Buletin Teknol. Tanaman, Bil.12, Tambahan.2 (2015): 336-339

The New Model to Control Access to Personal Medical Information Stored In the Cloud

Morteza Nikou Ghadam, F. Emad

Abstract
Cloud computing is of a scalable and virtual-resource set that can provide the required services to users based on how extent they use the service. E-health system by creating a personal health would record collect and integrate all information and history pertaining to the health of the patient. Integrating the information gives rise to ease of access for physicians, patients and other related users such as pharmacies, with respect to the privacy of patients. The technology, by sharing patient information stored, can have significant benefits in providing medical services electronically. Reducing costs and ease of access for physicians to patient information are the most important factors in addressing to the cloud computing E-health. The security and the privacy are the biggest obstacles to the admission of this style of computing in widespread. This article aiming to alleviate this problem suggests a new architecture which uses ecc encryption for storing and sharing secure PHR in the cloud.
Key words: security of cloud computing, cloud storage, PHR

Introduction
Moving and transferring treatment software to cloud-based models and their management through the clouds revolutionize the way of health and medical cares and its results make access to health care for everyone and everywhere possible and in addition to the reduction in treating travel costs of patients and physicians, it has been taken essential steps to provide and benefit the health patterned system integrated, synchronized and aligned with the technology, because of the welfare of patient. In the world of cloud computing, aside from the advantages of the use of this method computationally, there have been complex challenges in this area to fill a gap in security, privacy and controlling unauthorized access to the data stored in the cloud. The most important issues in the field of data storage in the cloud would be related to protect the data and prevent unauthorized access to data, as well as provide easy access for authorized users [2].

Literature
Hui et al in 2o14 provided a method for controlling secure access to personal health records of patients in the cloud [5]. Implementing Personal Health Record system under the cloud computing environment will lead to a reduction in the cost of infrastructure management and the handling of the users according to their demand and being a real-time service for users. ID-based encoding and bilinear paring calculations are used in this design. CA divides the data into several classes and determines a random number as a decoding key for every class. Then it employs an ID-based matrix of access control to connecting the users to the data related to their own health records in which 1 and 0 indicating a relationship and no relationship between them, respectively. Using bilinear pairing calculations and hash functions produce a function and puts available to the users. Request of each user is checked by the CA with regard to access control matrix, if an authorized user, the user obtains the keys of code, by using private key and intended function. The problem with this method is that user access in each time, heavy computation must be done to get the keys.

Setting up proposed design
In the proposal all users that may include: patients, doctors, nurses, health care professionals and medical research institutes etc. with various access levels are divided into classes of security with access rules are hierarchical. On the tree user hierarchy, the patients are put in leaf, so each patient has access to her/his own medical information, but not others' the medical records. Similarly other users would be placed on higher-level access, according to laws of medical records access for each patient. The highest-secure-class user can access a lower security class data, the method of key management based on cryptography (ECC) of user hierarchy provided by Dr. Nikou Ghadam is used [4]. Compared with other key-based ECC, the model has less computational overhead, storage and complication.

Setting up steps of the proposed design
Step 1: CA selects a secure elliptic curve C on Galois field GF (p) such that p is a prime number and G is an essential point on the elliptic curve,
Where q is greater than or equal to 163 bits.
Step 2: private key d_{CA} is selected from a range $d_{CA} \in [1, q-1]$ and the corresponding public key P_{CA} is produced by relation (1).

$$P_{CA} = d_{CA} . G \quad (1)$$

The CA selects unique private key d_i from a range of $d_i \in [1, q-1]$ and produces the corresponding public key P_i for class security SC_i, $1 \le i \le n$ with respect to (2). The private keys of class SC_i via secure channels will be sent to class members of SC_i.

$P_i = d_i . G$ (2)

Step 3: CA Z calculates value Z_i by (3) so that K_i as a random integer digit from [1, q-1] for each class of security SC_i.

$Z_i = K_i . G$ (3)

And CA calculates key value SK_i with respect to (3). H is a one-way hash function which transforms coordinate x of point Z_i on the elliptic curve to crypt key SK_i.

$SK_i = H(Z_i)$ (4)

Step 4: for classes of security with $SC_j \leq S_i$ and $1 \leq j \leq n$, point Y_{ijS} is calculated by relation (5) and points Y_{ijS} of each class are sent through a secure channel to the security class.

$Y_{ij} = K_j . P_i$ (5)

Step 5: CA distributes the values of p, q, G, P_i, P_{CA} and function H and holds its private key d_{CA}, keys Sk_i for all security classes but removes the private keys d_i related to the security classes.

Stages of making keys

Step 1: each security class calculates its private inverse key and saves it safe. For example, for the private key d_i, d^{-1}_i is calculated.

Step 2: every the security class SC_i calculates value of Z_i by equation (6) for itself and value of Z_j by (7) for security class SC_j by ($SCj \leq SCi$, $1 \leq j \leq n$):

$Z_i = d^{-1}_i . Y_{ij}$ (6)

$Z_j = d^{-1}_i . Y_{ij}$ (7)

Equations (8), (9) and (9) show how to calculate Z

$Y_{ij} = K_j . P_i = K_j . (d_i . G)$ (8)

$d^{-1}_i . Y_{ij} = d^{-1}_i . (K_j . (d_i . G)) = K_j . G = Z_j$ (9)

$Z_j = d^{-1}_i . Y_{ij}$ (10)

Step 3: security class SC_i calculates encrypt key SK_i by (11).

$SK_i = H(Z_i)$ (11)

H is a one-way hash function that transforms the coordinate x of point Z_i on the elliptic curve to encrypt key SK_i.

For example, the security class SC_3 Fig (3-2) can see the data of security classes {SC_2, SC_5, SC_6, SC_7, SC_8} of lower-level hierarchy, so it has to produce this class keys to be {SK_2, SK_5, SK_6, SK_7, SK_8}. First, the parameters Y_{ijS} are calculated by CA and sent to SC_3.

$SC_3 : Y_{3,2} = K_2 . P_3$, $Y_{3,5} = K_5 . P_3$, $Y_{3,6} = K_6 . P_3$, $Y_{3,7} = K_7 . P_3$ (12)

$Y_{3,8} = K_8 . P_3$, $Y_{3,3} = K_3 . P_3$

Changing encrypt key of security class

Each security class SC_j may need to change the secret key of SK_j. In this scheme, CA selects value of k^*_j from [1, 1-q] and the value of secret key of SK_j is calculated by (19).

$SK^*_j = H (K^*_j . G)$ (13)

For each class of security that there is a relationship $SCi \leq SCi$, CA calculates equation 20 and sends the obtained value in accordance with explanation of Part 3.3.3.

$Y^*_{ij} = k^* . p_i$ (14)

Other dynamic features such as adding new security class, removing existing class of security, creating a new relationship between security classes and cancellation of existing relationship can be done according to the design.

Proposed model of PHP storage in the cloud

System model

The proposed model is composed of three parts, which are as follows:

- CA: This component has a lot of data to store a high volume of calculations that for this purpose, cloud centers are under use. The data is encrypted by CA and sent to the provider of cloud service. It is also responsible for classification of data and user in different security classes. As formerly mentioned, CA is responsible for execution, secret key distribution to the users, generating and updating the access control matrixes.
- Cloud service provider (CS): it can be assumed that CS stores the high volume of data required by the CA.
- End users: the unit requests the information stored on the cloud to use. The authorized users can send data requests to the CA.

CA and CS are connected together through a secure channel such as safe leased lines. The users may communicate with CA using wired or wireless connection via public channels. Members of the project can have access to their data using wireless mobile device under limited resources.

Data storage process by CA

The following steps are executed by CA:

Step 1: All users U = {U_1, U_2, ..., U_n}, where n represents the number of the users, are established by CA and according to the hierarchy of the security classes shown in Fig 2.3 are divided into the different security classes SC = {SC_1, SC_2, ... , SC_n}.

Step 2: CA divides the data into sets of DS = {DS_1, DS_2, ... , DS_n}.

Step 3: CA produces a private key SK_i for every class of SC_i, $1 \leq i \leq n$ to be SK = {SK_1, SK_2, ... , SK_n} and it also produces parameters $Y_{i,j}$ for each class of security SC_i by equation ($SC_j \leq SC_i$, $1 \leq j \leq n$), presented in parts 3.3.3 and

Step 4: CA selects shared secret key SK^{pw}_v randomly for all users of $1 \leq v \leq n$ from $SK_v{}^{pw} \in [1, q-1]$ so that SK^{pw} = {$SK_1{}^{pw}$, $SK_2{}^{pw}$, … , $SK_n{}^{pw}$}.

Step 5: Parameter $Y_{i,j}$ produced by CA and sent through a secure channel for each class SC_i is used to generate secret keys of classes SC_j under ($SC_j \leq SC_i$, $1 \leq j \leq n$), presented in sections 3.3.2.

Step 6: CA sends shared secret key $SK_v{}^{pw}$ to all users through a secure channel.

Step 7: CA produces two access control matrixes. The first matrix is on relationship between users (U_v) and security classes (SC_i) . The second on relationship between security classes (SC_{ij}) and datasets (DS_i) . Within these matrixes, parameter 1 indicates relationship to exist and 0, lack of relationship.
Step 8: CA encrypts the data for each DS_i by private key Sk_i form security class SC_i and sends encrypted data to CS.

Stages of PHR information access to the users

When the user wants to access the data, the following steps are performed:
Step 1: When the user U_v wants to access data of DS_i, s/he sends a request to the CA as follows.
Ui → CA: $E_{SKv}{}^{pw}$ [U_v, RD, RI]
Parameter RD is the data requested by the user and parameter RI is a counter that whenever requesting, it is increased and also to prevent replay attacks. The user encrypts all information of [U_v, RD, RI] with shared secret key $SK_v{}^{pw}$ and sends to the CA.
Step 2: CA decrypts the user's application using the shared key $SK_v{}^{pw}$ and thereby it is validated by the user. Using Access Control matrix, CA checks Fig (3.3) that what security class of SC_i the user U_v belong to and also through other access control matrix Fig (3-4), that what data collection of DS_i the relevant security class SC_i access.
Step 3: After confirmation of request from user U_v by CA, the CA sends this (E_{SKi} [RD]) along with shared key $SK_v{}^{pw}$ to CS.
Step 4: CS re-encrypts the requested data (E_{SKi} [RD]) using user-shared key $SK_v{}^{pw}$ that was sent pervious stage, and sends to the user as follows.
CS → Uv : $ESK_v{}^{pw}$ [E_{SKi} [RD]]
Step 5: after receiving data user U_v decrypts it using a shared key $SK_v{}^{pw}$ alternatively decrypting using the secret key SK_i and thus data can be applicable for the user.
If the user U_v becomes unauthorized, the access to the data will be canceled, the CA updates Access Control Matrix. In this case, the user membership U_v will be removed of the relevant class. The user U_v may hold the key of security class SK_i, but updated access control matrix, CA does not allow for access to the request for the data. Re-encryption helps to protect the data available to users who their access was canceled, and it can be used also when a user is transferred from a security class to another one. For example, a security class member SC_5 is transferred to security class SC_8.

Evaluation of proposals

In this part, the proposals will be evaluated and compared with a similar project conducted. Table 2 shows all symbols that are used for comparison and evaluation.

Table 1: Defining Symbols

Symbols	Definition
T_{MUL}	Time complexity to do modular multiplication
T_{EXP}	Time complexity to do modular exponentiation
T_{ADD}	Time complexity to do modular summation
$T_{EC\text{-}MUL}$	Time complexity to do scalar multiplication
$T_{EC\text{-}ADD}$	Time complexity to do point summation
T_{INV}	Time complexity to do in-filed inverse
T_{Pa}	Time complexity to do bilinear paring calculations
T_{Hash}	Time complexity to do hash function

In Table 2, it is determined time complexity of implementing the various operating units, based on the time of doing modular multiplication.

Table 2: The time complexity of the various operating units in terms of modular multiplication

Time complexity in terms of modular multiplication	Time complexity of every operating unit
T_{EXP}	240* T_{MUL}
T_{ADD}	Ignorable
$T_{EC\text{-}MUL}$	29.3* T_{MUL}
$T_{EC\text{-}ADD}$	0.12* T_{MUL}
T_{INV}	3* T_{MUL}
T_{Pa}	586* T_{MUL}

Computational cost per unit of bilinear pairing computation is twenty times more than scalar multiplication. Thus, each unit of bilinear pairing computation costs 586 times more than a modular multiplication.1GB of data is assumed to exist; in this case, the computational overhead and storage of the data by method [5] are given in Table 3. And the computational overhead and storage scheme are investigated. In the proposed model, the computational complexity depends on the number of datasets (DS_i) within which it has been distributed. If 1GB of data is put into a dataset, the CA should calculate a secret key Sk_i and a parameter $Y_{i,j}$ and a parameter $Y_{i,i}$ that in total, there are 3 numbers of T_{EC_MUL} and a hash function. Hence, the computational overhead CA is listed in term 15.

5. T_{CA} = (3.T_{EC_MUL} + one hash) (15)

Similarly, if the data is distributed in more than one dataset (DS_i), the number of previous operations multiplied by the number of datasets, so if the data is distributed in the dataset v_i, the computational overhead CA is displayed .

6. $T_{CA} = v_i. (3. T_{EC\_MUL} + \text{one hash})$ (16)

However, to access this volume of data by user U_v, if the data is placed in a dataset the user's computational overhead is displayed (10-4).

7. $T_U = 1. T_{INV} + 1. T_{EC\_MUL} + \text{one hash}$ (17)

And similarly, if the data is distributed in more than one dataset (DS_i), the number of previous operations is multiplied by the number of datasets, as a result if the data is distributed in the dataset v_i, the user computational overhead is displayed in (11-4).

8. $T_U = v_i . (1. T_{INV} + 1. T_{EC\_MUL} + \text{one hash})$ (18)

Table 3: Comparison of the time complexity of the proposed plan with provided earlier one

Bilinear pairing calculations	In terms of T_{MUL}		Time complexity		Reference
	CA	User	CA	User	
Yes	v_i. (1655 T_{MUL} + two hash)	v_i. (241T_{MUL} + two hash)	v_i. (2. T_{EXP} + 3. T_{MUL} + 2. T_{Pa} + two hash)	v_i. (1. T_{EXP} + 1. T_{MUL} + two hash)	[14]
No	v_i. (87.9 T_{MUL} + one hash)	v_i. (32.2 T_{MUL} + one hash)	v_i. (3. T_{EC_MUL} + one hash)	v_i. (1. T_{INV} + 1. T_{EX_MUL}+ one hash)	Proposed design

There is no overhead for CA, cloud and the user in re-encryption proposal, because it does not need to generate the key. In table (3), the time complexity of proposed method is compared with the previous designs. First the required total time cost, according to the time of implementing each of the operators will be determined for the previous and the proposed projects. Then, according to Table (2) all the time will be expressed, in terms of time required to perform modular multiplication. According to Table (3), the proposal reduced significantly the time complexity in calculations compared with the project [5]. The proposal has no bilinear pairing calculations. But the proposal [5] with bilinear pairing calculations incurs very computational cost on both sides of entities. In general, the costs of the bilinear pairing schemes would be more than the cost of the plan based on ECDLP.

Conclusions

In the proposed method, combining access control and encryption have tried to provide the mechanisms of an efficient and secure access management for user access to PHR information. In this way, the information related to each patient is stored within the form of encrypted data in the cloud by CA and the user's requests are validated only by the CA. All keys needed by the CA are produced and distributed. And the data encryption is performed by the CA and after the user's request it is re-encrypted by the cloud. The comparisons prove that the proposed scheme compared with Hui Liu et al's, reduces the computational cost considerably.

References

1. A.Behl, "Emerging Security Challenges in Cloud Computing", 2011, word congress on Information and Communication Technologies, PP. 217-222.
2. AHIMA e-HIM "Personal Health Record Work Group, Defining the personal health record: AHIMA releases definition, attributes of consumer health record". J AHIMA 2005;76 (6): 24–30
3. AHIMA. "The Value of Personal Health Records a Joint Position Statement for Consumers of Health Care by American Health Information Management Association American Medical Informatics Association". February 2007
4. Nikooghadam, M., Zakerolhosseini, A., Moghaddam, M.E." Efficient utilization of elliptic curve cryptosystem for hierarchical access control". 2010, the Journal of Systems and Software 83(10), 1917–1929.
5. Chia-Hui Liu, Fong-Qi Lin, Chin-Sheng Chen and Tzer-Shyong Chen.(2014), 'Design of secure access control scheme for personal health record-based cloud healthcare service' , SECURITY AND COMMUNICATION NETWORKS, DOI: 10.1002/sec.1087
6. Barka, E., Sandhu, R.," A role-based delegation model and some extensions," 2000,In: Proceedings of the 23rd National Information Systems Security Conference, Baltimore,pp. 101-114.
7. Bethencourt, J., Sahai, A. and Waters, B. "Ciphertext-policy attribute-based encryption", 2007,in IEEE Symposium on Security and Privacy, IEEE Computer Society, pp.321–334.
8. Blundo, C., Cimato, S., di Vimercati, S.D.C., Santis, A.D., Foresti, S., Paraboschi, S. and Samarati, P, "Efficient key management for enforcing access control in outsourced scenarios", 2009 in SEC, Vol. 297 of IFIP, pp.364–375, Springer, Pafos, Cyprus.
9. Boneh, D., Boyen, X. and Goh, E-J. "Hierarchical identity based encryption with constantsize ciphertext", 2005, in EUROCRYPT, Springer, Aarhus, Denmark, Vol. 3494 of Lecture Notes inComputer Science, pp.440–456.
10. B.P.Rimal, E.Choi and I.Lumb, " A Taxonomy and Survey of Cloud Computing System" 2009 Fifth International Joint Conference on INC,IMS and IDC,Aug. 2009, pp. 44-51.
11. C.Squicciarini, Elisa Bertino. Lorenzo, D.Martino.Fedrica, Paci.Anna "Security for Web Services and Service- Oriented Architectures,"Springer, 2010.

Morteza Nikou Ghadam, Computer department, International University of Imam Reza, Mashhad, Iran,
Email:morteza.nikooghadam@gmail.vom
F. Emad, Computer department, International University of Imam Reza, Mashhad, Iran
Email:f_iut85@yahoo.com
.