Journal of AL-Qadisiyah for computer science and mathematics
Vol.9 No.1 Year 2017

Page 108 - 116



# Multi-Authority System based Personal Health Record in Cloud Computing

**Ghassan Sabeeh Mahmood** **Taha Mohammed Hasan** **Aymen Mudheher Badr**

Computer Science Department
College of Science
University of Diyala

College of Law and Political
University of Diyala

ghassan@sciences.uodiyala.edu.iq dr.tahamh@sciences.uodiyala.edu.iq aymen.mudher@sciences.uodiyala.edu.iq

**Recived : 2\4\2017** **Revised : 15\4\2017** **Accepted : 20\4\2017**

**Abstract**

Personal Health Records (PHRs) is service for health information interchange. Patients can create, control, and exchange their health information. PHRs are outsourced to be stored in the cloud. However, there have been serious privacy concerns about cloud service as it may expose user's data like PHRs to those cloud service providers or unauthorized users. To overcome these challenges, a cloud based PHRs for exchange PHRs among multiple users is proposed. In the proposed system, patients can encrypt their PHRs and store them on the cloud. Furthermore, patients can maintain control over access to their PHRs by assigning fine grained access control. To achieve fine grained access, the proposed PHRs are divided into the personal domain (PSDs) and public domain (PUDs). To ensure security in a cloud based PHRs, Multi-authority based weighted attribute encryption model and Attributed-based access control for the Multi authority model are implemented in the PSDs and PUDs, respectively. The proposed model based on PHRs improves the efficiency of the system in terms of encryption, decryption. Also, the proposed system has proven to be collusion resistant and enhancing the security of PHRs users in a multi owner environment.

**Keywords:** Personal Health Record, Cloud computing, encryption, access control.

## 1. Introduction

PHRs are health record storage service, which offers a summary of the patient's medical history online instead of paper, and allows patients to create, manage, control, and exchange his PHRs with others, including family members, friends, doctors, etc. [1]. Lately, with the development of cloud, PHRs are outsourced to be stored at cloud service providers [2,3]. However, when outsourcing PHRs to cloud, patients will lose the control of hardware's that store PHRs. PHRs stored in the cloud will suffer from attacks, and other information about body of patient, PHRs also includes some sensitive data like a disease. Furthermore, laws like the health insurance portability and accountability [4]. Consequently, to ensure security in a cloud based PHRs, Multi-authority based weighted attribute encryption model and Attributed-based access control for the Multi authority model are implemented in the PSDs and PUDs, respectively. The rest of the paper is organized as follows. In Section 2, Proposed cloud based PHR. In Section 3 and 4, Security and performance analysis are discussed. Finally conclusion of the paper is presented in Section 5.

## 2. Proposed Cloud Based PHR

In the proposed cloud based PHRs system, the key objective is to split the PHRs system into two domains, personal domain (PSDs) and public domain (PUDs). In PSDs, the users are personally related with a data owner (family members or friends), and they make access to PHR based on access rights assigned via the owner. The PUDs contains professional users (doctors, nurses, health researchers, etc.). To certify security in a cloud computing based PHR, Multi-authority based weighted attribute encryption scheme and Attributed-based access control for Multi authority models are implemented in the PSDs and PUDs, successively. The architecture of the proposed system is given in Figure 1.

### 2.1 Proposed Algorithm in PSDs

The owners assign access for personal users and encrypt a PHR files with its data attributes in PSDs. Multi-authority based weighted attribute encryption scheme is used in this domain [5]. The PHR domain defines a public space of data attributes shared via every PSDs, such as personal details, health profile, etc. In this domain, the algorithm operations include five algorithms:

**1) Attribute Weight Split Set:** In this algorithm, the input is a set of attributes.

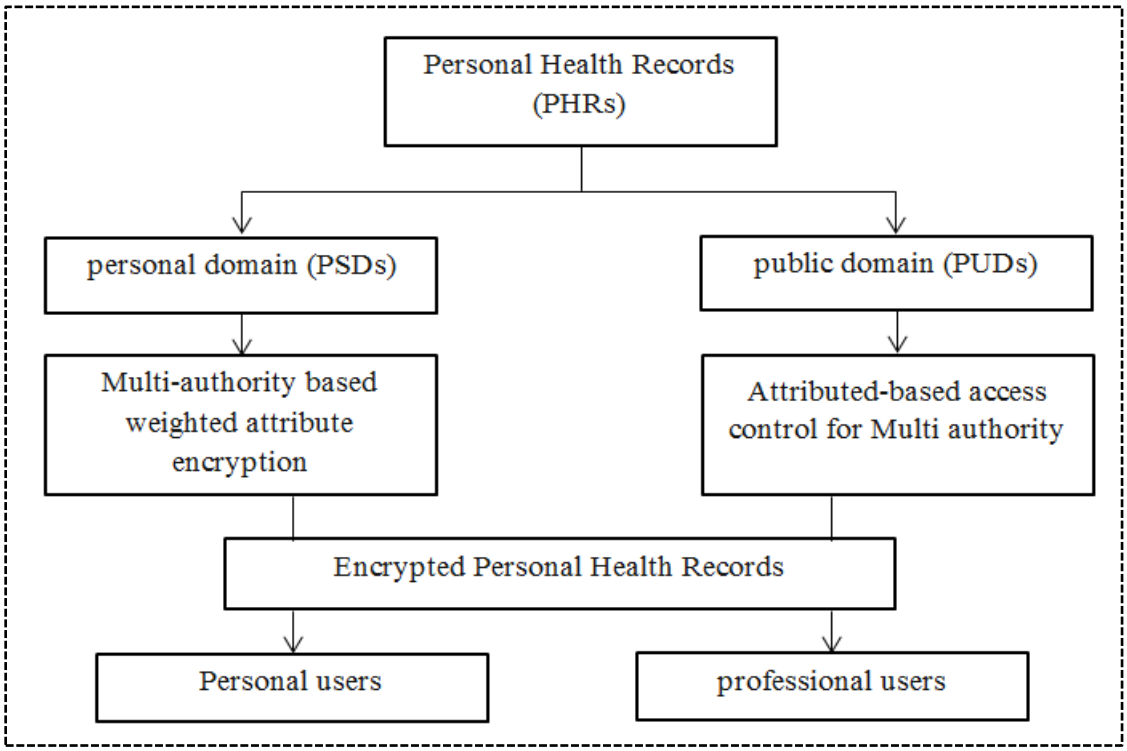


**figure 1: Architecture of the proposed system**

According to importance of attributes the system assigns different weights for different attributes. For each attribute ($a_i$) in the attribute set {A =$a_1$, $a_2$…, ar}, the assigned maximum weight allowed in the system is $\omega_i$ = weight ($a_i$). Each attribute ($a_i$) is split as (($a_i$ ,1) … ($a_i$,$w_i$ )), and the smallest unit is (1), the composed set is attribute weight split set A′.

## 2) Setup

**a) Global Setup:** select two groups G and $G_T$ with the matching prime order p in this system, meeting the bilinear map e: G×G →$G_T$. Let g be the generator of G. It randomly selects y ∈$Z_p$ and h ∈G, and then computes Y = e (g, g) $^{y}$. The outputs of system public key PK as {G, $G_{T,}$ g, h, e, Y}, the master key SK as $g^{y}$.

**b) AA Setup:** Let the number of AAs (multiple attribute authorities) be N. Each authority defines the A′ for the attributes inside its administrative domain. The $AA_k$ manages the attribute set $a_k$ = {$a_K$ ⋯ $a_{nk}$}, and defines the corresponding A′ as $A_k$ ′. Let $n_k$ ′ = | $A_K$ ′ | be the size of the set $A_k$ ′. It selects the first $n_k$ ′ elements from $Z_p^*$,

i.e., 1, $n_k$ ′ (mod p). Then, it randomly chooses elements $t_{1…}$, $t_{n\,k}$ ∈$Z_p$ ′ ⋯, (1 ≤ k ≤ N). It outputs the attribute public keys as $T_j = h^{t_j}$ (1 ≤ j ≤ n k ′).

### 3) Key Generation:

**a) CAKeyGen:** The CA (central authority) assigns a global single user (id) uid to a user. It takes (SK, uid) as input. The central authority first randomly chooses an element ($r_{uid} \in Z_p$). And the outputs of the system user public key are ($PK_{uid} = g^{r_{uid}}$), and the system user secret key is ($SK_{uid} = g^{y}. h^{r_{uid}}$).

**b) AAKeyGen:** It takes ($PK_{uid}$, and the user's attribute weight split set (S′) that corresponds to the user's attribute set (S)) as input. If the user's attributes belong to an attribute authority, it assigns the corresponding attribute secret keys as $D_j = PK_{uid}^{t_j - 1}$.

**4) Encrypt:** It takes (PK, access tree T, AA's identifier k, attributes public key $T_j$ and a message m) as input. Suppose L be the set of leaf nodes, and L′ be the corresponding A′. It first randomly chooses an component $s \in Z_p$, and calculates: $C_0 = g^s$, $C_1 = mY^s = me(g,g)^{ys}$ . Then, it sets (s) as the value of the root node of the T. With the order t −1, and q (0) = s, it defines a random polynomial q (x). And then it chooses t −1 elements as the polynomial coefficients randomly. For each leaf node $a_{j,i} \in$ the corresponding A′, it calculates , $C_{ji} = T^{s_i}{}_j$ , where $s_i = q(i)$ . Lastly, the outputs is the ciphertext: C = (T, C0, C1, $\forall a_{j,i} \in L'$:$C_{j,I}$ ).

**5) Decrypt:** If the S does not satisfy the T, then it returns ⊥. Else, i.e., if $\sum_{p \in \{S \cap L\}}$ weight (p) ≥ t. it chooses t elements from the set K = {S′∩ L′} and calculates: The decrypt message m= $C1/(e(g,g)^{ys})$.

### 2.2 Proposed algorithm in PUDs

The PUD includes professional users such as doctors, nurses, medical researchers, etc. To ensure security in a cloud based PHRs, Attributed-based access control for the Multi authority model is implemented in this domain [6]. Let (G) and ($G_T$) be the multiplicative groups with the equal prime order p and e: $G \times G \rightarrow G_T$ be the bilinear map. Let g be the generator of G. Let H: $\{0,1\}^* \rightarrow Z_p$ be a hash function. The construction of our access control model consists of: initialization of system, key generation, encryption and decryption.

### 1) System Initialization

**a) CA Setup.** The certificate authority (CA) authenticates all the authorities and assigns an authority identifier (AID) to all of them. It also authenticates the users in the system and assign a user identifier (UID) to all of the users. Then, it generates the $PK_{UID} = g^u$ by randomly selecting a top-secret $u \in Z_p$.

**b) AA Setup.** Each attribute authority runs the AAGen method and produces a version key $VK_{AID} = \alpha_{AID}$. And then calculates a public attribute key $PK_{x,AID}$ for all the attributes (x) managed through the attribute authority with an authority identifier as $PK_{x,AID} = g^{\alpha AID \cdot H(x)}$. The $PK_{x,AID}$ can be accessed by each owner.

**c) Owner Setup.** It randomly selects $\beta, r \in Z_p$ as the master key $MK_o = \{\beta, r\}$. And then, it calculates the owner's secret key as $SK_o = \{g^{1/\beta}, r/\beta\}$ and sends it to each attribute authority over a safe channel. Each owner runs the OwnerGen method.

## 2) Key Generation

**a) Public Key Generation.** The attribute authority with authority identifier calculates the owner's public key as $PK_{o,AID} = e(g,g)^{\alpha AID}$ from its $VK_{AID}$. Then, the attribute authority sends the $PK_{o,AID}$ to the owner.

**b) User's Secret Key Generation.** For each user with user identifier, each attribute authority initial authenticates whether the user has any attributes managed by this authority. If the user has attributes, the authority assigns a set of attributes $S_{UID,\ AID}$ to this user according to its role or identity. Then, the attribute authority generates the secret key $S_{KUID,\ AID,}$ according to its attribute set SUID, AID as $S_{KUID,\ AID} = (KUID_{,\ AID} = (PK_{UID})^{r/\beta} \cdot g^{\alpha AID/\beta}$, $\forall x \in$ SUID, AID: $K_{x,UID,AID} = (PK_{UID})^{\alpha AID \cdot H(x)}$ ).

## 3) Encryption

**a)** Splits the data into several data as M = {m1, · · ·, $m_n$}.

**b)** Encrypts M with different content keys {k1, · · ·, $k_n$} by using symmetric encryption.

**c)** Defines an access structure to each content key ($k_i$) and encrypts each content key by Encrypt method.

The encryption algorithm can be created as: Encrypt ($\{PK_{o,AIDk}\}_{k\in IA}$ ,$\{PK_{x,AIDk}\}_{x\in SAIDk\ ,k\in IA}$ ,$MK_o$,m,A). The inputs are $\{PK_{o,AIDk}\}_{k\in IA}$ from the involved authority set IA, $\{PK_{x,AIDk}\}_{x\in SAIDk\ ,k\in IA}$ , the owner's $MK_o$, m and an access structure A over all the selected attributes from the involved attribute authorities. Let access structure be l ×n matrix, where l means the whole number of all the attributes. The function ρ connections rows of M to attributes.

It first selects a random encryption exponent and selects a random vector are used to share the encryption exponent. For i = 1 to l, it calculates λi = random vector · access structure i. It then calculates the ciphertext as

CT = (C = m· $(\Pi_{k\in IA} PK_{o,AIDk}\ )^s$,C˜= $g^{\beta s}$,Ci = $g^{r\lambda i}$ $(PK_{\rho(i),AIDi}\ )^{-\beta s}$(i = 1, · · · , l) ).

## 4) Decryption

The decryption algorithm can be constructed as follows: Decrypt (CT, PKUID, $\{SK_{UID,AIDk}\}_{k\in IA}$ ). If the user's attributes can achieve the access structure, then the proceeds as: let I be $\{I_{AIDk}\}_{k\in IA}$, where $I_{AIDk} \subset$ {1,2, · · ·, l} is as $I_{AIDk}$ = {i: ρ(i) ∈ $S_{AIDk}$}. Let $n_A$ = $|I_A|$ be the number of attribute authorities involved in the ciphertext. Then, it selects a set of constants $\{w_i \in Z_p\}_{i\in I}$ and rebuilds the encryption exponent as s =$\Sigma_{i\in I}$ wiλi if {λi}. The decryption method can decrypt the m = C/$\Pi_{k\in IA}$ $e(g,g)_{\alpha AIDk}{}^s$.

## 3. Security analysis

**3.1 fine-grained access control:** It allows flexibility in specifying access rights for a user. Goyal et al. [7] used access tree structure into KP-ABE method to achieve fine grained access control. Other works that achieve it can be found in [8,9,10,11]. The proposed work can achieve fine-grained access control with Multi-Authority Based Weighted Attribute Encryption Scheme by allowing owner of data to describe and enforce flexible access policies over several attributes. If the user achieves the access policy, then he can access the data.

### 3.2 Collusion Resistance:

**a)** In Multi-Authority Based Weighted Attribute Encryption Scheme each user is assigned with a unique identifier. For each user, the central authority randomly chooses an element ($r_{uid}$) to generate the $SK_{uid} = g^{y} \cdot h^{r\ uid}$ which is associated with the user's uid. All the attributes secret keys $D_j = PK_{uid}^{tj\ -1}$ issued to the equal user from different AA are also associated with the uid of this user. Consequently, it is impossible for different users with different uids to collude together by joining their attributes and decrypt the cipher text. And the cloud server does not know anything about the content of the encrypted data. This is because it does not have the attribute top-secret keys $D_j = PK_{uid}^{tj\ -1}$. Even if it colludes with additional users,

**b)** In attributed-based access control for multi-authority system, each user is assigned with a user identifier, which is a single identifier. All the secret keys it acknowledged from different authorities are all generated based on the user identifier. Consequently, it is impossible for users with user's identifier collude together to decrypt the ciphertext. Furthermore, each attribute authority is assigned to an authority identifier. With the authority identifier, all the attributes are distinguishable even though attributes present the same meaning.

**3.3 Multi-authority Security:** Compared with the single authority attribute system, the multi-authority based system makes the risk of separation authority shared by several authorities and improves the security.

## 4. Performance analysis

The application is deployed on server of cloud computing. When a user needs access to the system, his authorizations are validated based on the information stored in cloud computing. When a user needs a file, the file will be downloaded to his machine, Also, when the user needs to upload a file, the file will be encrypted and then uploaded to the cloud. To measure the performance of the proposed system, the parameters are the encryption and decryption, the encryption and decryption time with different number of authorities are taken for the attributed-based access control for multi-authority in PUDs domain is shown in table (1) and table (2). And also, table (3) shows the Comprehensive Analysis for the multi-authority based weighted attribute encryption scheme in PSDs domain.

**Table (1): Encryption Time with Different Number of Authorities**

| No. of authorities | 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 | 18 | 20 |
|---|---|---|---|---|---|---|---|---|---|---|
| Encryption Time(s) | **0.05** | **0.10** | **0.14** | **0.19** | **0.24** | **0.29** | **0.34** | **0.38** | **0.43** | **0.49** |

**Table (2): Decryption Time with Different Number of Authorities**

| No. of authorities | 2 | 4 | 6 | 8 | 10 | 12 | 14 | 16 | 18 | 20 |
|---|---|---|---|---|---|---|---|---|---|---|
| Decryption Time(s) | **0.16** | **0.28** | **0.39** | **0.53** | **0.66** | **0.78** | **0.89** | **1.05** | **1.18** | **1.29** |

**Table (3): Analyses OF Multi-Authority Scheme**

| Access Policy | Multi-Authority | Weighted Attribute | Fine-Grained Access | Collusion Resistance |
|---|---|---|---|---|
| Access Tree | Yes | Yes | Yes | Yes |

## 5. Conclusion

An efficient cloud based PHR for secure sharing of PHRs is proposed. PHR owners will have complete control of their privacy by encrypting their files to allow fine-grained access by employing new attribute-based encryption, Multi-authority based weighted attribute encryption model and attributed-based access control for the multi authority model are implemented in the PSDs and PUDs, respectively in a cloud computing. The proposed system is collusion resistant and ensures privacy and offers enhanced security.

## References

[1] Huiling Qian et al, Privacy-preserving personal health record using multi-authority attribute-based encryption with revocation, International Journal of Information Security, November- Volume 14, Issue 6, pp 487–497 (2015).

[2] Fernandes, Diogo A.B., Soares, Liliana F.B., Security issues in cloud environments: a survey. Int. J. Inf. Secur., Volume 13, Issue 2, pp 113–170 (2014).

[3] Gouglidis, A., Mavridis, I., Hu, Security policy verification formulti-domains in cloud systems. Int. J. Inf. Security, Volume 13, Issue 2, pp 97–111(2014).

[4] Health insurance portability and accountability act of 1996, U.S. Government Printing Office (1996).

[5] Yun W. et al, Multi-authority Based Weighted Attribute Encryption Scheme in Cloud Computing, IEEE International Conference on Natural Computation, (2014).

[6] Kan Y. et al, Attributed-based Access Control for Multi-Authority Systems in Cloud Storage, IEEE International Conference on Distributed Computing Systems, (2012).

[7] Goyal, V., Pandey, O., Sahai, A., Waters, Attribute-based encryption for fine-grained access control of encrypted Data, Proceedings of the 13th ACM Conference on Computer and Communications Security, pp 89–98 (2006).

[8] Boldyreva, A., Goyal, V., Kumar, Identity-based encryption with efficient revocation, Proceedings of the 15th ACM Conference on Computer and Communications Security, pp 417–426 (2008).

[9] Yu, S., Wang, C., Ren, K., Lou, Attribute based data sharing with attribute revocation, Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security, pp 261–270 (2010).

[10] Yu, S., Wang, C., Ren, K., Lou, Achieving secure, scalable, and fine-grained data access control in cloud computing, Proceedings of the 29th IEEE International Conference on Computer Communications, pp 534–542 (2010).

[11] Ibraimi, L., Petkovic, M., Nikova, S., Hartel, P., Jonker, Ciphertext-Policy Attribute-Based Threshold decryption with Flexible Delegation and Revocation of User Attributes. University of Twente, (2009).

Journal of AL-Qadisiyah for computer science and mathematics
Vol.9 No.1 Year 2017



# نظام السلطة المتعددة بالاعتماد على سجل الصحة الشخصية في الحوسبة السحابية

غسان صبيح محمود
جامعة ديالى
كلية العلوم
قسم علوم الحاسوب
ghassan@sciences.uodiyala.edu.iq

طه محمد حسن
جامعة ديالى
كلية العلوم
قسم علوم الحاسوب
dr.tahamh@sciences.uodiyala.edu.iq

ايمن مظهر بدر
جامعة ديالى
كلية القانون والعلوم السياسية
aymen.mudher@sciences.uodiyala.edu.iq

**المستخلص :**

السجلات الصحية الشخصية (PHRs) هي خدمة لتبادل المعلومات الصحية. يمكن للمرضى إنشاء وتحكم وتبادل المعلومات الصحية الخاصة بهم. ويتم الاستعانة بمصادر خارجية ليتم تخزينها في السحابة. ومع ذلك، كانت هناك الخصوصية الخطيرة حول خدمة السحابة كما أنها قد تعرض بيانات المستخدم مثل سجلات الصحية الشخصية لمقدمي الخدمات السحابية أو المستخدمين غير المصرحين لهم. للتغلب على هذه التحديات، السحابة تعتمد من اجل تبادل سجلات الصحية الشخصية بين عدة مستخدمين . في هذا النظام المقترح، المرضى يمكنهم تشفير سجلات الصحية الشخصية وتخزينها في السحابة. وعلاوة على ذلك، المرضى يمكنهم الحفاظ على السيطرة من خلال الوصول إلى سجلاتهم الخاصة بتعيين التحكم بالوصول . لتحقيق الوصول ، السجلات الصحية الشخصية المقترحة تنقسم إلى المجال الشخصي والمجال العام. لضمان الأمن في السحابة على أساس سجلات الصحية الشخصية ، موديل متعدد السلطة بالاعتماد على تشفير سمة الوزن و موديل السمة بالاعتماد على سيطرة الوصول من اجل السلطة المتعددة سوف تنفذان في المجال الخاص و العام على التوالي. الموديل المقترح على أساس سجلات الصحية الشخصية يحسن من كفاءة النضام في التشفير وفك التشفير . كذلك النضام المقترح اثبت انه مقاوم للتواطئ ويحسن الحماية لمستخدمين سجلات الصحية الشخصية في بيئة المالك المتعدد.

**الكلمات المفتاحية :** سجل الصحة الشخصية، سحابة الحوسبة، التشفير، التحكم بالوصول .