Two-layer Ciphertext-Policy Attribute-Based Proxy Re-encryption for Supporting PHR Delegation

G. Wungpornpaiboon[1], S. Vasupongayya[2]
Department of Computer Engineering, Faculty of Engineering, Prince of Songkla University,
Hat Yai, Songkhla 90112, Thailand
E-mail: 5410120022@psu.ac.th[1], vsangsur@coe.psu.ac.th[2]

Abstract— **Medical treatment sometimes requires a case forwarding to a doctor who has a specific expertise. Typically, an electronic medical record (EMS) of a patient can be passed to another doctor without asking the patient because EMS belongs to the healthcare organization. Personal health record (PHR), however, is different because PHR is owned by an individual (e.g., patient) and all accesses to the PHR is controlled by its owner. This work proposes a two-layer ciphertext-policy attribute-based proxy re-encryption scheme (2-layer CP-AB-PRE) for the PHR delegation process. The inner layer policy belongs to the PHR owner while the outer layer policy belongs to the doctors or experts that might want to delegate the PHR to other doctors or experts. This way, the PHR can be delegated to others while the PHR owner still has the control on his/her data. The evaluation results on the size of the resulting ciphertext PHR produced by the proposed method, are shown. The security issues of the proposed method are also discussed.**

Keywords—delegation; access control; personal health record; ciphertext-policy attribute-based encryption; proxy re-encryption

I. INTRODUCTION

Data delegation often happens under the healthcare terms because some symptoms or diseases may require a special expert. A patient may first visit a doctor whom he/she knows. Then, the doctor may consult the case with another doctor who is an expert on such symptom or disease. That is, the medical records related to the case are also passed on to the specialist. Nowadays, the medical data is recorded in electronic format which is called electronic medical record (EMR). Even though the EMR contains the health data related to the patient, the EMR belongs to the associated healthcare organization [1]. Thus, the EMR can be used inside the organization under the terms of usage. Therefore, the doctor can forward the EMR to other doctors without a need to inform the patient. The patient has no authority on the access control over the EMR.

Unlike the EMR, personal health record (PHR) is a personal health data that is collected and controlled by an individual [2]. The PHR owner (i.e., patient) has a full access control over his/her PHR. To provide an accessibility property, the PHR storage is usually an online storage. The PHR owner can define an access policy for each of his/her PHR using an encryption technique such as ciphertext-policy attribute-based encryption (CP-ABE) [3]. Under such encryption scheme, the encrypted PHR can be accessed only by the users who possess the attributes specified by the access policy of that particular PHR. Since the PHR contains all the health related information of an individual, the PHR is useful for the doctor. The PHR may contain all medical histories from various healthcare organizations. This way, the doctor can use such information as a resource for a proper diagnosis.

Under the PHR system, the PHR owner will first allow a doctor to access to his/her PHRs. Now, the doctor wants to delegate his access rights on the PHR to another specialist. To achieve this task under the traditional PHR system, the doctor must first decrypt the PHR because he is allowed to do so according to the PHR owner policy. Next, the doctor must encrypt the PHR with a new policy that will allow the specialist to access such PHR. At this point, the PHR in question is no longer under the PHR owner control. Thus, the PHR concept is now violated.

To resolve such issue, two-layer ciphertext-policy attribute-based proxy re-encryption (2-layer CP-AB-PRE) is proposed in this work. The proposed scheme can provide a PHR delegation feature that allows the PHR owner to have a control over his/her PHR even in the delegation scenario. The two layers are called the outer and the inner layer. The outer layer is controlled by authorized users (e.g., permitted doctor) while the inner layer is controlled by the PHR owner. When the doctor wishes to forward the case to another specialist, the doctor can modify the outer layer policy. Meanwhile, the inner layer policy is still under the control of the PHR owner. The specialist can access the PHR in question if and only if the specialist possesses the attributes that satisfy both layers of the policy. Proxy re-encryption concept [4], [6] is applied in this work to provide a re-encryption process in order to alter the access policy on the ciphertext (i.e., encrypted data) without the need to decrypt the ciphertext.

The paper is organized as follows. First, the related basic concepts are given in Section 2. Then, the details of the proposed method are described in Section 3. Next, the experiments and discussions are given in Section 4. Finally, the conclusion is given in Section 5.

II. BASIC CONCEPTS

A. Proxy Re-encryption Concept

Several models of proxy re-encryption based on attribute-based encryption concept [4], [5] and [6] are reviewed. First, Liang et. al. proposed a proxy re-encryption based on attribute-based encryption scheme (ABE) [4]. Under such scheme, a trusted proxy server is required in order to add a new access policy to the ciphertext. The authorized user who can decrypt the ciphertext can specify a new access policy.

978-1-4673-7825-3/15/$31.00 ©2015 IEEE

Then, the new access policy and the private key of the authorized user are used for generating a re-key. The re-key is sent to the trusted proxy server. Then, the ciphertext will be modified such that the person who satisfies the new access policy can decrypt the ciphertext. Second, Mizuno and Doi proposed another proxy re-encryption to transfer the ABE model into the identity-based encryption model (IBE) [5]. Similar to the prior scheme, the later scheme also requires a trusted proxy server in order to add a new access policy. However the Re-key has different components because each scheme uses a different underlining encryption model (i.e., the ABE versus the IBE). Third, Luo et. al. also proposed another proxy re-encryption scheme however the underlining scheme is the ciphertext-policy attribute-based encryption scheme (CP-ABE) [6]. All these schemes allow authorized users to make changes to an access policy of ciphertext. In other words, the access policy is completely out of the PHR owner control because the authorized user can make changes to the access policy without the PHR owner knowledge.

This work proposes a two-layer ciphertext-policy attribute-based proxy re-encryption (2-layer CP-AB-PRE) to provide a way to perform a PHR delegation while the PHR owner still has a his/her control over his/her PHR. The authorized users (e.g., doctors) can delegate the patient's PHR to a specialist by modifying the outer layer policy while the inner layer policy is still belong to the PHR owner. Any person who satisfied both layers of the policy will be able to access the PHR in question.

B. *Personal Health Record (PHR)*

The Markle Foundation's Connecting for Health Collaborative defines personal health record (PHR) to be the health related information of an individual that is collected and controlled by the individual [7]. The PHR has a potential to reduce the healthcare cost of its owner. For example, a patient may have medical records at several healthcare organizations. Doctors can look for some laboratory test results even when the results are performed at several other organizations because PHR information is collected and owned by the patient. The PHR can be in various formats and domains such as nutrition, sleeping, exercise, treatments, vital sign from devices, or medical histories[8]. The patient usually stores his/her PHR on an online storage (e.g., cloud storage) for availability. Thus, any PHR user (e.g., a doctor or a caregiver) can access the PHR from anywhere at any time. The PHR must be stored in an encrypted format for security purposes. The PHR must be encrypted with a specific access policy defined by its owner. The PHR system must ensure that the access control is enforced.

C. *Proxy Re-encryption Using Ciphertext-Policy Attribute-Based Encryption for PHR*

Ciphertext-policy attribute-based encryption (CP-ABE) [3] is an encryption technique that encrypting a plaintext with an access policy to produce a ciphtertext. The CP-ABE is suitable with the PHR concept because the access control is stuck with the PHR even when it is stored on an untrusted storage (e.g., cloud storage) (see Fig. 1) and it has been used by several PHR related applications [9], [10] and [11]. Similar to the original ABE concept [13], the PHR owner can specify an access policy according to the user's attributes. Fig. 1 shows how the CP-ABE works on a PHR system. According to Fig. 1, the access policy is defined as "doctor" or ("nurse" and "caregiver"). Thus, the user who possesses "doctor" attribute or "nurse" and "caregiver" attribute can satisfy the policy. Each user has an individual private key that is generated from the user attributes. That is, the users' private key that related to "doctor" attribute or "nurse" and "caregiver" attributes satisfy the policy. Then, the users who satisfy the policy will be able to decrypt the encrypted PHR.

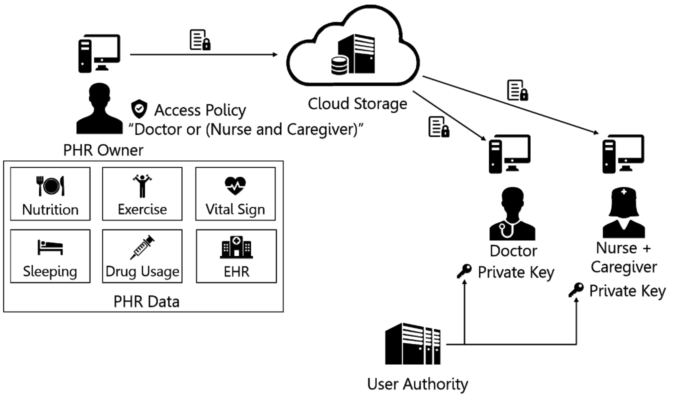

Fig. 1. CP-ABE applied to PHR system

Proxy re-encryption using CP-ABE was proposed in [6]. The proxy re-encryption technique can change the policy without a need to perform any decryption. Typically, the proxy re-encryption scheme requires a trusted proxy server to perform the re-encryption process by using a *Re-key*. The Re-key is generated by an authorized user who wants to change the policy. The authorized user defines a new policy and generates the Re-key with the policy and his/her private key. Then, the Re-key is sent to the trusted proxy server to perform the re-encryption process. As a result, the new ciphertext will be controlled by the new access policy. In other words, the access policy will be completely changed according to the Re-key. Therefore, the access to the PHR can be modified without the PHR owner's permission in some cases.

PHR Delegation happens while some authorized users would like to add some unauthorized people into the policy. This concept is different from the PHR revocation concept because the revocation removes some authorized users from the policy [12]. To add some people into the policy, the access policy must be modified by adding some attributes that those people possess. That is, the PHR must first be decrypted and then re-encrypt with the new policy.

Next the proposed method namely the two-layer ciphter-text-policy attribute-based proxy re-encryption (2-layer CP-AB-PRE) is described.

III. Proposed Method

A. *Overview*

To provide the PHR delegation feature that still allows the PHR owner to have a control over his/her PHR, this work proposes a two-layer ciphertext-policy attribute-based proxy re-encryption (2-layer CP-AB-PRE). The main idea is to add another layer of control for the delegation purpose. That is, the outer layer policy that belongs to the proxy re-encryption method or the delegation while the inner layer policy remains the PHR owner's policy. The CP-ABE library proposed by Bethencort et. al. [3] is used in this work as an underling method. The PHR payload is encrypted using advance encryption standard under cipher block chaining mode (CBC-AES). Thus, there is a *Secret* which can be unlocked from the policy by authorized users. To achieve the delegation purpose, in this work, the *Secret* is now doubled: *R_Secret* and *O_Secret*. The relationship between all three secrets is given in the following equation.

$$R\_Secret \oplus O\_Secret = Secret \quad .....(1)$$

Each secret is separately hidden in each layer of the policy. The *R_Secret* is hidden in the outer layer policy, and the *O_Secret* is hidden in the inner layer policy. The same process can be performed on both layers of the policy including hiding the secret and retrieving the secret. The only difference is the prefix of the attributes used in each layer. The 'R_' prefix is used in the outer layer while the inner layer uses the 'O_' prefix. For explanation purpose, an example is given below.

Mary, the PHR owner, visits Dr. John, a medical doctor. Thus, Mary allows Dr. John to access her PHRs. However, Mary assumes that Dr. John may need to consult with a specialist in another hospital. Therefore, the access policy that Mary defines on her PHRs, sent to Dr. John contains the following components:

- (*R_Doctor* AND *R_John*) to represent a doctor named John
- *O_Doctor* OR (*O_HospitalStaff* AND *O_Specialist*) to represent a doctor or a specialist who is working at a hospital

The outer layer policy contains ('*R_Doctor*' AND '*R_John*') for Dr. John. The inner layer policy contains ('*O_Doctor*' OR ('*O_HospitalStaff*' AND '*O_Specialist*')). The *R_Secret* is hidden in the outer layer policy while the *O_Secret* is hidden in the inner layer policy. Thus, the policy and the PHR payload is shown in Fig. 2

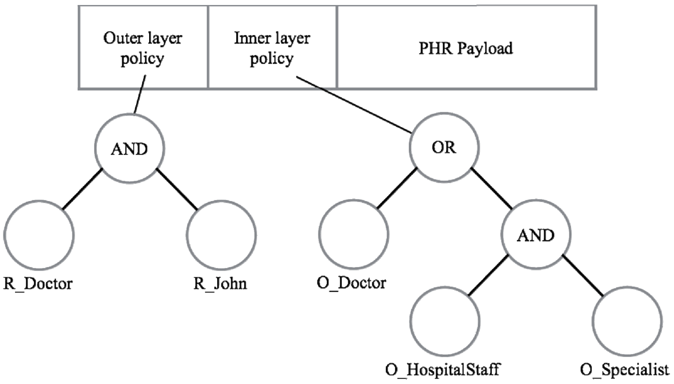

Fig. 2. The policy and the PHR paylod

To decrypt the PHR payload, an individual must possess a set of attributes satisfying both layers of the policy. According to the CP-ABE concept, each individual will be given a private key represented his/her attributes. That is, an individual with the private key containing (*R_Doctor* and *R_John*) and *O_Doctor* can decrypt the PHR payload. In this case, the only person who possesses such attributes is Dr. John.

According to the original CP-ABE, Dr. John will be given a private key with a set of attributes '*Doctor*' and '*John*'. In this work, Dr. John will be given a private key with a set of attributes '*R_Doctor*', '*O_Doctor*', '*R_John*' and '*O_John*' because Dr. John will need attributes for both layers of the policy. Therefore, the original attributes '*Doctor*' and '*John*' are now changed to '*R_Doctor*', '*O_Doctor*', '*R_John*' and '*O_John*'. As a result, the private key of Dr. John will be able to retrieve the *R_Secret* and the *O_Secret* to compute the original *Secret* which is required in order to decrypt the PHR payload.

Now, if Dr. John wants to perform a delegation, Dr. John must first generate a re-key which contains '*R_Doctor*' and '*R_John*' attributes and the new policy which contains '*HospitalStaff*' and '*Specialist*'. The re-key will be sent to the trusted proxy re-encryption server. At this step, the server acts as a broker to retrieve the *R_Secret* from the encrypted PHR on behalf of Dr. John. Next, the retrieved *R_Secret* will be hidden in the new policy. The attributes in the new policy will contain the attributes '*R_HospitalStaff*' and '*R_Specialist*' (see Fig. 3). The original outer layer policy is now replaced by the new outer layer policy.

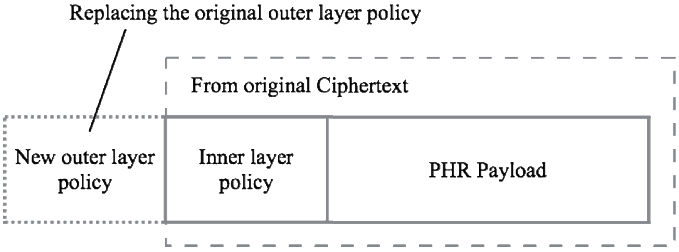

Fig. 3. The new encrypted PHR after the re-encryption process

The re-key contains only '*R_*' prefix attributes and the new policy for re-encrypting process. Thus, the proxy server can only retrieve the *R_Secret* using the re-key. The PHR payload is still safe because the *O_Secret* requires a set of 'O_' attributes that satisfies the PHR owner policy. Thus, the proxy server still cannot decrypt the PHR payload.

B. Hiding secret Process

The secret will be hidden in both layers of the policy according to the CP-ABE. The underlining mathematic is presented in [3]. An example is given in this section for explanation purposes. The policy is represented as an access tree with an attribute at each leaf node while an operation is represented by a non-leaf node. Fig. 4 shows the access tree of the inner layer policy described above. Each node can be assigned a unique integer.

Next, each node will be defined a node equation. The condition of equation is represented as following;

a) *If the node is a leaf node or represents an 'OR operation' node then* $f(x) = x^0$

b) *If the node represents an 'AND operation' node*

$$f(x) = x^k + x^{k-1} + \ldots + x^0$$
$$, where\ k\ is\ the\ number\ of\ child\ node - 1$$

Fig. 5 shows the resulting equations from the access tree shown in Fig. 4.

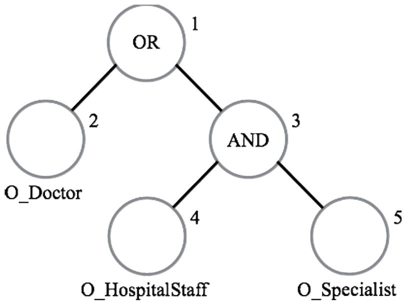

Fig. 4. The inner layer policy

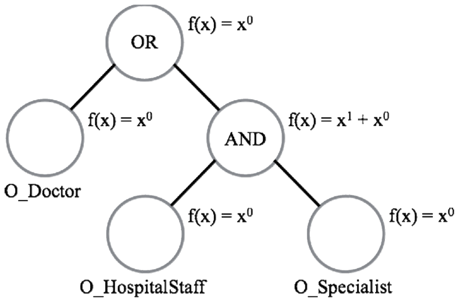

Fig. 5. The access tree with node equations

The secret hiding mechanism starts from top (root node) to bottom (leaf node). The rules to hide a secret are:

a) *If the parent node represents 'OR operation'*

$$x_{child}{}^0 = x_{parent}{}^0$$

b) *If the parent node represents 'AND operation'*

$$x_{child}{}^0 = f_{parent}(unique\ number_{child})$$

Fig. 6 shows the resulting access tree assuming the secret is 5. Thus, the root node is 5. Node 2 is a leaf node of the 'OR operation' parent node; the first rule above is applied; Node 2 is also 5. Node 3 is a non-leaf node but its parent is the 'OR operation' node; the first rule above is also applied; Node 3 is $x^1 + 5$. Node 4 is a leaf node of the 'AND operation' parent node; The second rule above is applied; Node 4 is $f(4)$ which is $4 + 5$ or 9. Node 5 is a leaf node of the 'AND operation' parent node; The second rule above is applied; Node 5 is $f(5)$ which is $5 + 5$ or 10.

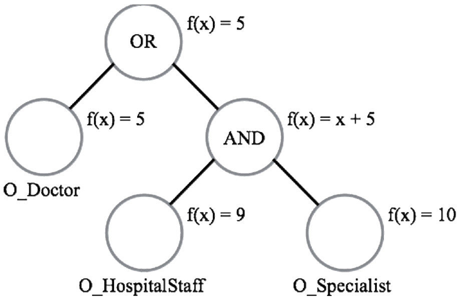

Fig. 6. The access tree with with a hidden secret 5

The leaf nodes will contain the public component of each attribute. These public components will be used during the secret retrieving process. The final access tree is shown in Fig. 7.

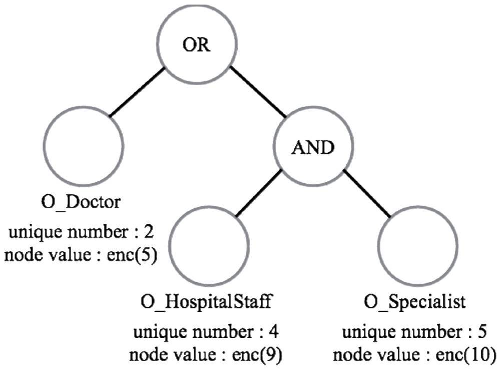

Fig. 7. The final access tree with the secret and public components

C. Secret Retrieving Process

To decrypt the PHR payload, the user must possess the attributes that satisfy the access policy. The secret hidden in the access policy can be retrieved using the user's private key. Each user will be given the private key containing his/her private component of each attribute that is related to the public component of the same attribute. The leaf node is encrypted with the public component that can be decrypted by the related private comment from the user's private key. Fig. 8 shows how the scheme matches each leaf node with the attribute in the user's private key. The private component of the matching attribute will be able to decrypt the matching leaf node and retrieve the secret. For example, Node 4 and Node 5 will be decrypted by the specialist private key with the return values of 9 and 10, respectively.

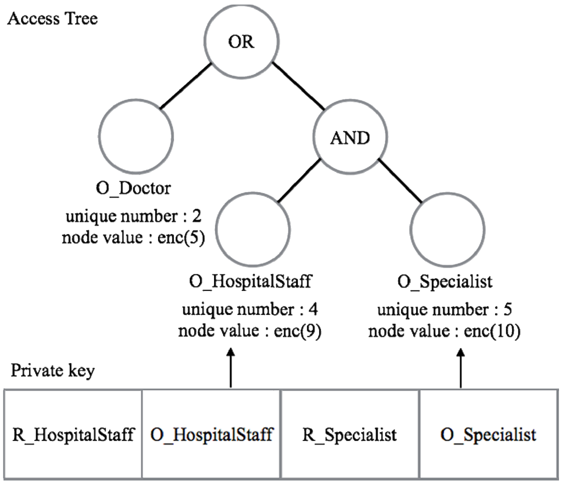

Fig. 8. The private key is used to retrieve the secret

Next, the scheme will calculate the value of the parent node using the following rules:

a) *If parent node represents 'OR operation'*

The parent node will be a duplicate of any node equation from its child node if one of them is defined.

$$f_{parent}(x) = f_{child}(x)$$

b) If parent node represents 'AND operation'

Each node can be defined as a 2-tuple value $(x, y) = (unique\ number, node\ value)$. The parent node can be calculated as a node equation from every child nodes using Lagrange Interpolating Polynomial equation as follow:

$$f_{parent}(x) = \frac{(x-x_1)(x-x_2)\cdots(x-x_n)}{(x_0-x_1)(x_0-x_2)\cdots(x_0-x_n)}y_0$$
$$+\frac{(x-x_1)(x-x_2)\cdots(x-x_n)}{(x_1-x_0)(x_1-x_2)\cdots(x_1-x_n)}y_1$$
$$+\cdots$$
$$+\frac{(x-x_1)(x-x_2)\cdots(x-x_n)}{(x_n-x_0)(x_n-x_1)\cdots(x_n-x_{n-1})}y_n$$
$$, where\ n\ is\ the\ number\ of\ child\ nodes - 1$$

From the above example, the two leaf nodes are $(4, 9)$ and $(5, 10)$. The parent of both leaf nodes represents AND operation. Therefore, the node equation is calculated using Lagrange Interpolating Polynomial equation as following;

Let $(x_0, y_0) = (4, 9)$ and $(x_1, y_1) = (5, 10)$

$$f_{parent}(x) = \frac{(x-5)}{(4-5)}(9) + \frac{(x-4)}{(5-4)}(10)$$
$$f_{parent}(x) = (-9x + 45) + (10x - 40)$$
$$f_{parent}(x) = x + 5$$

The process is repeated until the root node is reached. As the next parent node is root node that represents the OR Gate, the root node can duplicate the node equation from the child node (i.e., the previous parent node). Finally, we got the node equation of root node. Next, we can calculate the hidden secret according to the following equation.

$$The\ hidden\ secret = f_{Root}(0)$$

Thus, the *O_Secret* hidden in the above access tree can be calculated as follow:

$$O\_Secret = f_{Root}(0)$$
$$O\_Secret = 0 + 5$$
$$O\_Secret = 5$$

Note that the propose scheme uses 2-layer policy. Thus, the user must possess both *R_Secret* and *O_Secret* in order to calculate the actual *Secret.*

D. Proxy Re-encryption Process

A proxy server performs a proxy re-encryption using the Re-key. Dr. John sends the Re-key and the related PHR to the proxy server through a secure communication channel in order to delegate the rights to a specialist at the hospital. Once the proxy server receives the Re-key, the server retrieves the *R_Secret* from the outer layer policy of the encrypted PHR in question (the retrieving process is explained above). Next, the *R_Secret* is hidden into a new access tree representing the new access policy. The secret hidden process is also explained in details above. This new access tree will contain two leaf nodes. The first leaf node will contain the attribute *R_Specialist* while the second leaf node will contain the attribute *R_HospitalStaff.* Finally, the new outer layer policy will replace the old outer layer policy (see Fig. 3). The inner layer policy and the PHR payload remain unchanged.

IV. Evaluation and Discussion

The proposed scheme is compared against three attribute-based proxy re-encryption models [4], [5] and [6] in terms of the data size. IV.a)shows the comparison results. As the proposed scheme duplicates the policy into two layers, the size of the policy is double as a tradeoff. However, the increasing data size is still practical. The size of the public key, master key, ciphertext, re-encrypted ciphertext, private key and re-encryption key will be evaluated. The data size is represented in terms of the components including g, $e(g, g)$, Zp and C. The definition of each component is given below.

a) g *generates the component with* N_g

b) $e(g, g)$ *generates the component with* N_e

c) Z_p *randoms the component with* N_p

d) C *is a constant number*

TABLE I. Data Size Comparison with Related Schemes

Data	**Size**			
	Liang et. al. [4]	***Mizuno and Doi [5]***	***Luo et. al. [6]***	***Our scheme***
Public key	N_e+ (3n) N_g + C	N_e+ (3n+3) N_g + C	N_e+ (2n+nj+3) N_g + C	N_e+ (2n+2) N_g + C
Master key	(3n+1) N_p	(3n+2) N_p	(2n + nj + 1) N_p	(2) N_p+ (2) N_g
Ciphertext	(n) N_p+ N_e+ (n+2) N_g	(n) N_p+ N_e+ (n+1) N_g	(n) N_p+ N_e+ (n+2) N_g	(n+1) N_p+ (2) N_e+ (2n + 4) N_g
Re-encrypted Ciphertext	(2n) N_p+ (n+2) N_e+ N_g + C	N_e+ (3) N_g	(2n) N_p+ (n+2) N_e+ N_g + C	(2n) N_p+ (2) N_e+ (4n + 2) N_g
Private key	(n) N_p+ (2n+1) N_g	(n) N_p+ (2n+3) N_g	(n) N_p+ (4n+1) N_g	(n) N_p+ (4n+2) N_g
Re-encryption Key	(2n) N_p+ (2n+1) N_g + C	(4n+2) N_g + N_e	(2n) N_p+ (4n+1) N_g + C	(2n) N_p+ (4n+2) N_g

A. Public key and Master Key

The public key contains the public components that related to the private components in the private key. In [4] and [5], the public key and the private key are approximately $O(n)$, where n represents the number of attributes in the policy. Under [3], the size is larger than the others because it is designed to support the attributes with multi-value. Thus, the public and private components of each value is approximately $O(n^2)$. Thus, the proposed scheme uses the public key and the master key size within the range of other works.

B. Ciphertext

The ciphertext contains the policy and the PHR payload. Our ciphertext seems bigger than the others because we duplicate the policy into two-layer. The component N_e and N_g size is doubled to cover both layers of the policy. However, the size of the ciphertext is still $O(n)$, similar to other works.

C. Re-encrypted Ciphertext

The re-encrypted ciphertext in [5] has the smallest data size because of the underlining IBE model. The IBE ciphertext contains only one identity for only a specific user and it has no policy within the ciphertext. Unlike [5], the proposed scheme still packs the policy within the ciphertext. The [4] and [6] also add the proxy component that makes the ciphertext bigger. However, the size of the re-encrypted ciphertext (in [4], [6] and the proposed scheme) can still be represented as $O(n)$.

D. Private key

The actual size of the private key under the proposed scheme is largest because it must contain both the 'R_' and the 'O_' of each attribute. However, the size of the private key under the proposed scheme is still $O(n)$.

E. Re-encryption key

The re-encryption key is used to modify the ciphertext to add the new access rights. Usually, the private components from the authorized users and the new policy are included in the re-encryption key. All schemes have $O(n)$ as the size of the re-encryption key.

V. Conclusions

This work proposes a PHR delegation scheme to allow the PHR owner to have a control over his/her PHR even during the delegation process. The CP-ABE proxy re-encryption scheme is modified to support such feature by duplicating the access policy into two layers. The outer layer is controlled by the authorized user for delegating to others. The inner layer is still belonging to the PHR owner. In doing so, the size of the public key, the master secret key, the ciphertext, the re-encrypted ciphertext, the private key and the re-encryption key increased. However, the size of all these components is still in the range of other works and practical in the real scenario.

Under the proposed scheme, the original secret will be separated into two pieces: *R_Secret* and *O_Secret* according to equation 1. The *R_Secret* is hidden in the outer layer policy while the *O_Secret* is hidden in the inner layer policy. The authorized user can delegate the encrypted PHR to another person by generating a re-key containing the authorized user R_ attributes and the new policy. The proxy server which is the trusted server retrieves the *R_Secret* from the outer layer policy on behalf of the authorized user in order to create the new policy. Then, the new policy is replacing the old outer layer policy. The inner layer policy and the PHR payload remain unchanged. The user who possesses the attributes satisfies both layers of the policy will be able to compute the original *Secret* for decrypting the PHR. The PHR owner can modify the inner layer policy if he/she would like to change the access policy scope to allow or disallow the delegation process.

The PHR payload is still safe under the proposed scheme because the attacker still need to possess the *O_Secret*. The re-key contains only half a secret. However, the proxy server must be trusted and the re-encryption key must be secure. If the malicious users possess the re-encryption key, they can make changes to the outer layer policy. Thus, the transmission of the re-encrypted ciphertext process must be protected. Furthermore, the PHR storage must ensure that the users can only access the latest version of the PHR and the PHR owner must be able to re-version the access policy of his/her PHR.

Acknowledgment

This work is supported in part by the National Research University Funding no. MED540548S. The authors would like to thank Centre for Network Research (CNR) lab at Prince of Songkla University for providing equipments and facilities.

References

[1] L. Florman, 'Electronic Medical Record', The Portable Medical Mentor, pp. 153-155, 2014.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68-73.

[2] P. Tang, J. Ash, D. Bates, J. Overhage and D. Sands, 'Personal Health Records: Definitions, Benefits, and Strategies for Overcoming Barriers to Adoption', *Journal of the American Medical Informatics Association, vol. 13, no. 2, pp. 121-126*, 2006.

[3] J. Bethencourt, A. Sahai and B. Waters, 'Ciphertext-Policy Attribute-Based Encryption', *2007 IEEE Symposium on Security and Privacy (SP '07)*, 2007.

[4] X. Liang, Z. Cao, H. Lin and J. Shao, 'Attribute based proxy re-encryption with delegating capabilities', *Proceedings of the 4th International Symposium on Information, Computer, and Communications Security - ASIACCS '09*, 2009.

[5] T. Mizuno and H. Doi, 'Hybrid Proxy Re-encryption Scheme for Attribute-Based Encryption', *Information Security and Cryptology, pp. 288-302*, 2010.

[6] S. Luo, J. Hu and Z. Chen, 'Ciphertext Policy Attribute-Based Proxy Re-encryption', *Information and Communications Security, pp. 401-415*, 2010.

[7] Connecting for Health. The personal health working group final report. *Markle Foundation*; 2003 Jul 1.

[8] I. Iakovidis, 'Towards personal health record: current situation, obstacles and trends in implementation of electronic healthcare record in Europe', *International Journal of Medical Informatics, vol. 52, no. 1-3, pp. 105-115*, 1998.

[9] P. Thummavet and S. Vasupongayya, 'A novel personal health record system for handling emergency situations', *2013 International Computer Science and Engineering Conference (ICSEC)*, 2013.

[10] N. Xavier and V. Chandrasekar, 'Cloud Computing Data Security for Personal Health Record by Using Attribute Based Encryption', *Business and Management vol. 7, no. 1, pp. 209-214,* 2015.

[11] F. Xhafa, J. Feng, Y. Zhang, X. Chen and J. Li, 'Privacy-aware attribute-based PHR sharing with user accountability in cloud computing', *J Supercomput, vol. 71, no. 5, pp. 1607-1619*, 2014.

[12] X. Liang, R. Lu, X. Lin, and X. S. Shen, 'Ciphertext policy attribute based encryption with efficient revocation', *Technical Report, University of Waterloo*, 2010.

[13] A. Sahai and B. Waters, 'Fuzzy Identity-Based Encryption', Lecture Notes in Computer Science, pp. 457-473, 2005.