rt. ABP 20.6/5600, Lt. ABP NR; Rt. TA
19.5/2200, Lt. TA NR
SEP: Jan/12/2021 Rt. Median 19.0/24.2, Lt. Median NR; Rt. PT
38.2/45.0, Lt. PT NR
Evaluation data: Jan/10/2021
Bathel index: 30
MMSE: 24
GDS: 2
MVPT: 30
MFT: .30/8
Purdue: 16/NT
Grip Power: .50/6
Monofilament: 3.22/5.22
Two point discrimination: .4/6
Shoulder abductor: 1
Elbow flexor: 2
Finger flexor: 2
(Continued to)
```

```
Finger extensor: 1
Hip flexor: 2
Knee extensor: 2
Ankle D/F: 1
FAC: 1
Aphasia type: conduction aphasia
AQ: 55
LQ: 48
Light tough: .20/12
Kinesthetic: .20/12
MBC: 2
GCS: 15
= = = = = = = = = = = = = = = = = = = = = = = = = = = = = =

A normal transaction was found.
= = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
Block number: 2
Created time: 2021-06-14 01:51:42.704
Previous hash: 6345e8d4fe0b749dfe1862cbfd90817f38d-
d3692cd2cd918be58e5becd540944
Block hash: dbf72d6bf7143b6272308fb0708291c10733c
0787553355753fb90e477113a7e
Medical record:
NAME: Gil Dong Hong
AGE: 50
GENDER: M
STROKE: Lt. hemiplegia d/t Rt. CR infarct
ONSET: Jan/01/2021
MEP: Jan/12/2021: Rt. ABP 20.6/5600, Lt. ABP NR; Rt. TA
19.5/2200, Lt. TA NR
SEP: Jan/12/2021 Rt. Median 19.0/24.2, Lt. Median NR; Rt. PT
38.2/45.0, Lt. PT NR
Evaluation data: Feb/10/2021
Bathel index: 52
MMSE: 27
GDS: 2
MVPT: 36
MFT: .30/10
Purdue: 16/NT
Grip Power: .50/8
Monofilament: 3.22/5.18
Two point discrimination: .4/6
Shoulder abductor: 2
Elbow flexor: 3
Finger flexor: 3
(Continued to the next page)
```

```
Finger extensor: 1
Hip flexor: 3
Knee extensor: 3
Ankle D/F: 1
FAC: 2
Aphasia type: conduction aphasia
AQ: 58
LQ: 52
Light tough: .20/14
Kinesthetic: .20/14
MBC: 3
GCS: 15
= = = = = = = = = = = = = = = = = = = = = = = = = = = = = =

A normal transaction was found.
= = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
Block number: 3
Created time: 2021-06-14 01:51:45.737
Previous hash: dbf72d6bf7143b6272308fb0708291c10733
c0787553355753fb90e477113a7e
Block hash: f55e598438047feeafac137016eda953396244cff-
fe148730499727203e72733
Medical record:
NAME: Gil Dong Hong
AGE: 50
GENDER: M
STROKE: Lt. hemiplegia d/t Rt. CR infarct
ONSET: Jan/01/2021
MEP: Jan/12/2021: Rt. ABP 20.6/5600, Lt. ABP NR; Rt. TA
19.5/2200, Lt. TA NR
SEP: Jan/12/2021 Rt. Median 19.0/24.2, Lt. Median NR; Rt. PT
38.2/45.0, Lt. PT NR
Evaluation data: Mar/13/2021
Bathel index: 58
MMSE: 28
GDS: 2
MVPT: 38
MFT: .30/12
Purdue: 16/NT
Grip Power: .50/8
Monofilament: 3.22/5.14
Two point discrimination: .4/6
Shoulder abductor: 3
Elbow flexor: 4
Finger flexor: 4
(Continued to)
```

```
Finger extensor: 2
Hip flexor: 3
Knee extensor: 3
Ankle D/F: 2

FAC: 3
Aphasia type: conduction aphasia
AQ: 58
LQ: 54
Light tough: .20/16
Kinesthetic: .20/16
MBC: 4
GCS: 15
= = = = = = = = = = = = = = = = = = = = = = = = = = = = = =

A normal transaction was found.
= = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
Block number: 4
Created time: 2021-06-14 01:51:48.754
Previous hash: f55e598438047feeafac137016eda953396244cff-
fe148730499727203e72733
Block hash: 5fdfbc0a7d30f454dfc24975805e353ca5912b4d-
81c9875447e6a99644650a8f
Medical record:
NAME: Gil Dong Hong
AGE: 50
GENDER: M
STROKE: Lt. hemiplegia d/t Rt. CR infarct
ONSET: Jan/01/2021
MEP: Jan/12/2021: Rt. ABP 20.6/5600, Lt. ABP NR; Rt. TA
19.5/2200, Lt. TA NR
SEP: Jan/12/2021 Rt. Median 19.0/24.2, Lt. Median NR; Rt. PT
38.2/45.0, Lt. PT NR
Evaluation data: May/01/2021
Bathel index: 60
MMSE: 28
GDS: 2
MVPT: 40
MFT: .30/12
Purdue: 16/NT
Grip Power: .50/8
Monofilament: 3.22/5.14
Two point discrimination: .4/6
Shoulder abductor: 3
Elbow flexor: 4
(Continued to the next page)
```

```
Finger flexor: 4
Finger extensor: 2
Hip flexor: 3
Knee extensor: 3
Ankle D/F: 2
FAC: 4
Aphasia type: conduction aphasia
AQ: 58
LQ: 56
Light tough: .20/16
Kinesthetic: .20/16
MBC: 4
GCS: 15
= = = = = = = = = = = = = = = = = = = = = = = = = = = = =
```

MEP, motor evoked potential; ABP, abductor pollicis brevis; Rt, right; Lt, left; TA, tibialis anterior; PT, posterior tibial; NR, no response; SEP, sensory evoked potential; MMSE, mini-mental state examination; GDS, global deterioration scale; MVPT, motor-free visual perception test; MFT, manual function test; NT, not testable; D/F, dorsiflexor; FAC, functional ambulatory category; AQ, aphasia quotient; LQ, language quotient; MBC, modified Brunnstrom Classification; GCS, Glasgow Coma Scale.

Discussion

In this study, we created a simple blockchain for the purpose of allowing hospitals to exchange patient information on a network in a way that makes hacking practically impossible. The proposed information sharing method only allows the medical staff of a hospital to see a patient's information stored on the network in the form of a blockchain if there is a private key stored on the patient's smartphone. This method circumvents the inconvenience of the current system in which patients must print out their medical records and deliver them to the new hospital when being transferred. Clinicians may be concerned that due to its decentralized nature, if blockchain is used in the medical field, patients will become the guardians of their medical information. Nevertheless, in the proposed system, the patients do not directly possess their own medical information. Hence, we expect that the aversion of doctors to information sharing via blockchain will be small.

This study only included the information that is required by clinicians to treat stroke rehabilitation patients in the blockchain. The information was based on the paper by Kim et al. [7] published in 2021. Through a Delphi study, they collected the essential medical information of stroke rehabilitation patients when transferred to another hospital from 31 physiatrists. They found that the degree of muscle strength at major joints; a brief cognitive function test; and hand, language, and sensory function were essential information for treatment. We, therefore, added this essential information of stroke patients to the blockchain, and the information was stored.

This is the first study to demonstrate the possibility of using blockchain for the storage and delivery of the patient information of stroke patients by storing the information in a blockchain. The code that we wrote may not be suitable for direct implementation in the medical field, but it can serve as a foundation for researchers to create a blockchain system that can be actually used in the field based on future research results.

Notes

Conflicts of interest

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

Funding

The present study was supported by a National Research Foundation of Korea grant funded by the Korean government (grant no. NRF-2019M3E5D1A02068106).

Author contributions

Conceptualization, formal analysis, investigation, supervision, Writing-original draft, wiring-review & editing: CMC.

ORCID

Min Cheol Chang, https://orcid.org/0000-0002-7629-7213

References

1. Chang MC, Hau YS, Park JC, Lee JM. The application of blockchain technology in stroke rehabilitation. Am J Phys Med Rehabil 2019;98:e74.
2. Kuo TT, Kim HE, Ohno-Machado L. Blockchain distributed ledger technologies for biomedical and health care applications. J Am Med Inform Assoc 2017;24:1211–20.
3. Chang MC, Hsiao MY, Boudier-Revéret M. Blockchain technology: efficiently managing medical information in the pain management field. Pain Med 2020;21:1512–3.
4. Dutta P, Choi TM, Somani S, Butala R. Blockchain technology in supply chain operations: applications, challenges and research opportunities. Transp Res E Logist Transp Rev 2020; 142:102067.
5. Karandikar N, Chakravorty A, Rong C. Blockchain based transaction system with fungible and non-fungible tokens for a community-based energy infrastructure. Sensors (Basel) 2021;

21:3822.

6. Fang HS, Tan TH, Tan YF, Tan CJM. Blockchain personal health records: systematic review. J Med Internet Res 2021; 23:e25094.

7. Kim JK, Hau YS, Kwak S, Chang MC. Essential medical information for stroke patients undergoing interhospital transfer: a Delphi study. Am J Phys Med Rehabil 2021;100:354–8.