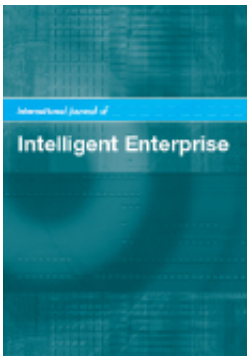


International Journal of Intelligent Enterprise

ISSN online: 1745-3240 - ISSN print: 1745-3232
https://www.inderscience.com/ijie

Information security protection for eHealth records using temporal hash signature

R. Charanya, R.A.K. Saravanaguru, M. Aramudhan

DOI: 10.1504/IJIE.2021.10039424

Article History:

Received:	06 November 2018
Last revised:	05 January 2019
Accepted:	17 February 2020
Published online:	30 November 2022

Information security protection for eHealth records using temporal hash signature

R. Charanya*

School of Information Technology and Engineering,
VIT University,
Vellore Campus,
Tamil Nadu, India
Email: charanya.r@vit.ac.in
*Corresponding author

R.A.K. Saravanaguru

School of Computer Science and Engineering,
VIT University,
Vellore Campus,
Tamil Nadu, India
Email: charanya.r@vit.ac.in

M. Aramudhan

Department of IT,
Perunthalivar Kamarajar Institute of Engineering and Technology,
Karaikal, India
Email: aranagai@yahoo.co.in

Abstract: Patient health record information management becomes an important and challenging task through the different practices followed by hospitals. Also, the risks associated with securing the data from cyber-attacks and data breaches become inevitable. Ransomware is the biggest cyber-attack in the history of National Health Service (NHS) which has affected nearly 100 countries worldwide in 2017. In the healthcare system, loss of sensitive data leads to embarrassment whereas loss of integrity leads to loss of the patient's life. Though many technical solutions exist to store the electronic health record (EHR) information in a secure way, it is not sufficient to satisfy the security requirements such as clear attributes for role-based access, common regulations that protect patient's privacy and specific guidelines to control the data. This paper provides a brief discussion on the existing security mechanisms with a proposal based on the temporal shadow in the cloud. The integrity of the patient's document is modelled with linked records and verified by temporal signature.

Keywords: eHealth; electronic health record; EHR; binary Merkle tree; temporal hash signature.

Reference to this paper should be made as follows: Charanya, R., Saravanaguru, R.A.K. and Aramudhan, M. (2023) 'Information security protection for eHealth records using temporal hash signature', *Int. J. Intelligent Enterprise*, Vol. 10, No. 1, pp.14–30.

Biographical notes: R. Charanya received her Bachelor of Engineering in Computer Science and Engineering from Anna University, in 2006, and Master of Engineering in Software Engineering from Anna University, in 2008 and PhD from Vellore Institute of Technology, Vellore, 2019 in the field of securing the eHealth System in Blockchain. She is currently associated with the Vellore Institute of Technology (VIT), Vellore since 2010, and presently working as an Assistant Professor in the School of Information Technology and Engineering (SITE). Her area of interest includes cloud computing, blockchain, software engineering. She published more research papers and conference papers in reputed journals.

R.A.K. Saravanaguru received his Bachelor of Engineering in Computer Science and Engineering from the Madras University in 2000 and received a Master of Technology in Advanced Computing from SASTRA University, in 2002. He received his Doctor of Philosophy in Computer Science and Engineering at Vellore Institute of Technology in 2013 in the field of Context-Aware Middleware for the Vehicular Ad-Hoc Network. He has been associated with the Vellore Institute of Technology (VIT), Vellore since June 2004, and presently working as an Associate Professor in School of Computer Science and Engineering (SCOPE) and Assistant Dean Academics. He has 16 years of teaching experience. His area of interest includes context-aware systems, middleware, web services, VANET, and data science.

M. Aramudhan received his BE in Computer Science and Engineering from the Regional Engineering College, Trichy in 1997. In 2001, he completed his ME in Computer Science and Engineering from Regional Engineering College, Trichy. He received his Doctor of Philosophy in Computer Science and Engineering at Anna University, Chennai in 2008. He is currently working as an Associate Professor in the Department of Information Technology at Perunthalivar Kamarajar Institute of Engineering and Technology since 2009. His area of interest is computer networks, web technology, operating system, data structure, programming languages, DBMS. He published more journals and conference paper in the reputed journal.

This paper is a revised and expanded version of a paper entitled ‘Information security protection for eHealth records using temporal hash signature’ presented at International Conference on Advanced Computing and Big Data Analytics-ICACB’18, Kingston Engineering College, 23–24 March 2018.

1 Introduction

By using modern communication infrastructure, healthcare services under the term eHealth. The eHealth system is used to provide medical services over the internet. By using cloud computing technique, the doctor, patient, and government, establish a heterogeneous network to improve the health service. All the patient information is stored in the distributed cloud every day so that we can access the patient details from anywhere and at any time. It provides heterogeneous communication between doctor, patient, nurse, and pharmaceutical. A secure healthcare system is needed to store the health record safely, effectively and inexpensively. By using modern communication technology the quality of healthcare service is improved by lowering the capital and operation cost (Cheong et al., 2009). Hence the government of different countries like Korea, Japan,

UK, Canada, USA and European union shift from traditional healthcare services to the eHealthcare service (Dzenowagis and Kernen, 2005; Canada Health Infoway, 2009). The main advantage, it brings all doctors and patient in one network. The developing and underdeveloped countries like Myanmar, Bangladesh, India have not used the eHealth service yet (Federal Health IT Initiatives, 2009; IDE Deliverable 2.1.4 European Good Practices, 2010). The key challenge in eHealth information, stored in a different form and giving proper access rights. Patient details are very sensitive information, so we need to protect them from an unauthorised user. eHealth systems are facing a lot of privacy and security issues (Li et al., 2011; Charanya et al., 2013) and explained in Table 1. There is a need to protect the highly sensitive healthcare information over the cloud.

Security is the major challenge in de-centralised systems (Charanya and Aramudhan, 2016). Availability, confidentiality and data integrity come under the properties of security. We considered the main parameter as security, in existing systems several algorithms proposed related to public key cryptography which involves trusted third party send key before exchange the data. In this work, we propose a new framework using temporal signature for securing healthcare records of the patients without a third party in a cloud environment.

Table 1 Summary of the literature survey

Algorithm	*Who, when*	*Pros*	*Cons*
Attribute-based encryption	Li et al. (2013)	Patient able to manage all own data	Access rights based on attribute only
RSA	Raya and Biswasb (2012)	Authentication and confidentiality	Brute force attack
Digital signature algorithm	Li et al. (2011)	Achieve authentication, non-repudation	Loss of private leads to severe damage
Elliptic curve cryptography (ECC)	Lee et al. (2014)	Computation cost reduced	Complex to implement
Public key cryptography	Huang and Liu (2011)	Security	Trust the third party
Key policy attribute-based encryption	Han et al. (2012)	Confidentiality	Data owner have limited control
Ciphertext policy attribute-based encryption	Zhou et al. (2015)	Data owner have full rights	The combined attributes form a single set issued to satisfy policies
BOAT algorithm	Vaid and Verma (2014)	Better accuracy	'Big data' not considered
Identity-based encryption	Tseng et al. (2016)	Reduce the complexity of encryption process	PKG need to be online
Role-based access control	Zhou et al. (2013)	The setup and managing is easy	It validate partial ordering relation among these keys are required

In this work, still, keys are applied for authentication and verification of the signature. It is a scalable digital signing-based authentication for electronic data and it is verified. Simply temporal signature is used to prove the data integrity and detect changes in data

authenticity. It uses formal mathematical methods to authenticate any type of data. Mainly temporal signature is used for speed, security, and scalability.

The temporal signature technique is implemented in the healthcare system with respect to store and verifies the digital assets independently in real-time. The proposed system achieves confidentiality, data integrity and especially transparency between the data owner and service providers. This work derived mathematically for links of all the files in time and creates root as a hash value in the top using Merkle tree. The purpose of using this tree is to provide the integrity and validity of the data and maintain its consistency of the data.

1.1 Objective

To provide proper authentication to access the eHealth system using temporal signature infrastructure.

This paper is organised as follows: Section 1 is introduction, Section 2 discussed the related work, Section 3 describes the Proposed Work, Further, Section 4 discussed about the proposed work to secure the healthcare system in blockchain, Section 5 illustrates the results and discussion, Section 6 discussed security and performance analysis, Section 7 concluded the system with future works.

2 Related work

2.1 Rivest-Shamir-Adleman algorithm

It is used for key exchange with variable key size and variable encryption block. This method is used to ensure the authenticity and confidentiality of the healthcare data. Here two keys are used public key is used for encryption which is kept in public, and the private key is used for decryption which is kept secret. This algorithm has several drawbacks like little resistance to brute force attacks, timing attacks, mathematical attacks, side channel analysis attacks, and adaptively chosen cipher attacks. Similarly in the decryption process, by using several known ciphertexts, the attacker can measure the decryption time, thus, can deduce the decryption key quickly. In the decryption process, Chinese remainder theorem is used which is further extended to Gaussian integer for better efficiency (Raya and Biswasb, 2012).

2.2 Digital signature algorithm (DSA)

Digital signature is used to sign electronic reports to validate the authenticate person and approve to view the message in electronic format. By using this we achieve integrity, non-repudiation, authentication, and identification. To give a safe secure electronic signature scheme, the following traits must be fulfilled like user identification, PINs, and passwords. The National Institute for Standards and Technology (NIST) is approved the digital signature standard. The user has to obtain both private and public key for using the digital signature (Li et al., 2011). The user's loss of the private key leads to cause severe damage.

2.3 Elliptic curve cryptography

Its new smart card-based key administration scheme for HIPAA protection and security direction in elliptic curve cryptography (ECC) (Lee et al., 2014). ECC comes under public key cryptography, so, it has a public key and private key. The advantage of ECC is small key size 160-bits key is considered which is as secure as 1,024-bits key in Rivest-Shamir-Adleman (RSA) and the computation cost is also reduced. It is more complex and difficult to implement than RSA, which increases implementation error and also reduce the security of the algorithm (Lee et al., 2014).

2.4 Public key cryptography

Public key cryptography key uses two different keys in the eHealth record, one key is utilised to encode the health information and another key is utilised to decode the health record. Each patient or user use two different keys for encryption and decryption (Huang and Liu, 2011). The famous two public key algorithms are RSA and Diffie-Hellman algorithms. Its drawback is that it has to trust the third party to certify the public keys.

2.5 Attribute-based encryption

By using attribute-based encryption (ABE) the patient encrypts the electronic health information with the set of attributes and the decryption depends on these attributes and some access control policies. To reduce the key management complexity the users were split into different security domain (Li et al., 2013). During the emergency situation change the access policies or file attributes. Separation the user area into various security domain to diminish the key administration multifaceted nature for patient and clients User emergency situation, access policies or file attribute will change. The ABE is not much efficient. The two types of ABEs are key policy attribute-based encryption (KP-ABE) and ciphertext policy attributes-based encryption (CP-ABE).

2.6 Key policy attribute-based encryption

The patient encrypts the health records with a set of attributes and stores in the cloud. The secret key is issued by the trusted authority which joined with access structure like a tree, which describes user identity (Han et al., 2012). Each person's access policy is different. All the data is not obvious to everybody. According to the role, the data will be visible. During decryption, if the private key is satisfied with attributes, then the user can decrypt the ciphertext. The main drawback is that the data owners have restricted control over who can decode the information. Another issue is re-encryption, i.e., the private keys should be re-issued to all users.

2.7 Ciphertext policy attribute-based encryption

Each user is depicted with a set of attributes and the private key is distributed by a trusted authority. Based on the attributes the permission is given to the user to decrypt the data. Here encryptor indicates the policy and dependent on that user can decrypt the data (Zhou et al., 2015). During the decryption process if the attribute of a user fulfils the access policy of the ciphertext, then the user can decrypt the ciphertext.

2.8 Integrity based on RSA partial homomorphic and MD5 cryptography

The patient data is initially encrypted using the RSA algorithm and then outsourced to the cloud. Hashing is the second step. On the encryption of the file, a key pair public and private is generated. The hashing of this encrypted data is only done once it reaches the cloud. The message digest-5 algorithm is used for this purpose. A replica of the hash obtained from the file is sent back to the patient or the concerned authority for any further verification process (Ora and Pal, 2015). The verification also takes place at the cloud and is fairly simple. The client generates a verification request. On receiving this request, the cloud recomputes, the hash of the file of that particular client. After this, a cross-verification is done to check if the file has been altered or not. But, a huge concern for this system is the secrecy of the private keys, which, once breached, makes the whole system vulnerable.

2.9 Trust-based intrusion detection in cloud

Each client in a cloud environment has its own concerns based on its level of risk. Depending on the needs and requirements of different users of different cloud services the trust level of each host is identified. There can be a high risk, medium risk, and low-risk users (Salek and Madani, 2016). Different rule-sets are applied while dealing with different type of clients. All the different kinds of intrusion detection systems lie within the same cloud architecture on different types of hosts. A global agent gathers alerts from all these IDSs to analyse and find correlation among different alerts. A trust evaluation model is established and the risk level of a user can be decided on the basis of the degree of abnormality in its behaviour. However, the system is not sophisticated enough to identify new and unknown attacks.

2.10 BOAT algorithm

This approach detects the anomalies in the cloud by comparing the data gathered from cloud environments to a standard set of correct data. It follows an anomaly-based approach in which the system is trained with the existing correct dataset and capture any deviations from the normal behaviour (Vaid and Verma, 2014). Two-stage intrusion detection is used. First thing this system does is to collect the data from the cloud network and then compare it with the given correct dataset. Here it opaquely tries to discover an intrusion, if any, may have occurred. K-means clustering is used to group similar kinds of attacks together. In the second stage, the deviations are scrutinised for their behaviour and classification is done and represented as a decision tree. The BOAT algorithm allows this system to dynamically alter the decision trees in case the training dataset varies dynamically.

2.11 Identity-based encryption

The doctor knows the unique information of patient like SSL, electronic healthcare system number, etc. then doctor pass the health information to the patient and it can read only by the patient. In cryptographic ideas, the doctor passes the health information to the patient, he signs it with the secret key and encrypt the health information with a patient

unique ID and send it to the patient. While encryption, the patient decrypt the health information by utilising the private key, at that point confirms the mark by utilising specialist name and address as the check key.

2.12 Role-based access control encryption

The patient encrypts the health data and stored in the cloud. Relies on the job they can access health information. In role-based, roles represented with hierarchies. Each user assigned with a set of roles, based on the role, the access permission given. Once the user leaves the group then he is not eligible to view the health information.

3 Proposed work

The proposed work focus on healthcare security system in a cloud environment which provides more data confidentiality of patient's healthcare records without third-party control. It achieves the following objectives: data validation and data integrity proposed work. The doctor, before storing the patient's health information in the cloud, encrypts it using the RSA algorithm and then forwards it to the EHR system. Each encrypted health record is appended with a temporal shadow and then the hash value is calculated using a Hashing algorithm. Here we used a SHA 256 algorithm for creating hash signatures. The top root value is stored in the database.

3.1 Basic components

3.1.1 Doctor

The doctor enters the patient's health information in the EHR system. For, e.g., doctor A works in Apollo hospital, he enters the patient details (patient name Ram) in EHR system. To get the treatment, patient Ram has already visited Apollo hospital many times. Patient Ram's health history is maintained in the local database of Apollo hospital and is used for future references. If the patient is interested in getting treatment from other hospitals, records can be encrypted with the patient's public key and stored in the cloud.

3.1.2 Patient

Before deploying the encrypted health information in the cloud, the patient gets a notification for acknowledgment. The patient decides who should access which data. Each patient record is represented with a unique ID like SSN or Aadhar number, etc.

3.1.3 Local DB

Initially, electronic health information is stored in the local database so that, doctors can access the patient's health information whenever needed. Each patient record is represented with a unique ID for easy access. Health information is stored without encryption. For easy reference in the hospital, the patient's health record is stored locally.

3.1.4 Root server

The signed encrypted health data is converted into the hash of health data is given to the registry server. To get the signature token as the evidence of the temporal signature, previous hashed health record, etc. no keys are expected to make a signature token. It maintains an accurate clock timing like in time, time out, and in-between time, so it is easy to track them when they started the process, and when its get completed and in-between time it takes automatically calculated. The hashed leaf node is again hashed and forms a parent hash node. The root server which maintains the root values and returns back the signature token to the user.

3.1.5 Linker server

The linker server is responsible for getting the request from the lower server and building the hash tree and then sending it to the upstream server. The parent hash accepts requests only from the authenticated child node. Linker conveys the response to all child linker with the hash way of its own tree.

3.1.6 Registry server

It is used to receive the hashed value from the user also used to send a response to the customers and also receive requests from the user. The registry acts as an initial linker and sends the request to the upstream linker. EHR systems send a hash of patient data to the registry server.

3.1.7 Cloud

The request is received from the doctor, initially, it is checked in the cloud to verify the authenticity of users. It maintains the details such as patient ID, access permission, attributes of the health database, signature token, etc. The doctor requests to access the record by giving the valid patient's ID and patient signature token. The cloud verifies the authenticity of the user by sending the session key to the user. The EHR System checks the received session key with the cloud. The next level of security, the registry server ask the user to enter the signature token, it is checked with registry server. Now the authenticated user can access the record from the cloud.

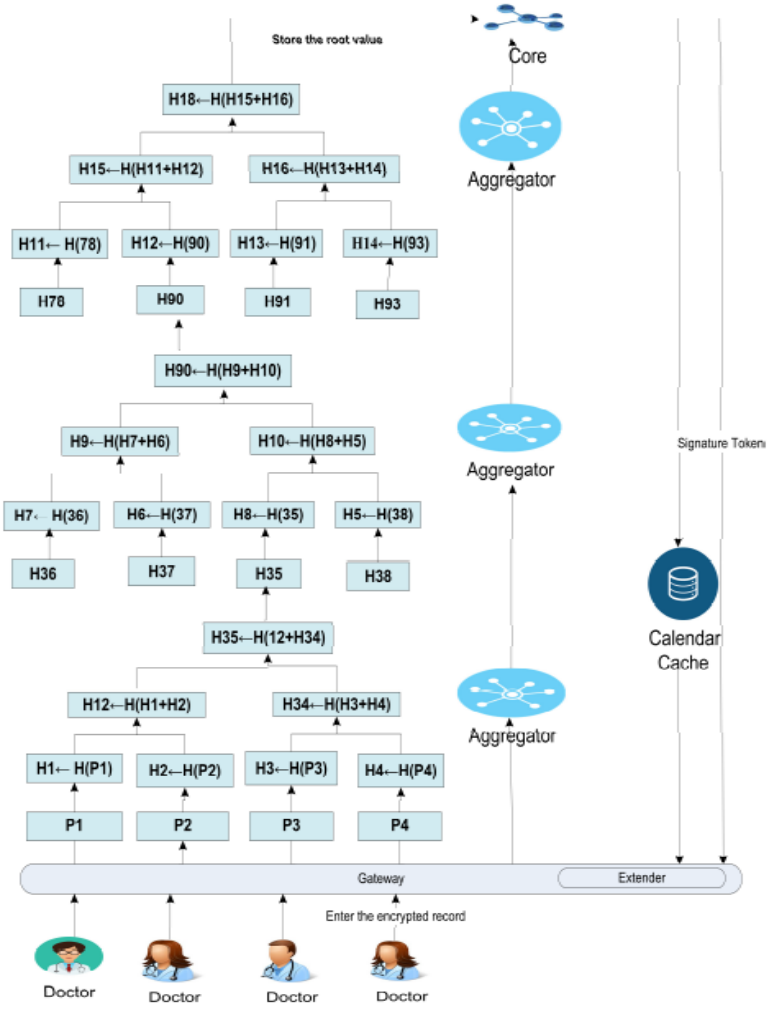
4 Proposed work to secure the healthcare system in blockchain

For both data owner and service providers, it provides transparency and also reduces the risk for a most valuable asset. It is impossible for the attacker to attack the system. Asymmetric cryptography is used for signer identification and cryptography is used to protect the data. Using temporal signature many health records are signed at a time. In this system, the user enters the hash value and also provides the signature for cryptographic proof so that they can access the data (Ora and Pal, 2015). The main advantage of the proposed work is providing authentication, Time of signature and verification of the origin. In healthcare records, loss of honesty leads to loss of the patient's life. The temporal signature is explicitly for confirmation and checks the

integrity of medical records and is also maintained. The temporal signature infrastructure is introduced along with to secure the integrity of the patient data which provide secure authentication without any help of trusted authority.

Doctor encrypts the patient's electronic health record (EHR) before storing it in the cloud. Doctor encrypts the patient health record using the patient public key. The signed data is saved, which can be utilised later to check the signing time, signing entity and data integrity. Data is received in the proper format. The doctor should register in the cloud to become authorised cloud user. The EHR system converts the encrypted EHR into the hash of documents.

Figure 1 Secure the eHealth system temporal hash signature (see online version for colours)



In Figure 1 registry server receives the hash of document which is appended with temporal shadow, and by utilising the linker, hash values are accumulated and sent to the next forthcoming server. Generating numerically inferred record called hash and another document made in the similar time, combining both the records (hashes) called as hash

tree. The files created or modified in the time increment, by using cryptographic links, top roots hash is generated and it is used as proof that shows the contribution of every file. Linker forms the hash tree, that is sent to the next upcoming server (root). The root hash value is stored in the calendar database.

The hash path is stored with the document so that it is easy to check the trustworthiness and time of the document. Per-round global hash trees are created by a hierarchy of aggregation servers. The first layer is a registry server which is directly collecting requests from clients, each linker server receives a request from lower level servers, hashes them together and forms a hash tree. The top hash value is sent to the higher-level servers.

Steps to upload eHealth record uploaded in cloud

- The patient eHealth record converted into an encrypted record.
- Store the encrypted health record with patient ID and access information stored in the cloud. The system sends intimation message to the patient for acknowledgment.
- The patient specifies the access permission to access the health record by whom and what. Patient family members are allowed to read the patient record, whereas doctors are allowed to write it in the health record.
- Now the encrypted patient record is converted into the hash record. Each record is appended with temporal shadow.
- Registry server receives the hash of record and forward it to linker. The hash of record is converted into the hash tree and top hash value stored in the cloud and calendar database.
- The signature token is generated and its send to the user for authentication purpose.

Steps to download the record

- Doctor access the patient record by giving the patient ID and patient private key.
- The authenticated doctor request is forwarded to the registry server.
- Request to enter the signature token. After validation, the cloud sends the session key to doctor registered mobile or e-mail ID.
- Once validated, the authenticated user can access the file from the cloud.

4.1 Framework functionalities

4.1.1 Temporal signature

In the digital signature, keys are used, the public key is the root value of the tree, the private key is the signature token of the user, based on the path from leave node to the root node. It is a scalable digital signing-based authentication for electronic data. To verify the integrity of the patient health information, the cryptographic key is used. Simply temporal signature is used to prove the authenticity of health record (Emmadi and Narumanchi, 2017). The temporal shadow means considering the in time and out time and in-between time. It uses formal mathematical methods to authenticate any type of

health information. Mainly temporal is used for speed, security, and scalability (Nielson and Gollmann, 2013).

Table 2 Pseudocode for SigGen

Pseudocode for SigGen
Parameter:
r: patient health record
n_i: nonce
ts_i: temporal shadow
P_{id}: patient identity
z_{i-1i}: previous hash value
Input: Set of medical record received from the user
Output: Signature token S_t
Public ledger $z_i = H(z_{i-1}, n_i, ts_i, r_i)$
Process:
1 Health record 'r' in proper format.
2 Convert the original health record into hash of data h(r) by using SHA 256. Public ledger $z_i = H(z_{i-1}, n_i, ts_i, z_{ii})$
3 To create the signature for the hashed record 'r_i' and the request is send to signature server with machine ID, patient identity P_{id}
4 Binary hash tree for the health records generated
5 The signature S_t for record 'r' is (P_{id}, siblings, root)

The following steps are involved to provide the secure authentication of the health data.

a SigGen

b Hash Tree form

c SigVerfify

d SigToken.

The encrypted health record is converted to hash values. The eHealth system takes the encrypted record from the user and converts into an alphanumeric string called 'hashing'. The input size is same regardless of the size of the input. The changing the single character reflects the changes in the output hash with the same size.

a SigGen for hashed document

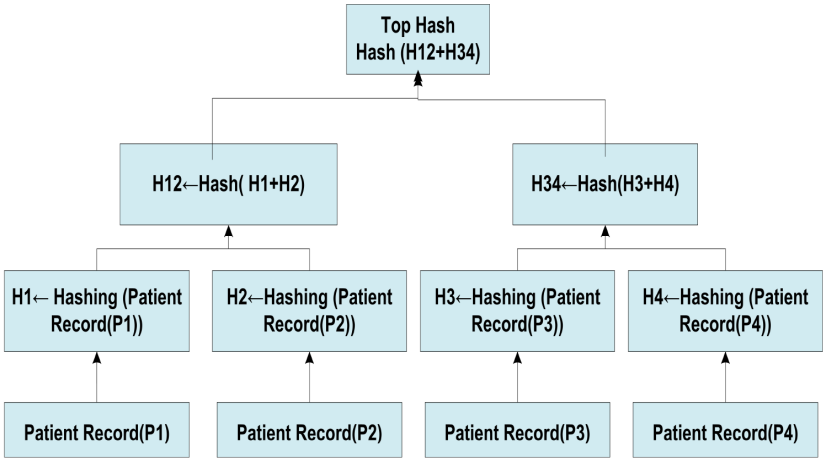
First, the patient health information is encrypted and the encrypted information is converted into hash values by applying hash function and a signed hashed document is sent to the server. By using hash algorithms SHA 256, health records are converted to hash values. In Table 2 shows the signature is generated for a hash document by using the identity of the user like client ID, time (at what time request is sent). The doctor sends the signed hash of health data to the registry server (https://m.guardtime.com/files/KSI_data_sheet_201509-2.pdf). The registry server accepts the hash of patient health record and forms hash tree. Top hash values in hash trees stored in the hash calendar. The signature token is generated based on the

path traversed from the top hash value to the hash value of the leaf node. It automatically generates the signature token to the user. It contains the token for reconstructing the path through the hash tree. It is the user's responsibility to keep the signature token safe. By using a signature token, the user can easily identify the transaction and at particular time. By using Merkle tree, authenticate the large set of data. It is possible to use cryptographic hashes without Merkle trees, but it is extremely inefficient and not scalable. In terms of scalability, it accepts the 'n' number of hashed patient record at a particular time and not depend on the transaction.

b Hash tree for medical record

The doctor sends a hash of patient data to the registry server. Hash of various patient information is collected to form the hash tree and the root value is sent to the root server. The linked server aggregates all requests into the hash tree and top hash values are retained for each second. Once the top hash value is identified, it is combined with the previous top hash values in the hash calendar. The repeatable steps of combining the hashes are well defined. Combining all the artefacts for a particular time can be summarised on a publication code.

Figure 2 Merkle hash tree (see online version for colours)



The path used to move from initial to publication code is based on the temporal signature. The new leaves are added just to the other side of the tree. A forged file or altered version is identified by the hash of file under test. This file is starting hash and processing through Merkle tree using temporal signature and publish results. If the process generates the same publication result, identical to the original file, then it is successfully authenticated. By running hash function, it verifies the authenticity of the information and compares the outcome and the one that is stored in the calendar database.

c SigToken

The signature token contains a way through the hash tree-beginning to the top hash value. It is the user's duty to keep the signature token safe. By using a signature

token, the user can easily identify if a transaction was in a particular location at a particular time. The signature checks the integrity of the healthcare data even if the user does not monitor it.

d SigVerify

The authentication is done by using the hash of file under test. To verify the authenticity of the digital file by the original signature is compared with the registry server which is explained in Table 3. If the process generates same code, identical to the original file, then it is an authentic user, else it is a forgery or altered version of the file. The hash is a one way function, no mathematical process can recreate the file from a hash.

Table 3 Verification

Pseudocode for verification parameter:
X: root value
r_i: patient health record
ts_i: time stamp
C_{id}: client identity
Input: Signature token
Output: Patient record
Process:
To verify signature <C_{id}, root, $siblings_i$> with root value which is in publication file.
Ensure the Signature with time, sibling, machine id, client id, root value.
Check the accurate registered signing time.

5 Results and discussion

During message exchange, time is captured for each message sent to the server. The temporal signature is generated before hashing the record. The hashing is performed using SHA 256 algorithm. It produces a 32 byte string, when you concatenate the 32 byte string with 32 byte string, it creates a 64 byte string which is again hashed to a 32 byte value until all transaction are hashed and joined up.

5.1 Append time with record

The message is appended with time and is measured in seconds, minutes, hours and microseconds. Using this temporal shadow the record is uniquely identified with the user. The patient record is stored in another file. That file is given as input. Figure 1 shows the time appended with the encrypted record. This is generated before hashing the record. So, each time hash values vary because of the appended temporal shadow. So security also increases.

5.2 Generating the hash chain

The hash chain is generated by applying hashes one after another.

$$HC = h\left(h\left(h(x)\right)\right)$$

Figure 1 shows the output changes when executing the same message again and again with the appended temporal shadow.

5.3 Formation of Merkle tree

The hashes are used to generate Merkle tree. The leaf nodes are hashed and grouped into two, and the new hash is generated. The temporal shadow is created, appended with each record and converted to the hash values and to form the Merkle tree. The same procedure is repeated until we get the single hash value as at the top root of the tree.

6 Security and performance analysis

The three properties that need to be followed by a cryptographic hash function(r) are as follows.

1 Pre image resistance: Given y it is difficult to find r with h(r) = y.

2 Second-pre-image resistance: It is difficult to find r2 for the given r1 with h(r1) = h(r2).

3 Collision resistance: It is difficult to find r1 and r2 such that h(r1) = h(r2).

The hash function is defined the computational complexity is hard, the collision will be there but not always. According to pre-image resistance, hash is broken into h1 and h2. For the given y of the hash h, it returns r1. To calculate the pre-image for y under h2, an algorithm running on h1(y) gives r1. Similarly, in h2(r1) = y, r1 is the pre-image for y under h2, else it forms a collision (Zhou et al., 2013).

Second pre-image resistance: the hash h is strong as inner function h2, by giving r1, such that h2(r1) = h2(r2). In other words, given r1, such that h(r1) = h(r2) then second pre-image collision on h2 for collision on h1.

To avoid this attack in Table 5, we use temporal shadow techniques, so that data cannot be easily compromised.

Table 4 Breakage effect in SHA 256

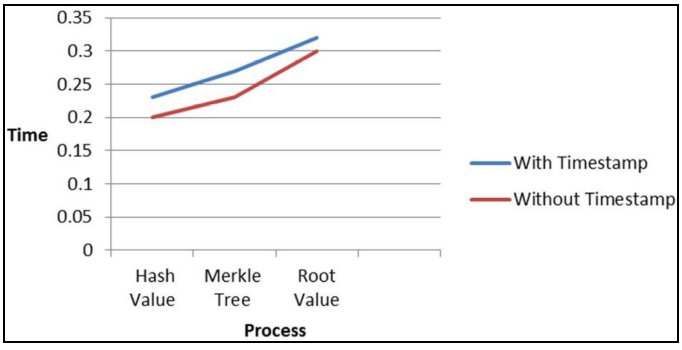
Attacks	*Steel data*
Pre-image	Complete breakdown
Second pre-image	Twofold spend

Table 5 Compare with temporal shadow and without temporal shadow

	With temporal shadow	*Without temporal shadow*
Hash value	0.23	0.14
Hash tree	0.27	0.20
Calendar	0.32	0.30

Each patient's record is appended with the temporal shadow and hashed values form a Merkle tree. The same procedure will be done until we get the root value of the Merkle tree. The main use of appending temporal shadow and without temporal shadow is mentioned in Table 4. By using this technique the system is made more secure. Figure 3 shows how security and performance increase in the Merkle tree with temporal shadow. By using the temporal shadow in Merkle tree, the following security-related benefits we achieved.

Figure 3 Security increases with time to build Merkle hash tree with and without temporal shadow (see online version for colours)



6.1 *Confidentiality*

The health records are encrypted and secured with the public key and each record has a digital signature. It allows the authentic users to access the health record by verifying the digital signature. The patient health records and transactions are secured with temporal shadow and cryptography.

6.2 *Integrity*

In distributed ledger, the patient health record are hashed and stored in the database and there is no way to change the hashed record because each record is linked with the previous records in time. It is impossible to change the records.

6.3 *Non-repudiation*

Once patient records are stored, then it is not possible to change the record. The digital signature is created for each record and is sent to the user for authentication purpose. It is the user's responsibility to keep the signature safe.

6.4 Authentication

Authentication is another benefit in the temporal signature. For each record, a signature token is created and it is given to the user. Each authentic user is evaluated by the signature token.

Table 6 Compare with the temporal shadow and without temporal shadow

Characteristic	*Record proof extraction*
Per-record storage	O(log n)*
Time	O(log n)
Memory	O(log n)

6.5 Testing

Testing was done in java and modules were implemented for checking the different and same messages with the temporal shadow for their integrity. Merkle tree gives more security between the peers. Adding the temporal shadow with the message is requires logarithmic space and time. Time taken to build Merkle tree which takes time and space complexity less than O(log n) and * asymptotically it's O(log N) in Table 6.

A comparative analysis of the proposed framework with RSA. Two attributes have been considered:

- time
- integrity.

7 Conclusions

Data integrity is essential for eHealth cloud. In this paper, we discussed how our proposed technique is more effective compared to previous techniques. Authentication and integrity is an important factor for eHealth cloud. In this paper, we constructed a framework to secure the eHealth system using temporal signature. By using, secure the EHRs without thrusted third party. In this paper, the encrypted patient record is given as input, and it is converted to hash value using SHA 256. Temporal shadow is appended with each hash values. The master root hash is generated to maintain integrity. For security purpose signature token is generated and send to the user for authentication purpose.

As per our proposed architecture, future work is to store the root hash value in the blockchain. Each second, leaf nodes are added to the Merkle tree and a final root hash value can be stored it in the blockchain so that we can avoid the attack of Merkle tree while storing the root value on the blockchain.

References

[online] https://m.guardtime.com/files/KSI_data_sheet_201509-2.pdf.

Canada Health Infoway (2009) [online] http://www.infoway-inforoute.ca (accessed June 2009).

Charanya, R. and Aramudhan, M. (2016) 'Survey on access control issues in cloud computing', *IEEE Conference Proceedings of ICETETS*, pp.237–240.

Charanya, R., Aramudhan, M., Mohan, K. and Nithya, S. (2013) 'Levels of security issues in cloud computing', *International Journal of Engineering and Technology*, Vol. 5, No. 2, pp.1912–1920.

Cheong, H.J., Shin, N.Y. and Joeng, Y.B. (2009) 'Improving Korean service delivery system in health care: focusing on national eHealth system', in *Proc. of eTELEMED'09*, IEEE, pp.263–268.

Dzenowagis, J. and Kernen, G. (2005) 'Global vision, local insight', *Report for the World Summit on the Information Society*, World Health Organization Press.

Emmadi, N. and Narumanchi, H. (2017) 'Reinforcing immutability of permissioned blockchains with keyless signatures' infrastructure', *ICDCN'17*, ACM, Hyderabad, India, 4–7 January.

Federal Health IT Initiatives (2009) [online] http://www.hhs.gov/healthit (accessed June 2009).

Han, J., Susilo, W., Mu, Y. and Yan, J. (2012) 'Privacy-preserving decentralized keypolicy attribute based encryption', *IEEE Transactions on Parallel and Distributed Systems*, Vol. 23, No. 11, pp.2150–2162.

Huang, H.F. and Liu, K.C. (2011) 'Efficient key management for preserving HIPAA regulations', *Journal of Systems and Software*, Vol. 84, No. 1, pp.113–119.

Lee, Y.S., Alasaarela, E. and Lee, H.J. (2014) 'An efficient encryption scheme using elliptic curve cryptography (ECC) with symmetric algorithm for healthcare', *International Journal of Security and its Applications*, Vol. 8, No. 3, pp.63–70.

Li, M., Yu, S., Zheng, Y., Ren, K. and Lou, W.L. (2013) 'Scalable and secure sharing of personal records in cloud computing using attribute based encryption', *IEEE Transactions on Parallel and Distributed Systems*, Vol. 24, No. 1, pp.131–143.

Li, Z.R., Chang, E.C., Huang, K.H. and Lai, F. (2011) 'A secure electronic medical record sharing mechanism in the cloud computing platform', in *Proc. 15th IEEE Int. Sympo. Consum. Electron*., June, pp.98–103.

Nielson, H.R. and Gollmann, D. (2013) 'Secure IT systems', *18th Nordic Conference, NordSec 2013*, Ilulissat, Greenland, 18–21 October.

Ora, P. and Pal, P.R. (2015) 'Data security and integrity in cloud computing based on RSA partial homomorphic and MD5 cryptography', in *2015 International Conference on Computer, Communication and Control (IC4)*, IEEE, September, pp.1–6.

Raya, S. and Biswasb, G.P. (2012) 'Design of RSA-CA based e-health system for supporting HIPAA privacy-security regulations', *2nd International Conference on Communication, Computing & Security (ICCCS-2012), Procedia Technology*, Vol. 6, pp.954–961.

RIDE Deliverable 2.1.4 European Good Practices (2010) [online] http://www.srdc.metu.edu.tr/webpage/projects/ride/.

Salek, Z. and Madani, F.M. (2016) 'Multi-level Intrusion detection system in cloud environment based on trust level', in *2016 6th International Conference on Computer and Knowledge Engineering (ICCKE)*, IEEE, October, pp.94–99.

Tseng, Y-m. et al. (2016) 'Identity based encryption with cloud revocation authority and its applications', *IEEE Transactions on Cloud Computing*, Vol. 6, No. 4, pp.1041–1053,

Vaid, C. and Verma, H.K. (2014) 'Anomaly-based IDS implementation in cloud environment using BOAT algorithm', in *2014 3rd International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions)*, IEEE, October, pp.1–6.

Zhou, L., Varadharajan, V. and Hitchens, M. (2013) 'Achieving secure role-based access control on encrypted data in cloud storage', *IEEE Transactions on Information Forensics and Security*, Vol. 8, No. 12, pp.2381–2395.

Zhou, Z., Huang, D. and Wang, Z. (2015) 'Efficient privacy-preserving ciphertextpolicy attribute based-encryption and broadcast encryption', *IEEE Transactions on Computers*, Vol. 64, No. 1, pp.126–138.