TEST Engineering & Management

May-June 2020
ISSN: 0193-4120 Page No. 3521-3528

Securing Healthcare Data by Providing Identity Based Access Control and PKI System Using Time Domain Attribute

[1]D.Anuradha, [2]S.Bhuvaneswari
[1]Part Time Research Scholar, Department of Computer Science, Pondicherry University-Karaikal Campus, [1]anuradha.d@vit.ac.in
[2]Registrar, Central University of Tamilnadu, Thiruvarur, sbhuvaneswari@cutn.ac.in

Article Info
Volume 83
Page Number: 3521-3528
Publication Issue:
May-June 2020

Article History
Article Received: *19 August 2019*
Revised: *27 November 2019*
Accepted: *29 January 2020*
Publication: *12 May 2020*

Abstract

One of the most prominent barrier of public key infrastructure (PKI) usage is the dependence of sharing the keys among users. We have developed a new security model for telemedicine system which implements an encryption scheme for securing sensitive health care information using PKI, but without sharing the keys. We have used time domain attribute to increase the security and reliability of the data. Also an efficient and novel access control system, based on authenticating the user identities of the users who want to access the data, is tested. This access control method also guarantees the integrity of the data that is shared among users. The identities are hashed and encrypted using symmetric key algorithm.

Keywords: *Health care data, time domain attribute based PKI system, identity based access control.*

1. Introduction

Nowadays, healthcare domain is substantially growing fast and becoming difficult to handle than in past decades. Healthcare developments are progressively crucial in healthcare innovativeness and health care entities are developing rapidly but in different velocity, more volumes of sensitive data in an unpredictable ways. Since these information are very sensitive and need to be very reliable for accurate analysis of patient's future health condition and new innovations in treatments and medicines, they have to be protected in an efficient way [1, 2]. One of the prominent development in health care industry is telemedicine. Since the mobile applications for health care are available in large number over internet and they are very easy to install and use, people now are moving to telemedicine. Telemedicine is a new health care paradigm in which the patient and the doctor need not to meet physically for consultation. If we could make the patient and doctor information available online for both patient and doctor, we can easily implement telemedicine. Telemedicine is a powerful methodology wherein we can realize all kinds of medical practices without any difficulties of patient and doctor[3, 4]. One of the solution for any health care system(HCS) could be, we can encode the sensitive health information and authorize the selective customers to get the decryption keys. Since we deal with very valuable health care information, we need more data security than that is provided by typical standard encryption algorithms like, AES, DES, RSA, etc.

The community of identity based cryptography schemes are proved to be computationally efficient as they are built up on the computations associated with bilinear non degenerate maps. It is a mathematical function of a set element pairs from one cyclic group to another cyclic group and their order is same prime number. The only difference is that the discrete log problem of the first group is hard handle[5]. It is always assumed that the bilinear maps we choose are one way functions as it is simple to compute the result of a given pair of values but it is difficult to compute the reverse of it. This is the basic criteria for providing the data security in our proposed system. It is often referred to as the Bilinear Diffie-Hellman(BDH) Assumption, as the BDH problem can be reduced to the discrete log or reverse operation for the bilinear maps [6].

A bilinear mapping can be defined as a pairing with the property:

pair($op1 \cdot op2$, op3 $\cdot op4$)=*p*air(op3 $\cdot$ *op2*, op1 $\cdot$ *op4*)

May-June 2020
ISSN: 0193-4120 Page No. 3521-3528

In a couple of well-known IDE systems, the · (dot) operator implies the multiplication of an elliptic curve point with an integer [7]. Given op1 and op2, it is easy to compute *op1*· op2, but finding op1 given op2 and *op1*· op2 is computationally infeasible. In 2001, Boneh and Franklin proposed an efficient IDE system, which is based on Weil pairing. More than 20 years later Shamir had presented his first research idea, Boneh and Franklin gave a new dimension to Identity based encryption (IDE). By following these three authors, a very good number of similar IDE and Identity based system(IDS) systems had been developed. As a large number of research works in this domain are based on pairing values, later the identity based cipher is known as pairing based cipher. The cipher algorithms employed in the IDE scheme developed by Boneh and Franklin are illustrated below[7]:

Setup:
The IDSchoses a random point 'P' on some elliptic curve, a private key '*sk'*. The master public key is form using the set {*sk·P, P}.*
Encryption:
The sender converts the selected identity attribute value, the time *'ti',* to a point on the elliptic curve. Sender computes a key *k*:*k*=pair(*r·ti*, *sk·P*), were *r*is a random number. The cipher text is calculated as $E_k[M]$ and this cipher text is sent to receiver along with *r·P.*
Decryption:
The receiver may or may not yet poses a private key. To get it, he/she get authenticated by IDS. IDS in turn computes $s \cdot ti$ and sends it to the receiver thorough a trusted mode of communication. It acts as the secret key for decryption. Once the receiver obtain the cipher text and $r \cdot P$, he/she can get back the decryption key *k* as pair($s \cdot ti$, $r \cdot P$). This computation can recover back the key because of the bilinear property of the bilinear maps. Receiver then decrypts the messages using the key *k*.

Only the receiver can compute the decryption key as only receiver has $s \cdot ti$. Since Shamir had laid the research seed on identity based signature system using RSA in his proposal, many new and young researchers have discovered and developed different implementations of pairing-based IDS systems. Boneh, Lynn and Shacham also contributed their research outputs in this domain[8].

Most of the available attribute revocation techniques [9 - 14] require to encrypt again all the already encrypted data, enabling all new users to decrypt all the stored messages, but they should satisfy the access control policies. Since this scheme does not deal with any time limit, a fine-grained access control could not be implemented.

So, we analysed the attribute based encryption (ABE) schemes and developed a new novel methodology by combining ABE with time domain attribute. Here, the important challenge is that how we introduce this time attribute value in encryption and decryption. The main issue in using the time domain attribute is that it is dynamic in nature, i.e. whose value will keep on changing. We studied the ways in which the dynamic property of this time domain is used effectively. We decided to add the time domain attribute as a part of identity of the message generated by the patient and doctor. Hence, we embedded the time domain value in message before encryption of the message.In this paper, we put forward a new and novel cryptographic solution, to securely share the sensitive health care information among properly authenticated users. Even though we use asymmetric keys for encryption and decryption, the users are not provided with the public and private key pair. Only the authorized user will receive the key needed for decryption.

2. Related work

This section analyses the previous works done in the attribute based encryption field and the advantages and disadvantages are enumerated.

2.1 Attribute-based encryption(ABE)

The attribute-based encryption (ABE) model was first presented by Sahaiand Waters [15] which emphasis on providing data access control with secure data. This ABE was implemented with PKI by allowing the uses to use their own attributes to encrypt and decrypt the data. Since the private key is generated using the user's identity, the cipher text computed will also depend upon the user's identity. So, while decryption the attributes of the user, who is decrypting, and the attributes of the cipher text should match. Otherwise, decryption will not yield the original plain text. Even though any adversary holding multiple keys, at least one of them must form the threshold key that satisfies the personal attributes of the user.

The disadvantage of this system is the necessity of the data owner to be aware of public keys of all the legitimate users for encryption of his/her own data.

2.2 Key policy attribute based encryption (KP-ABE)

In the year 2012, Chang-Ji, Sun Yatsen and Jian-Fa proposed a key policy attribute based encryption [16]. This security model guarantees data access control along with secure data and also we can predefine the size of the cipher text. This system is basically an enhanced version of attribute based encryption. In this scheme the users are provided with a tree structure describing the data access policies. The attribute values can be derivedfrom leaf nodes of the tree structure and thereby the secret keys are derived from the attribute values user and also these secret keys show the monotonic behaviour in determining the access structure. The output of encryption is labelled with the attributes associated. KP-ABE scheme is implemented using the following algorithms.

Setup
The predefined value of a security parameter(K) is used to generate two keys: a public key(PK) and a master private key(MK). The public key(PK) is used in

May-June 2020
ISSN: 0193-4120 Page No. 3521-3528

encryption of messages and the individual user private keys are derived from the master private key(MK).

Key Generation

For every user the private key(SK) is derived from the master private key(MK) using the particular user's data access policies. This private key iss used in the decryption process of that user.

Encryption

The data owner encodes the message(M) using PK and his/her own data access attribute values. The output is the cipher text(E).

Decryption

The cipher text(E) is decrypted with the user's SK for his/her own access policies. The decryption will be successful only if user's attributes are matching with the access policies associated with the cipher text(E).

This KP-ABE is better than ABE scheme in providing data access control. The disadvantage of KP-ABE is that the user who owns the data cannot decide which user category can decrypt the data. In this scheme, we need to rely on a trusted third party for key management.

2.3 Cipher text policy attribute based encryption(CP-ABE)

It is known to be another variation of ABE, which was proposed by Sahai [17]. In this model, the access policies of every user determine the final cipher text, and a set of attributes of the legitimate users determines the private key for the corresponding user. An user can do decryption successfully only if the set of private key user attributes matches the cipher text access policies. CP-ABE works differently when compared to KP-ABE. But, the CP-ABE algorithms are very similar to that of KP-ABE. Depending on the user attributes access policies, messages are encrypted. The access structure residing in cipher text determines which key can decrypt the data.

Setup

An unique value K, which is responsible for the system security, is selected as the input to this algorithm. A public key(PK) and a master private key(MK) are returned as output. The public key PK is used with encryption and individual user secret keys are derived from MK in key generation process. This secret key is used for decryption and is visible only to the authority.

Key Generation

Using a set of user attributes and MK this process generates secret key SK for all users that is used by the user in the decryption of the cipher text satisfying T.

Encryption

Using the public key PK and the access structure T, the owner of the data encrypts the message to compute the cipher text 'CT'.

Decryption

The cipher text CT is decrypted using the secret key SK of the user who does the decryption to derive the message M, only if the constraints on data access of the ciphertext CT matches with that of SK.

CP-ABE shows the improvement over KP-ABE that the cipher text need not determine which user can decrypt it. This is very much useful in implementing the access control in real world applications. Since the secret key SK is derived from the master secret key using a particular user's access policies, an user can realize only the data access determined by his/her own attribute set. The only flaw we can find in this CP-ABE scheme is that it is less flexible and is costly in terms of CPU computation. For these reasons most of the organizations are not willing to use it.

2.4 ABE scheme with non-monotonic access policies

The data access policies of older ABE systems could be described in only monotonic way. They lack sophisticated methods to realize an access structure with negative attributes. Ostrovskyz [18] et al. introduced non-monotonic data access policies in an ABE during 2007 and it made use of negative word to describe the message attributes. This scheme includes the following four algorithms:

Setup

While defining the core system with an access structure 'T' anattribute 'd'is used to specify how many number of attributes that every ciphertext has.

Key Generation

A public key 'PK' for encryption and another key 'D' for decryption of cipher text are made available for the legitimate users. The decryption is done by the legitimate users and it is successful only if the attributes of cipher text and that of T are matching.

Encryption

The plain text M is encrypted using PK with a predefined attribute set and T. The encrypted message CT is returned.

Decryption

If the access structure T is successfully checked, the plain text M can be derived from the cipher text CT using the decryption key D. This aspect makes the system to be non-monotonic in nature.

2.5 Comparison of ABE schemes

The problem with ABE with non-monotonic data access policies is that the cipher text contains many negative attributes. These negative attribute values does not actually associate with the cipher text. Even though every attribute is described by at least one negative word, none of them are used in decryption of the cipher text. This aspect only increases the cipher text data size and thereby increase the overhead of manging huge data. It was proven that it is infeasible and difficult to implement and use as different constants are used for encrypting different messages. The performance analysis of the ABE schemes, we analysed is presented in Table1. From the table we can infer that each ABE is has its own advantages and disadvantages. We have to select the

May-June 2020
ISSN: 0193-4120 Page No. 3521-3528

ABE scheme by analysing the specific needs of the system environment where it is used.

Table 1: Performance Analysis of ABE schemes

Criteria	ABE	KP_ABE	CP-ABE
Fine grained access control	Very less	Excellent with re-encryption possibilities otherwise its average	Medium
Efficiency	Medium	Excellent and/or mediumfor broad-casttype system	Average, not suitable for modernbusiness environments
Computational Overhead	High	Medium	Medium
Collision Resistant	Medium	Fair	Fair

3. Proposed Security model

A typical telemedicine system is implemented in this work and tested for its security measures. As our security model require time attribute for our encryption process, we decided to utilize system time. We made time slots from the 24 hours in a day and the time space could reasonably defined as T= $\{t_0,t_1, \ldots, t_{10},t_{11}\}$.For every time slot a PKI key pair is generated at the beginning of the time slot. We have added the time slot information with the cipher text and also the encryption key in the CP-ABE system, so that the users who have all required attribute values in a particular time slot only can use the decryption key to decrypt the data.

This scheme make use of a trusted security managing module called Trusted Security Provider (TSP). Before any operation begins in the HCS, the TSP generates a secret key *skPIN* using AES key generation algorithm. This key will be used for encrypting the PIN of the user before storing it in database. TSP also public/private keypair for every time slot (denoted *pkTSP* and *skTSP*in the following figures) and make *pkTSP* is used to encrypt data and *skTSP* is made available to users who got authenticated successfully for decryption. The different processes involved in the proposed system is illustrated below:

3.1 Key generation

- In this proposed system we maintain time slots to check the data integrity. The duration of a time slot can be defined at our choice. In this implementation we have chosen 120 minutes. So, there will be 12 time slots in a day $[t_0,t_1, \ldots, t_{10},t_{11}]$
- RSA key generation algorithm is used to generate a public and private kay pair of length 512 bit for each time slot.
- A log (‘log’ table) is maintained for key pairs and the time slots in the TSP server.
- The domain of the nonce ‘n’ used in the authentication process is changed for every time slot. It is usually a natural integer domain of length 100.
- A 512 bit AES secret key ‘skPIN’ is generated only once and AES algorithm is used to encrypt the PIN of all the users. It will be used by TSP alone.

3.1. Registration process for doctor

- Doctor enters his/her details. [Name, Qualification, Specialization, Experience, Age, Gender, Social status, Address, Contact number, Mail id]
- The details entered by the doctor are stored in ‘doctor_details’ table.
- Unique ID for the doctor is created with high entropy.
- An unique and highly random 4-digit PIN is also generated for the doctor.
- Both ID and PIN are sent to the doctor’s registered mobile number for further login process.
- MD5 hash algorithm is used to hash the ID to a hash digest, which is stored in the same table.
- The PIN is encrypted using AES algorithm with 512 bit secret key ‘skPIN’ and the cipher is stored in the table.

3.2. Registration process for patient:

- Patient enters his/her details: [Name, Age, Gender, Social status, Address, Contact number, Mail id, Insurance policy number, Allergic to, Blood group, Height and weight]
- The details entered by the patient are stored in ‘patient_details’ table.
- Unique ID for the patient is created with high entropy.
- An unique and highly random 4-digit PIN is also generated for the patient.
- Both ID and PIN are sent to the patient’s registered mobile number for further login process.
- MD5 hash algorithm is used to hash the ID to a hash digest, which is stored in the same table.
- The PIN is encrypted using AES algorithm with 512 bit secret key ‘skPIN’ and the cipher is stored in the table.

3.3. Authentication process

We have designed a two layered protocol for authenticating an user. The user name and ID are read first to verify the user. Later the PIN number is validated for correctness. The hash value of the user ID is stored in the database during registration process. The PIN is

May-June 2020
ISSN: 0193-4120 Page No. 3521-3528

encrypted using AES algorithm with the secret key *skPIN*. The authentication process is illustrated in figure1.

- A request to use the health care system (HCS) from the user (Doctor / Patient) is sent to the Trusted Security Provider (TSP). The user submits his / her name and ID.
- The ID is verified for the user name first and if it is correct, TSP responds the user with a nonce n, n Є N.
- The user have to add 'n' with his / her PIN and also increment the sum by one. (S = n+PIN+1). The final sum S is sent to TSP.
- TSP verifies S and upon successful verification of S the user is allowed to use the HCS further. Otherwise not.

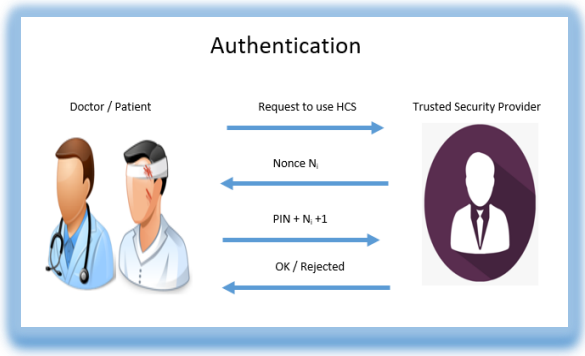

Figure 1: Authentication process

3.4. Encryption process by patient

The encryption of the health compliant given by the patient is encrypted by TSP using the public key of PKI. Figure2 explains this process.

- Once the patient is successfully authenticated he/she can enter the health complaints (M).
- When the patient submits his / her information M, it is encrypted using RSA algorithm with the public key 'pkTSP' generated for the current time slot 't_i'.
- The current time slot $t_i$is also encrypted using RSA algorithm with pkTSP.
- The cipher text C is prepared by encrypting the concatenating M and t_ii.e C = En (M, t_i)
- The cipher text C is stored in the 'consultation' table along with an ID, patient ID, doctor ID, and date & time.

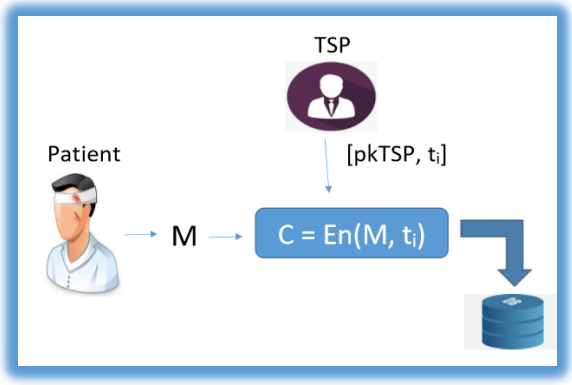

Figure 2: Encryption by patient

3.5. Decryption process by doctor:

Decryption will be done by legitimate user, after acquiring relevant key and time slot information from TSP. Figure3 depicts this algorithm.

- After successful authentication, doctor can get the private key 'skTSP' and the current time slot 't_x' information from TSP.
- The cipher text C is brought to the local memory of the doctor's system.
- Using RSA algorithm C is decrypted to get M and t_i using skTSP.
- De (C) => [M, t_i]
- The decrypted time slot t_i and the time slot received from TSP t_x are compared for a match.
- If t_i= = t_x, then M is reliable, otherwise not. The data integrity of M is preserved using time slot.

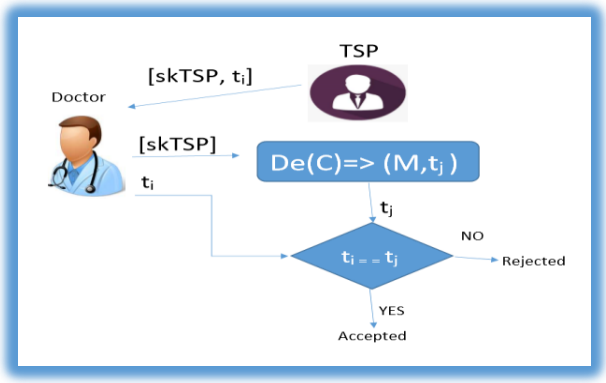

Figure 3: Decryption by doctor

3.6. Encryption process by doctor

When the doctor submits his medical advices TSP will encrypt the message using its public key including the current time domain information. The Figure4 cited below shows this encryption process.

- After analysing the health complaints (M) and previous health history of the patient in the 'consultation' table, doctor submits his advice. It includes Prescription ID, Prescribed medicines, Suggested clinical tests, Food plan, Do's and dont's, Next consultation date and time, and Other relevant information.
- When the doctor submits his / her information M (cited above) , it is encrypted using RSA algorithm with the public key 'pkTSP' generated for the current time slot 't_i'.
- The current time slot $t_i$is also encrypted using RSA algorithm with pkTSP.
- The cipher text C is prepared by encrypting the concatenating M and t_ii.e C=En(M,t_i)
- The cipher text C is stored in the 'prescription' table.

May-June 2020
ISSN: 0193-4120 Page No. 3521-3528

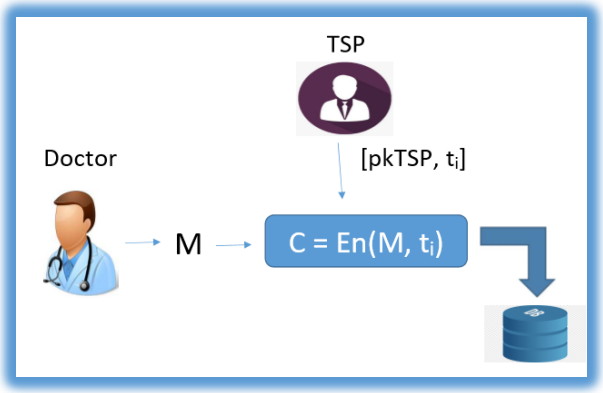

Figure 4: Encryption by doctor

3.7. Decryption process by patient

Once the doctor gave his health advices the patient can view it after decrypting the message from doctor. Figure5 describes the decryption process done by patient.

- After successful authentication, patient can get the private key 'skTSP' and the current time slot 't_x' information from TSP.
- The cipher text C is brought to the local memory of the patient's system.
- Using RSA algorithm C is decrypted to get M and t_i using skTSP. De (C) => [M, t_i]
- The decrypted time slot t_i and the time slot received from TSP t_x are compared for a match.
- If t_i= = t_x, then M is reliable, then the information M, i.e. the doctor's response for the patient's compliant is displayed to patient. Otherwise the patient is advised to contact the hospital for guidance.

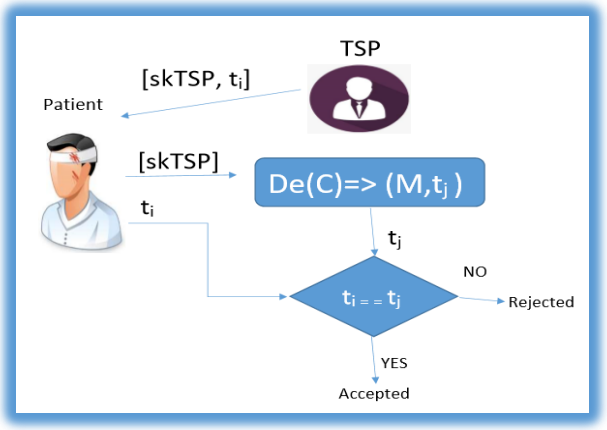

Figure 5: Decryption by patient

4. Experimental setup

The experiments are carried out using the database 'Telemedicine' created in MySQL 8.0 database server. The proposed system is implemented Java and SQL queries. The details of the database is given below. The Entity-Relationship diagram of the database is depicted in the figure6.

Database name: Telemedicine
Table1: patient_details
Table2: doctor_details
Table3: consultation
Table 4: prescription
Table5: log

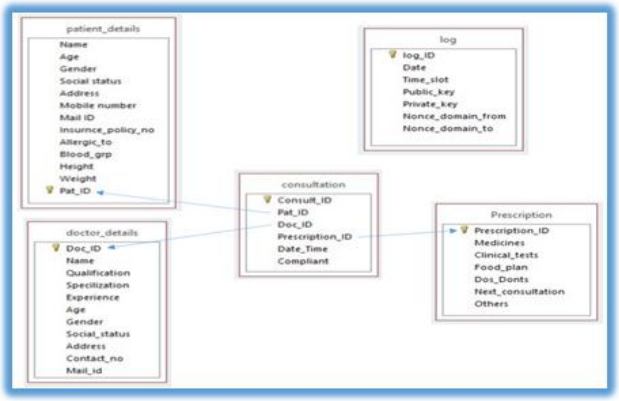

Figure 6: Entity-Relationship diagram of the database Telemedicine

The specifications of the system in which the test done is given below.

- Processor - Intel Xeon – 4 core
- Speed - 3.3 GHz
- OS – Microsoft10
- Memory - 4GB
- Storage -500GB

5. Security analysis of proposed system

The authentication process is basically a two layered authentication protocol. This process reads the user name and ID for verification. In the database the user ID is stored as a hash value. The ID read from the user is also hashed and checked for a match. If they are the same then only the user can proceed further. Once ID verification is done then the user is expected to successfully pass through the PIN validation. The PIN generated during the registration process is encrypted using AES algorithm with its secret key (*skPIN).* This key is generated only once and will be used only by TSP. User need not enter his/her PIN, rather a very simple computation to be done and the result to be submitted to the TSP for validation. Even though it require a little bit effort, it is appreciable when data security is concerned. Only three chances will be given for entering correct user ID. If the third attempt is also failing the user with particular username cannot use the HCS for next 48 hours. The user have to contact the HCS personally to recover his credentials for further usage of HCS. If the PIN validation fails the HCS terminates the service at once.

Even though the PKI is the used in the proposed system the private and public key pair is not assigned to any user. Only the TSP will hold the key pairs. The RSA key pair will be newly generated at the beginning of every time slot. So the cryptanalysis done by the eavesdroppers and man-in-the-middle attackers can never

May-June 2020
ISSN: 0193-4120 Page No. 3521-3528

predict the plaintext. Since the key pairs are not shared among users, there will not be any leakage of the keys. Encryption is done by TSP itself and only the decryption will be done by user. The current time slot information is added with the message to be encrypted. After encryption the cipher text is stored in the database. For decrypting the message from the database, the must have successfully gone through the two layered authentication process. Only the legitimate users will get the private *skTSP* and the timeslot associated with the key pair for decryption. After decryption, the user have to check for a match between the decrypted time slot and time slot received from TSP. this check is done to ascertain the data integrity of the message.

In our proposed security model all sensitive information are hashed and/or encrypted before storing them inn database. If any data leakage happens also the adversary can get only encoded data. He/she cannot make any use of it, but only tampering of data is possible. If any such attack happens also, data reliability is checked with the time attribute. Even though we can design our own cryptographic algorithm, we have chosen the standard algorithms RSA and AES, as they are very robust in nature. Breaking them is a cumbersome process taking years together to finish.

While analysing our system, it has showed its robustness by terminating the service when any kind of hacking is tried. So, we can make use our security model in any kind of HCS which deals with sensitive information. If we could provide a secure communication channel and trusted insiders then this system would be an ideal security model for any kind of HCS.

6. Conclusion

We studied and analysed the various encryption schemes which use the attribute values of authenticated users. Even though KP-ABE and CP-ABE have laid a strong basis to new data access control techniques, they have showed flaws in being flexible and efficient. Because of these reasons these techniques were unable to be used in business applications. To address the identified flaws in these techniques we have presented our novel secure system using time domain attribute in which we embed time slot information into both the cipher texts and the encryption keys. In this encryption scheme we designed a TSP, which generates the required keys. The public key of the TSP is used for encrypting the message and the private key of the user is used for decrypting the cipher text. Also identity based authentication for access control is proposed. The user who holds a valid and unique identity only can get the private key from TSP for decryption.

Even though this security framework establishes a better data protection, it is bit complicated and time consuming method when compared to the existing ABE schemes. The ID recovery system is under design process to enable the legitimate user to get back his forgotten ID or to get a new ID. In future this security frame work could be implemented with some modification on cloud to realize health care big data systems. We are planning to implement few more ideas to refine the time domain encryption and identity based access control scheme to increase the overall performance of our system.

References

[1] RuiZhangand Ling Liu: '' Security Models and Requirements for Healthcare Application Clouds'' in IEEE 3rd International Conference on Cloud Computing, 2010

[2] J. D. Halamka, "Using Big data to Make Wiser Medical Decisions," Harvard Business Review, 2015. [Online]. Available: https://hbr.org/2015/12/using-big-data-to-makewiser-medical-decisions.

[3] B. Marr, "How Big data Is Changing Healthcare," Forbes, 2015. [Online]. Available: http://onforb.es/1bfRQ0b.

[4] A. Patrizio (2007). Salesforce.com Scrambles To Halt Phishing Attacks. nternetNews.com. Published: November 7, 2007. http://www.internetnews.com/ent-news/article.php/3709836/alesforcecom+Scrambles+To+Halt+Phishing+Attacks.htm

[5] *Bilinear Pairings.* http://rooster.stanford.edu/~ben/maths/ep/pairing.php

[6] Yacobi, Yacov, *A Note on the Bi-Linear Diffie-Hellman Assumption*, Cryptology ePrint Archive, Report 2002/113, 2002. http://citeseer.ist.psu.edu/yacobi02note.html.

[7] *Identity-based encryption.* http://www.voltage.com/technology/ibe.htm

[8] D. Boneh, B. Lynn, H. Shacham, *Shor Signatures from the Weil Pairing*, Asiacrypt, Lecture Notes in Computer Science, vol. 2248, pages 514+, 2001. http://citeseer.ist.psu.edu/boneh01short.html.

[9] Sahai A, Seyalioglu H, Waters B. Dynamic credentials and cipher textdelegation for attribute-based encryption. In advances in cryptology-CRYPTO (pp. 199-217). Springer Berlin Heidelberg.

[10] Yang K, Jia X, Ren K. Attribute-based fine-grained access control with efficient revocation in cloud storage systems. In proceedings of the ACMSIGSAC symposium on information, computer and communications security 2013 (pp. 523-8). ACM.

[11] Yang K, Jia X, Ren K, Zhang B, Xie R. DAC-MACS: effective data access control for multiauthority cloud storage systems. IEEE Transactions on Information Forensics and Security. 2013;8(11):1790-801.

[12] Yang K, Jia X. Expressive, efficient, and revocable data access control for multi-authority cloud storage. IEEE Transactions on Parallel and Distributed Systems. 2014;25(7):1735-44.

May-June 2020
ISSN: 0193-4120 Page No. 3521-3528

[13] Lewko A, Waters B. Decentralizing attribute-based encryption. In annual international conference on the theory and applications of cryptographic techniques 2011 (pp. 568-88). Springer Berlin Heidelberg.

[14] Boldyreva A, GoyalV, Kumar V. Identity-based encryption with efficient revocation. Inproceedings of the ACM conference on computer and communications security 2008 (pp. 417-426). ACM.

[15] Goyal V, Pandey O, Sahai A, Waters B. Attribute-based encryption for fine-grained access control of encrypted data. In proceedings of the 13th ACM conference on computer and communications security 2006 (pp. 89-98). ACM.

[16] Wang CJ, Luo JF. A key-policy attribute-based encryption scheme with constant size cipher text. Ineighth international conference on computational intelligence and security, 2012 (pp. 447-51). IEEE.

[17] Bethencourt J, Sahai A, Waters B. Cipher text-policy attribute-basedencryption. In 2007 IEEE symposium on security and privacy (SP'07) 2007 (pp. 321-34). IEEE.

[18] Ostrovsky R, Sahai A, Waters B. Attribute-based encryption with non-monotonic access structures. In proceedings of the ACM conference on computer and communications security 2007 (pp. 195-203). ACM.