A Ciphertext Policy Attributes-based Encryption Scheme with Policy Revocation

Phyo Wah Wah Myint, Swe Zin Hlaing, Ei Chaw Htoon
University of Information Technology
Yangon, Myanmar
phyowahwah@uit.edu.mm, swezin@uit.edu.mm, eichawhtoon@uit.edu.mm

Abstract

There are a lot of *data exchanges among the parties by using cloud computing. So data protection is very important in cloud security environment. Especially, data protection is needed for all organization by security services against unauthorized accesses. There are many security mechanisms for data protection. Attributes-based Encryption (ABE) is a one-to-many encryption to encrypt and decrypt data based on user attributes in which the secret key of a user and the ciphertext are dependent upon attributes. Ciphertext policy attributes-based encryption (CP-ABE), an improvement of ABE schemes performs an access control of security mechanisms for cloud storage. In this paper, sensitive parts of personal health records (PHRs) are encrypted by ABE with the help of CP-ABE. Moreover, an attributes-based policy revocation case is considered as well as user revocation and it needs to generate a new secret key. In proposed policy revocation case, PHRs owner changes attributes policy to update available user lists. A trusted authority (TA) is used to issue secret keys as a third party. This paper emphasizes on key management and it also improves attributes policy management and user revocation. Proposed scheme provides a full control on data owner as much as he changes policy. It supports a flexible policy revocation in CP-ABE and it saves time consuming by comparing with traditional CP-ABE.*

Keywords- Attributes-based encryption (ABE), Ciphertext policy attributes-based encryption (CP-ABE), Personal Health Records (PHRs), Trusted Authority (TA)

1. Introduction

Modern societies and organizations are motivated to outsource more and more sensitive information into the cloud servers. Protecting data from unauthorized users and other threats is a very important task for security providers. ABE performs as an attributes-based access control with an encryption mechanism for data confidentiality. ABE allows users to encrypt and decrypt data based on user's attributes. In ABE, if the attributes of a user satisfy an access structure of ciphertext, the user can get a secret key associated with that ciphertext. Collusion-resistance is crucial security feature of ABEs. Another modified form of ABE is Key-Policy ABE (KP-ABE) as shown in Figure 1. In KP-ABE scheme, data owner cannot decide a user who can decrypt the encrypted data. The problem is that it can only choose descriptive attributes for the data [4] [10]. Then, another modified form of ABE is CP-ABE as shown in Figure 2. CP-ABE improves the existing ABEs because the encryptor can choose the decryptor who can decrypt a cipher. It can support an access control in the real environment [4] [10]. CP-ABE has still limitations in terms of specifying policies and managing user attributes [4] [6]. In this paper, a policy revocation scheme is added in traditional CP-ABE scheme. Traditional CP-ABE scheme has not considered policy revocation case. A sample PHRs data sharing scenario is shown in Figure 3. For PHRs data sharing in a health care organization, the two cases are considered such as *simple users case* (i.e., PHR owner has never changed access policy for his users yet) and *policy revocation case* (i.e., PHR owner has changed access policy for his users). In this paper, traditional CP-ABE is used for simple users case and proposed scheme is used for policy revocation case. The PHRs owner may be a data administrator of the whole health care organization who manages PHRs. User may be anyone who is interested in different fields of health care organization (i.e., researchers, staffs, physicians, lab members, nurse, hospital head, and so on.). For this PHRs data sharing, traditional CP-ABE uses a symmetric secret key for encryption/decryption phase. When a policy revocation occurs, proposed scheme uses an updated secret key which is generated by TA according to a new access policy. Both of schemes use an Advanced Encryption Standard (AES) function to encrypt/decrypt PHRs data by using a different secret key. This paper presents a comparison for time measurements of both schemes. Different procedures of encryption, decryption, and key generation algorithms for both schemes are explained in section 4.2. This paper emphasizes on the attributes policy management, key management and supports a flexible policy updating access control. It considers to reduce encryption/decryption times comparing with traditional CP-ABE. Section 2 discusses the related work for ABEs literature reviews. Section 3 describes preliminaries for this paper. Section 4 presents a CP-ABE scheme with

policy revocation. Section 5 includes experimental results. Section 6 describes conclusion and further extensions, and finally includes the references.

2. Related Work

Researchers have described the problems occurred in ABE schemes in various ways. Bethencourt et al. proposed CP-ABE by additional consideration for a delegation on an essential attribute structure [1]. They have improved ABE features but they had limitations that it was proved secure under the generic group heuristic and has not considered policy revocation yet [1]. Li et al. studied a survey on ABE scheme of data access control in cloud computing [6]. They listed some unsolved issues of existing schemes such as key management, flexible access and efficient user revocation challenges. They proposed a new scheme Categorical Heuristics on ABE (CHABE). CHABE describes a message and a predicate over the universe of attributes. The attributes set satisfies the predicate, endorsed the message. However, it needs to keep the predicate and message pairs over the universe of attributes in database on server [6]. Yu et al. proposed a combining technique of ABE, proxy re-encryption, and lazy re-encryption technique to achieve a fine-grained data access control in cloud computing [12]. It had multiple system operations and computation on cloud servers which is proportional to the number of system attributes. Ibraimi et al. proposed an encryption scheme for a secure policy updating [5]. They have shown an open problem to provide security proof and to break that scheme for reducing a well-studied complexity-theoretic problem. Wungpornpaiboon and Vasupongayya proposed two-layer ciphertext-policy attribute-based proxy re-encryption for supporting PHR delegation [11]. In [11], the encryption layer is divided into two layers such as inner and outer layer. The inner layer is possessed by data owner and the delegation is processed by satisfying an access structure in the outer layer. Chen and Ma proposed efficient decentralized attribute-based access control for cloud storage with user revocation [2]. It did not need any central authority and coordination among multiple authorities. The authors proposed to consider the user revocation for more practical. Mo and Lin proposed a dynamic re-encrypted ciphertext-policy attributed-based encryption scheme for cloud storage [9]. The authors proposed to consider for re- encryption the ciphertext by using re-key in case of attribute revocation or delegation by delegator. In [9], the re-encryption case was moved to the cloud side to make the data management of the data owner simpler. If that scheme is used, user needs to trust the cloud side. Li et al. proposed flexible and fine-grained attribute-based data storage in cloud computing [7]. In [7], the authors proposed a fine- grained access control (ABE) scheme with efficient user revocation for cloud storage system. The issue of user revocation could be solved by introducing the concept of user group. When any user leaves, the group manager updates users' private keys except for those who have been revoked [7]. In [8], Myint et al. proposed a flexible policy updating access control scheme for cloud storage but it has still an ongoing work for key exchanges and analysis on conventional ABEs. Cui et al. [3] introduced an expressive CP-ABE with partially hidden access structures. Each attribute is divided into an attribute name and an attribute value, and attribute values of the attributes in an access structure are not given in the ciphertext [3].

3. Preliminaries

This section initially describes a number of concepts that provides the basis for proposed scheme.

3.1. CP-ABE Scheme

The four polynomial time algorithms in CP-ABE are as follows:

- Setup (λ, U): This algorithm takes as input the initial information λ such as security parameter and attributes universe description U, and outputs a public key PK and master secret key MK.
- Encrypt (PK, M, $A_{C\text{-}CP}$): This algorithm takes as input PK, plaintext message M and attributes access policy $A_{C\text{-}CP}$. It outputs the ciphertext C associated with $A_{C\text{-}CP}$.
- KeyGen (PK, MK, A_u): This algorithm takes as input PK, MK and access policy of user A_u then outputs a secret key SK.
- Decrypt (PK, SK, C): This algorithm takes as input PK, SK and C then outputs the plaintext message M if and only if A_u satisfies $A_{C\text{-}CP}$ associated with the ciphertext C.

The above algorithms are illustrated in Figure 2. For KP-ABE illustration in Figure 1, $A_{u\text{-}KP}$ is denoted by an access policy of user. A_C is denoted by a descriptive attributes set for a data owner. In KP-ABE, A_C needs to satisfy the structure of $A_{u\text{-}KP}$ as shown in Figure 1. CP-ABE improves the limitation of KP-ABE scheme. In CP-ABE, the data owner has full right on defining access policy before encrypting the message.

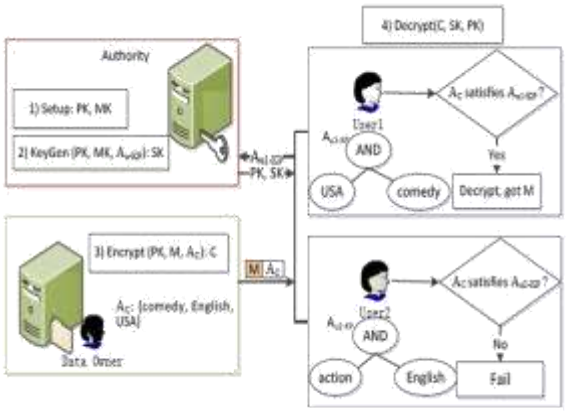

Figure 1. KP-ABE illustration [6]

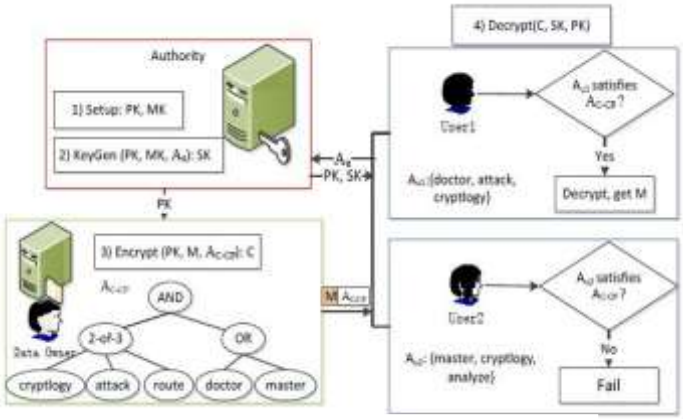

Figure 2. CP-ABE illustration [6]

3.2. Bilinear Maps

The proposed scheme is based on pairings over groups of prime order. Let G_0 and G_1 be two multiplicative cyclic groups of prime order p, g be a generator of G_0, and Z_p be the additive group associated with integers from {0, … , p-1}. A pairing or bilinear map e: $G_0 \times G_0 \rightarrow G_1$ satisfies the following properties:

1. Bilinearity: for all u, v $\in G_0$ and a, b $\in Z_p$, we have $e(u^a,v^b) = e(u, v)^{ab}$.
2. Non-degeneracy: $e(g, g) \neq 1$. Observe that bilinear map also enjoys the symmetry property, i.e. $e(g^a,g^b) = e(g,g)^{ab} = e(g^b,g^a)$. Group G_0 is said to be a *bilinear group* if the group operation in G_0 and the bilinear map e: $G_0 \times G_0 \rightarrow G_1$ can be computed efficiently.

3.3. Access Tree

Another important concept used in this paper is the concept of an access tree. Let T be an access tree associated with an access policy. A leaf node k in the access tree T represents an attribute from the attribute set $w \in \Omega$, where Ω is a universe of attributes. A non-leaf node k in T represents a threshold gate, which is described by its child nodes and a threshold value. Let num_k be the number of children of a node k and T_k be its threshold value, then $0 < T_k < num_k$. If $T_k = 1$, then k corresponds to an OR gate; if $T_k = num_k$, the node k is an AND gate. For leaf nodes, $T_k = 1$.

4. CP-ABE Scheme with Policy Revocation

This section describes the system structure and system algorithms to implement the proposed methods.

4.1. System Structure

The proposed system structure is shown in Figure 3.

The four entities in proposed system structure are as follows:

- Trusted Authority (TA): An entity which is trusted by all other participating entities in this system. It is responsible for issuing keys to the users upon valid requests.
- Data Owner (DO): The entity who owns data and encrypts those data.
- Data User (DU): The entity who would like to access encrypted data with proper authorization.
- Cloud Storage Provider (CSP): The entity that will provide storage service to store encrypted data.

In Figure 3, PHRs administrator or a DO encrypts each PHR content associated with each policy respectively. The ciphers of PHRs are stored in a cloud server which is a CSP. User or a DU tries to access PHR cipher by proving his credential attributes. A sample PHR training dataset is used in proposed system. A PHR training data consists of an attributes set in which PatientID, Name, NRC, Address, Phone, Disease, Hospital, PolicyID, RevokedPolicyID, and so on. Among these attributes, PolicyID is used for an access control to grant or deny users' accesses. PolicyID is a unique identity number which represents a threshold of three attributes per policy. Each policy consists of the three attributes such as Role, Field, and Hospital. For example, PHRs administrator encrypts a PHR content 'XXX' by defining PolicyID = '15'. Suppose that PolicyID = '15' represents a threshold for " Role = 'Lab member', Field = 'Allergy and Immunology', and Hospital = 'SSC' ". If user has a role of 'Lab member', 'Allergy and Immunology' field, and hospital name 'SSC' in his threshold attributes as in PolicyID = '15', he can decrypt the cipher of 'XXX'. If PHRs administrator changes any attribute in PolicyID = '15' for 'XXX', it means that PolicyID = '15' is updated to another policy identity number for 'XXX'. To prevent collusion attack, the users associated with an old policy '15' for 'XXX' who have to be revoked.

Figure 3. A scenario of PHR data sharing by using traditional CP-ABE and CP-ABE with proposed scheme

4.2. Proposed Algorithms

Algorithms for encryption, key generation and decryption in CP-ABE with proposed policy revocation are as shown in Figure 4, Figure 5 and Figure 6 respectively. Table 1 shows symbols and meanings of proposed algorithms.

Algorithm 1: Encryption algorithm for both simple user case and policy revocation case

Input : M_{PHR}

Output : C_{PHR}

1. Initialize PHR_Curr_Pol = Pol_{id}
2. If Revo_Pol = NULL && $Status_{id}$ = NULL then
3. C_{PHR} = Enc(M_{PHR}, PHR_Curr_Pol, SK)
4. Else
5. Revo_Pol = PHR_Curr_Pol
6. PHR_Curr_Pol = New_Pol_{id}
7. U_{id} = PHR_{id}
8. Upd_{SK} = KeyGen(U_{id}, Upd_{level})
9. C_{PHR} =Enc(M_{PHR},Revo_Pol,PHR_Curr_Pol,Upd_{SK})
10. End If

Figure 4. Encryption algorithm

Algorithm 2: Key generation algorithm for updated secret key (policy revocation case)

Input : U_{id}, Upd_{level}

Output : Upd_{SK}

1. Set up a unique value to $Status_{id}$ according to Upd_{level}
2. SK_{token} = $Status_{id}$ + U_{id}
3. Upd_{SK} = GetHashCode(SK_{token})
4. Return Upd_{SK}

Figure 5. Proposed key generation algorithm by TA

Algorithm 3: Decryption algorithm for both simple user case and policy revocation case

Input : C_{PHR}

Output : M_{PHR}

1. Initialize User_CurrPol = U_Pol_{id}
2. If Revo_Pol = NULL && User_CurrPol = PHR_Curr_Pol then
3. M_{PHR} = Dec(C_{PHR}, User_CurrPol, SK)
4. Else If Revo_Pol = NULL && User_CurrPol ≠ PHR_Curr_Pol then
5. Notify “Unauthorized Access!”
6. Else If Revo_Pol ≠ NULL && User_CurrPol ≠ PHR_Curr_Pol then
7. Notify “Unauthorized Access!”
8. Else If Revo_Pol ≠ NULL && User_CurrPol = PHR_Curr_Pol && $User_{Gid}$ = $Revoked\_User_{Gid}$ then
9. Notify “You have been revoked. Don’t try a collusion attack!”
10. Else If Revo_Pol ≠ NULL && User_CurrPol = PHR_Curr_Pol && $User_{Gid}$ ≠ $Revoked\_User_{Gid}$ then
11. Upd_{SK} = KeyGen(U_{id}, Upd_{level})
12. M_{PHR} = Dec(C_{PHR}, Revo_Pol , PHR_Curr_Pol, Upd_{SK})
13. End If

Figure 6. Decryption algorithm

Table 1. Symbols and meanings in proposed algorithms

Symbols	Meanings
PHR	Personal Health Record
M_{PHR}	PHR content data
C_{PHR}	Ciphertext of PHR content
PHR_Curr_Pol	Current policy identity number of M_{PHR}
Pol_{id}	A unique policy identity number for M_{PHR} which is defined by PHR

	owner
Revo_Pol	A policy identity number which is revoked by PHR owner
$Status_{id}$	A unique predefined identity number for updating status according to a policy updating level Upd_{level} (There are four Upd_{level}, so four $Status_{id}$ are predefined for updating status.)
SK	A symmetric secret key (for simple users case)
New_ Pol_{id}	A new unique policy identity number of M_{PHR} (i.e., old policy identity number of M_{PHR} is revoked by PHR owner)
U_{id}	A unique user identity which performs as a temp to keep PHR_{id}
PHR_{id}	A unique identity number of M_{PHR} in PHRs dataset
Upd_{SK}	A unique updated secret key (for policy revocation case)
KeyGen	Key generation function by TA
Upd_{level}	A policy updating level (Four types of Upd_{level} are 'All-Attributes-changes' in policy, 'BelowTheHalf-Attributes-changes' in policy, 'OverTheHalf-Attributes-changes' in policy and 'ByName-changes' in policy.)
Enc	AES encryption function
SK_{token}	A unique secret token key
GetHashCode	A MD5 hash function
TA	Trusted Authority which generates Upd_{SK}
User_CurrPol	Current policy identity number of user
U_Pol_{id}	A unique policy identity number which is proved by user
Dec	AES decryption function
$User_{Gid}$	A unique global identity number of user
Revoked_$User_{Gid}$	$User_{Gid}$ of a revoked user

5. Experimental Results

This section shows experimental results for traditional CP-ABE by comparing with proposed policy revocation in CP-ABE. All of the experimental results are carried out for each PHR data of a sample training PHRs dataset. Training PHR data was explained in previous section. These experiments are configured on a machine of Intel CORE i3 processor, 4GB of RAM, 500GB of HDD and CPU 2.30GHz on Windows7 Ultimate system. It runs on the software version of Microsoft Visual Studio 12.0. Figure 7 shows the measurements of performance evaluation for key generation time, encryption time and decryption time respectively. All of the performance evaluations are measured by an average execution time after testing five times on system algorithms. The key generation time for 3 leaf nodes per policy takes 0.482 seconds by traditional CP-ABE and it takes 0.962 seconds by proposed scheme respectively. The running times for CP-ABE are almost perfectly linear with respect to the numbers of leaf nodes in an access policy. The key generation time for CP-ABE with proposed scheme is longer than traditional CP-ABE because proposed scheme includes an extra consideration to generate an updated secret key for detecting and protecting revoked users according to policy revocation as shown in Figure 7(a). However, longer key generation time is not a weakness for proposed scheme because both encryption and decryption times of proposed scheme are less than traditional scheme. The encryption and decryption times per 10 leaf nodes are 0.37 seconds and 0.34 seconds by traditional CP-ABE. In proposed scheme, the encryption and decryption times per 10 leaf nodes are 0.34 seconds and 0.311 seconds. In proposed algorithms, both encryption and decryption algorithms firstly call the key generation algorithm, secondly take a corresponding key either from TA (in proposed scheme) or from a simple key generator (in traditional scheme). Thirdly, call AES function with the help of CP-ABE. Fourthly, AES inputs that corresponding secret key to do encryption/decryption, and AES transforms it to a system secret key, and then execute corresponding outputs. According to the difference between an old secret key and an updated secret key, the execution times for encryption/decryption are also changed in both schemes. Hence, proposed scheme saves time consuming on overall evaluation for encryption/decryption outputs (which includes calling key generation phase and returning a key) as shown in Figure 7(b) and 7(c). It supports a flexible policy revocation control for user and PHRs owner and it also performs key exchange management, policy management and revoked user detection.

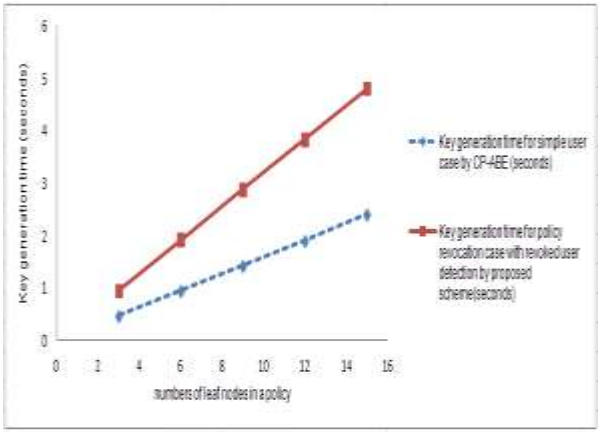

Figure 7(a). Key generation time

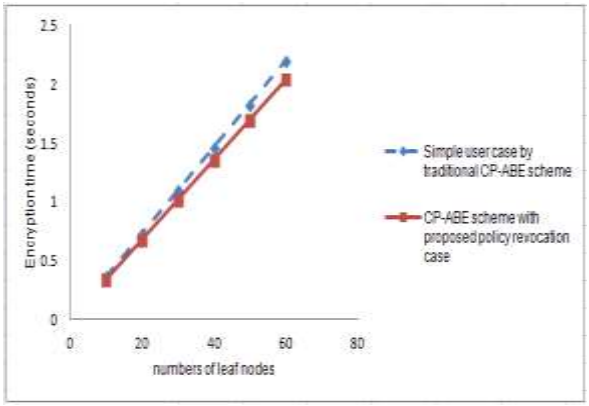

Figure 7(b). Encryption time

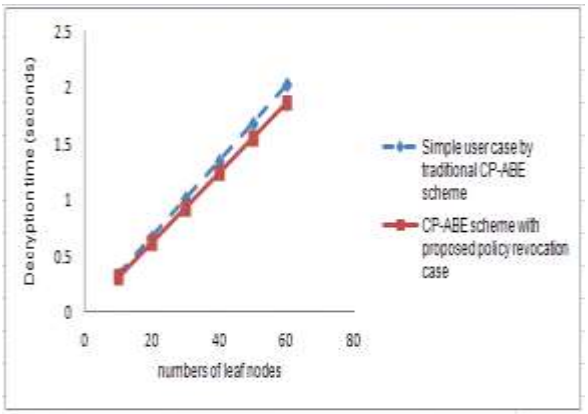

Figure 7(c). Decryption time

6. Conclusion and Future Work

The proposed policy revocation scheme adapts and solves the key management problem for CP-ABE. It focuses on the efficient access policy updating by data owner according to new access policy or policy revocation. It considers generating an updated secret key for encrypting/decrypting the updated PHR. It intends to be more flexible policy management and overall time safe by doing full right on data owner. It is going to study multi authority domains for data protection in cloud storage. As a future work, it is going to do performance analysis by comparing proposed scheme with the existing enhanced CP-ABE schemes.

7. References

[1] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-Policy Attribute-Based Encryption", in Proc. of IEEE Symposium on Security and Privacy (SP'07), 321-334, IEEE Computer Society Washington, DC, USA, May 20-23, 2007.

[2] J. Chen, and H. Ma., "Efficient Decentralized Attribute-based Access Control for Cloud Storage with User Revocation", in Proc. of IEEE International Conference on Communications (ICC), Sydney, NSW, Australia, 3782-3787, Jun. 10-14, 2014.

[3] H. Cui, R. H. Deng, J. Lai, X. Yi, and S. Nepal, "An Efficient and Expressive Ciphertext-policy Attribute-based Encryption Scheme with Partially Hidden Access Structures, revisited", in Proc. of Computer Networks, 133(2018), 157-165, Mar. 14, 2018.

[4] M. George, C. S. Gnanadhas, and S. K, "A Survey on Attribute Based Encryption Scheme in Cloud Computing", in Proc. of International Journal of Advanced Research in Computer and Communication, 2(11), 4408-4412, Nov., 2013.

[5] L. Ibraimi, M. Asim, M. Petkovic, and B. Waters, "An Encryption Scheme For A Secure Policy Updating", in Proc. of the 5th International Conference on Security and Cryptography(SECRYPT 2010), Athens, Greece, Jul. 26-28, 399-408, 2010.

[6] T. Li, L. Hu, Y. Li, J. Chu, H. Li, and H. Han, "The Research and Prospect of Secure Data Access Control in Cloud Storage Environment", in Proc. of Journal of Communications , 10(10), 753-759, Oct., 2015.

[7] J. Li, W. Yao, Y. Zhang, H. Qian, and J. Han, "Flexible and Fine-Grained Attribute-Based Data Storage in Cloud Computing", DOI:10.1109/TSC.2016.2520932, in Proc. of IEEE Transactions on Services Computing, 10(5), 785-796, Sept.-Oct. 1, 2017.

[8] P. W. W. Myint, S. Z. Hlaing, and E. C. Htoon, "An Encryption Access Control Scheme for Flexible Policy Updating in Cloud Storage", in Proc. of the 15th International Conference on Computer Applications(ICCA), Yangon, Myanmar, 28-33, Feb. 16-17, 2017.

[9] L. Q. Mo, and F. Y. Lin, "A dynamic re-encrypted ciphertext-policy attributed-based encryption scheme for cloud storage", in Proc. of IEEE 9th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, Guangdong, China, 14-19, Nov. 8-10, 2014.

[10] C. Vinoth, G. R. A. Raman, "A Survey on Attribute Based Encryption Techniques in Cloud Computing", in Proc. of International Journal of Engineering Sciences & Research Technology, 4(1), 494-497, Jan., 2015.

[11] G. Wungpornpaiboon, and S. Vasupongayya, "Two-layer Ciphertext-Policy Attribute-Based Proxy Re-encryption for Supporting PHR Delegation", DOI: 10.1109/ICSEC.2015.7401447, in Proc. of International Computer Science and Engineering Conference (ICSEC), 23-26, Chiang Mai, Thailand, Nov., 2015.

[12] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing", in Proc. of IEEE Transactions on Parallel and Distributed Systems, 24(1), 131-143, Jan., 2013.