A Study of Attribute-based Proxy Re-encryption Scheme in Cloud Environments

Pei-Shan Chung[1], Chi-Wei Liu[2], and Min-Shiang Hwang[2]
(Corresponding author: Min-Shiang Hwang)

Department of Management Information Systems, National Chung Hsing University[1]
250 Kuo Kuang Road, Taichung 402, Taiwan, R.O.C.
Department of Computer Science and Information Engineering, Asia Universtiy[2]
500 Liufeng Road, Wufeng, Taichung 402, Taiwan, R.O.C.
(Email: mshwang@asia.edu.tw)
(Invited Paper)

Abstract

Attribute-based proxy re-encryption (ABPRE) scheme is one of the proxy cryptography, which can delegate the re-encryption capability to the proxy and re-encrypt the encrypted data by using the re-encryption key. ABPRE extending the traditional proxy cryptography and attributes plays an important role. In ABPRE, users are identified by attributes, and the access policy is designed to control the user's access. Using ABPRE can have these advantages: (i) The proxy can be delegated to execute the re-encryption operation, which reduces the computation overhead of the data owner; (ii) The authorized user just uses his own secret key to decrypt the encrypted data, and he doesn't need to store an additional decryption key for deciphering; (iii) The sensitive information cannot be revealed to the proxy in re-encryption, and the proxy only complies to the data owner's command. In this paper, we survey two various access policy attribute-based proxy re-encryption schemes and analyze these schemes. Thereafter, we list the comparisons of them by some criteria.

Keywords: Attribute-based proxy re-encryption, cloud Computing, data sharing, revocation

1 Introduction

Cloud computing is an emerging economic and computing paradigm with the development of Internet technology. Various application services can be provided to satisfy users' requirements by the cloud computing [1]. One of cloud computing application services commonly used is data storage [49], and users can outsource their storage and store their sensitive data in the cloud. Due to the property of pay-as-you-go [9], users just need to pay money for the storage space used. Users can spend less money to enjoy the benefit which the cloud provides. However, one security issue related to data confidentiality [17, 25, 28, 29, 48, 55, 59] exists in the cloud. To solve this problem, the data is encrypted by the data owner before uploading to the cloud [26, 47]. Besides, for sharing data with the other users, the data is encrypted according to the corresponding user's key so to achieve access control by the user [7, 27, 42, 52].

Nevertheless, if these operations are performed by the data owner, the computation overhead would be heavy for the data owner. Traditional proxy encryption can solve this problem [19, 20, 24, 37, 50, 51, 54]. If the data would be transmitted to the other user Bob, the data owner Alice firstly designs a proxy and generates a re-encryption key by using the part of user's secret key and an encrypted data, and then sends them to the proxy [21, 22, 23, 30, 32, 53, 56]. The proxy executes the re-encryption operation by using this re-encryption key and produces the re-encrypted data that the user Bob can decrypt with his own secret key. After Bob receives the encrypted data, he can use his secret key to decipher it. However, some problems can be found in this manner: (i) To grant the re-encryption right, an unrealistic level of the trust in the proxy is required [2]; (ii) There is a bidirectional property that let the user Bob can delegate to the data owner Alice; (iii) The sensitive information can be revealed in re-encryption.

The first delegating decryption right methodology was introduced by Mambo and Okamoto in 1997 [39], and this is a new type of the public key cryptography. Later, Blaze et al. proposed an atomic proxy cryptography [4] that can convert a user's key with the encrypted data into the other user's key by the semi-trusted party, but this scheme contains the above problems. For improving this disadvantages, Ateniese et al. proposed an improved proxy re-encryption scheme in 2005 [2], and it is the first unidirectional and single-use scheme. Meanwhile, Sahai and Waters presented the first notion of attribute-based encryption (ABE) scheme in 2005 [43], and it leveraged user's identity as attributes and encrypted data with

these attributes. In addition, Goyal et al. introduced a key-policy attribute based encryption scheme in 2006 [12], where the access structure in the user's secret key is added and the encrypted data is associated with a set of attributes. Then Bethencourt et al. also proposed the other policy ABE [3], ciphertext-policy attribute-based encryption scheme, where the access structure in the encrypted data is added and the user's secret key is associated with a set of attributes. In these schemes, users can decipher the encrypted data, if a set of attributes would satisfy the access structure. In 2007, Green and Ateniese presented an identity-based proxy re-encryption scheme [13], but it is secure in the random oracle model. After that, a serial of researches [5, 8, 16, 34, 40] were proposed based on the other security model.

After the attribute-based encryption scheme has emerged, the first ABPRE based on key policy was proposed by Guo et al. [15], but their scheme exists in bidirectional property. Next, Liang et al. introduced the first ABPRE based on ciphertext policy [33], and this scheme combined the traditional proxy re-encryption with attribute-based encryption scheme, the multi-use property was in this scheme, but the size of the encrypted data increases linearly at multi times. In 2010. Luo et al. presented a ABPRE based on ciphertext policy [38], this scheme can let the data owner decide whether re-encrypting the ciphertext or not, and the positive attributes have multi-value. Besides, in the same year, Yu et al. proposed two different policy types of ABPRE [57, 58], the scheme [57] can simultaneously satisfy the properties of fine-grained, scalability and data confidentiality, and the scheme [58] provided an efficient revoking user's attributes scheme which is based on the ciphertext policy. Moreover, in 2011, more ABPRE schemes were proposed [10, 14, 18, 41, 45], one of these schemes was proposed by Do et al. [10] to improve the collusion attack of [57]. After that, Liu et al. presented an ABPRE scheme [35, 36] that adds the timestamp to their scheme, and it can prevent the revoked user to decipher the encrypted data. Apart from this, in 2012, Seo et al. pointed that the computation cost of these ABPRE schemes are associated with the number of attributes, so they proposed an ABPRE with a constant number of paring operations.

According to these types of policy, ABPRE schemes are roughly categorized as either key policy or ciphertext policy. In this paper, the survey starts from an atomic proxy cryptography that can delegate the semi-trusted party to re-encrypt the encrypted data, followed by the first notion of the attribute-based proxy re-encryption based on ciphertext policy. Two key-policy attribute-based proxy re-encryption schemes and two ciphertext-policy ABPRE schemes are introduced. Thereafter, an attribute-based proxy re-encryption with a constant number of paring operations is described at the end.

1.1 The Criteria of an Ideal Attribute-based Proxy Re-encryption Scheme

According to these schemes, a summary of the criteria with which an ideal attribute-based proxy re-encryption schemes should be satisfied is given, and these are listed as follows.

C1. ***Unidirectionality*** **[2, 13]**
The cloud is granted to re-encrypt the data, but the cloud only lets the encrypted data CT be translated into CT'; it can't change back from CT' to CT. Therefore, the user's encrypted data doesn't permit the data owner to decrypt.

C2. ***Data confidentiality***
The data is encrypted by data owner before uploading to the cloud, and the encrypted data can be deciphered by the authorized user. The unauthorized party including the cloud can't obtain the information about the encrypted data.

C3. ***Non-interactive*** **[2, 13]**
The data owner generates the re-encryption key by himself. He doesn't need the untrusted third party including the cloud to participate; the data owner doesn't interact with the untrusted party.

C4. ***Non-transitive*** **[2]**
The cloud can't be granted with the re-delegate decryption right. If the cloud has two re-encryption key, $rk_{A\to B}$ and $rk_{B\to C}$, it can't generate the re-encryption key $rk_{A\to C}$ from $rk_{A\to B}$ to $rk_{B\to C}$.

C5. ***Multi-use***
The encrypted data can be encrypted multiple times by the cloud. The encrypted data is re-encrypted from the user Alice, and then passed to Bob, and it also can be re-encrypted from Bob, and then passed to Clare.

C6. ***Re-encryption control*** **[38]**
The data owner can determine whether the encrypted data can be re-encrypted. If the data owner doesn't want to do this, the cloud can't re-encrypt the encrypted data.

C7. ***Master key security*** **[2]**
The data owner can delegate the cloud to transfer the encrypted data with the other key, and the data user can decrypted the re-encrypted data by using his secret key. However, the data user cannot collude with the cloud to obtain the data owner's master key.

C8. ***Collusion resistant***
The revoked user cannot collude with the cloud to obtain the encrypted data which does not belong to him.

1.2 Organization

In this paper, we survey several attribute-based proxy re-encryption schemes including two varied access policies: key policy and ciphertext policy. The rest of this paper is organized as follows. In next Section , we introduce a proxy cryptography and these attribute-based proxy re-encryption schemes. In Section 3, we compare these attribute-based proxy re-encryption schemes by using the criteria which is illustrated in Section 1, and in Section 4, our conclusions are described.

2 Related Works

In this section, we review a proxy re-encryption scheme, an atomic proxy cryptography, two various access policy attribute-based proxy re-encryption schemes. First, we list the notation used in this paper which are defined in Table 1. In ABPRE, if the data owner wants to share the data with the data user, he can generate a re-key and a ciphertext associated with attributes, and then forward to the proxy (note that the proxy is the cloud service provider in cloud environment). The proxy can comply the data owner's command to transfer the key with the ciphertext into another key with which the data user can decrypt. Besides, the data user's secret key is associated with attributes. When the data user wants to decrypt the ciphertext, he just use his own secret key to do it, and he can obtain the plaintext if the attributes in secret key satisfies the attributes in the ciphertext. In other words, if ABPRE is key policy based, a user will obtain the plaintext when a ciphertext with a set of attributes satisfy the access policy in his secret key. If ABPRE is ciphertext policy based, a user will obtain the plaintext when his secret key with a set of attributes satisfy the access policy in the ciphertext. Generally, ABPRE contains five algorithms: KeyGen(), Encrypt(), ReKeyGen(), ReEncypt(), and Decrypt().

2.1 Blaze et al.'s Scheme

In 1998, Blaze, Bleumer, and Strauss proposed a proxy encryption scheme, atomic proxy cryptography which is based on ElGamma scheme [4]. This scheme can change the key with the ciphertext into another key without disclosing the secret decryption keys and the information of the underlying plaintext. And it can be executed in a semi-trusted party (Only receiving the encryption request from user, this party can execute the encryption algorithm; otherwise this party can't execute this algorithm) because the information of the secret decryption keys and plaintext cannot be known by the semi-trusted party. If the data owner wants to share the data with the data user, he would encrypt the data and generate a proxy key, and then send to the semi-trusted party. The semi-trusted party can change the encrypted data with the data owner's public key into the data user's public key. Therefore, the data user can decipher the encrypted data by his private key. Besides, this is a bidirectional scheme that the data owner and the data user must trust mutually. This scheme would work as follows:

1) Let p and q be two prime numbers such that $p = 2q+1$, and let g be a generator in Z_p^*. Then, p and g denote global parameters. The data owner chooses a random number a from Z_{2q}^* as his secret key, where $0 < a < p-1$, and also computes the inverse of his secret key, $a^{-1}\ mod\ 2q$. The data owner computes $g^a\ mod\ p$ as a public key and publishes this key.

2) The data owner chooses a random number k from Z_{2q}^* as a secret parameter. When the data owner encrypts the message m with his key, he computes $c_1 = mg^k\ mod\ p$, $c_2 = (g^a)^k\ mod\ p$, and computes $a^{-1}b$ as the proxy key $rk_{A\to B}$, where b is the data user's secret key, and then transfer two ciphertext values (c_1, c_2) and $rk_{A\to B}$ to the semi-trusted party.

3) The semi-trusted party computes $c_2^{rk_{A\to B}}$ by using the proxy key $rk_{A\to B}$, and transfers $(c_1, c_2^{rk_{A\to B}})$ to the data user.

4) When the data user receives the message $(c_1, c_2^{rk_{A\to B}})$, he decrypts $c_2^{rk_{A\to B}}$ with his secret key b, and then decrypts c_1 with k. Therefore, the data can be recovered.

2.2 Liang et al.'s Scheme

In 2009, Liang et al. introduced the attribute based proxy re-encryption (ABPRE) scheme which is based on the ciphertext policy [33]. This scheme combines the traditional proxy encryption scheme with attribute-based encryption scheme, and it uses attributes to control the user's access. In this scheme, if the data user's attribute does not appear in the access structure but in the system attributes, the encrypted data can be re-encrypted via Re-encrypt algorithm; and let the data user decrypts the encrypted data with his secret key. This algorithm inputs another re-encryption key and g^d to generate the re-encrypted data which the data user can decipher. However, as the re-encryption time increases, the size of the encrypted data increases linearly. Besides, the design of secret key is similar as [6], and it doesn't need too much exponential computation in re-encryption operation. Next, these algorithms of this scheme work as follows.

- Setup(1^k): Let A_{system} be a set of attribute in system, where $A_{system} = \{1, 2, ..., n\}$. Beside, a index i is given for each attribute $a_i \in A_{system}$, where $1 \le i \le n$. Let G be a bilinear group of prime order p, and $e : G \times G \to G_T$ denote a bilinear map. Next, y, t_i are chosen as two random numbers from Z_p, where $1 \le i \le 3n$, and g, h are selected randomly as two generators of G. In addition, for each $1 \le i \le 3n$, let $Y = e(g,h)^y$ and $T_i = g^{t_i}$, $T_i' = h^{\frac{1}{t_i}}$. The public

Table 1: Notation table

Notations	Signification
G, G_T	The bilinear group of prime order p
A_{system}	A set of attributes in the system
A_U	Attributes of data user U in secret key
A_{CT}	Attributes with the encrypted data CT
A_{U-KP}	The access structure in user's secret key
A_{CT-CP}	The access structure in the encrypted data
SK	User's secret key
M	The message
CT	The encrypted data or the ciphertext

parameter is $PK = (e, g, h, Y, \{T_i, T_i'\}_{1\le i\le 3n})$. The master key is $MK = \langle y, \{t_i\}_{1\le i\le 3n}\rangle$.

- KeyGen(A_U, MK): Let A_U be a set of attributes of data user U. Next, let $r_1, ..., r_n$ be random numbers which are chosen from Z_p, such that $r = \sum_{i=1}^{n} r_i$. It computes $\hat{D} = h^{y-r}$, and for each $i \in A_{system}$, $D_{i,2} = h^{\frac{r_i}{t_{2n+i}}}$ and $D_{i,1} = h^{\frac{r_i}{t_i}}$ if $i \in A_U$, or $D_{i,1} = h^{\frac{r_i}{t_{n+i}}}$ otherwise. Output the user's secret key $SK = \langle A_U, (D_{i,1}, D_{i,2})_{i\in A_{system}}, \hat{D}\rangle$.

- Encrypt(M, A_{CT-CP}): Let A_{CT-CP} be an access structure, and choose a random number $s \in Z_p$ to encrypt a message $M \in G_T$ by computing $\tilde{C} = M \cdot Y^s$. Compute $\hat{C} = g^s$ and $\breve{C} = h^s$. Beside, let $+d_i$ be a positive attribute, and $-d_i$ be a negative attribute. For $i \in A_{system}$, possible conditions are described as follows.

  1) If $+d_i \in A_{CT-CP}$, compute $C_i = T_i^s$.
  2) If $-d_i \in A_{CT-CP}$, compute $C_i = T_{n+i}^s$.
  3) Otherwise, compute $C_i = T_{2n+i}^s$.

  Output the encrypted data $CT = \langle A_{CT-CP}, \tilde{C}, \hat{C}, \breve{C}, (C_i)_{i\in A_{system}}\rangle$.

- ReKeyGen(SK, A_{CT-CP}): Let SK be a valid user secret key, and A_{CT-CP} be an access structure. Choose a random number d from Z_p ,and set $\mathcal{D} = g^d$ and $\hat{D}' = \hat{D}$. For $i \in A_{system}$, possible conditions are described as follows.

  1) If $i \in A_U$, compute $D_{i,1}' = D_{i,1} \cdot (T_i')^d$ and $D_{i,2}' = D_{i,2} \cdot (T_{2n+i}')^d$;
  2) Otherwise, compute $D_{i,1}' = D_{i,1} \cdot (T_{n+i}')^d$ and $D_{i,2}' = D_{i,2} \cdot (T_{2n+i}')^d$.

  And $\mathcal{D}$ is encrypted with the access structure A_{CT-CP} that is the ciphertext of $\mathcal{D}$, $\mathbb{C}$. Output a re-key $rk = \langle A_U, A_{CT-CP}, (D_{i,1}', D_{i,2}')_{i\in A_{system}}, \hat{D}', \mathbb{C}\rangle$.

- ReEncrypt(rk, CT): Let rk be a valid re-key, and CT be a well-formed encrypted data. Check whether A_U satisfies A_{CP-CT} or not, if not, output "$\perp$"; otherwise, for $i \in A_{system}$, possible conditions are described as follows.

  1) If $+d_i \in A_{CP-CT}$, compute $E_i = e(C_i, D_{i,1}') = e(g^{t_i s}, h^{\frac{r_i+d}{t_i}}) = e(g,h)^{s(r_i+d)}$;
  2) If $-d_i \in A_{CP-CT}$, compute $E_i = e(C_i, D_{i,1}') = e(g^{t_{n+i} s}, h^{\frac{r_i+d}{t_{n+i}}}) = e(g,h)^{s(r_i+d)}$;
  3) Otherwise, $E_i = e(C_i, D_{i,2}') = e(g^{t_{2n+i} s}, h^{\frac{r_i+d}{t_{2n+i}}}) = e(g,h)^{s(r_i+d)}$.

  Then, compute $\bar{C} = e(\hat{C}, \hat{D}') \prod_{i\in A_{system}} E_i = e(g^s, h^{y-\sum_{i=1}^{n} r_i}) \cdot e(g,h)^{nds+s\sum_{i=1}^{n} r_i} = e(g,h)^{ys+nds}$. The re-encrypted data $CT' = \langle A_{CT-CP}', \tilde{C}, \bar{C}, \breve{C}, \mathbb{C}\rangle$ is outputted.

- Decrypt(CT, SK): Let SK be a valid user's secret key. And, check whether A_U satisfies A_{CT-CP} or not, if not, output "$\perp$"; otherwise, do

  1) If CT is an original well-formed encrypted data, for $i \in A_{system}$, possible conditions are described as follows.
     a. If $+d_i \in A_{CP-CT}$, compute $E_i = e(C_i, D_{i,1}) = e(T_i^s, h^{\frac{r_i}{t_i}}) = e(g,h)^{sr_i}$;
     b. If $-d_i \in A_{CP-CT}$, compute $E_i = e(C_i, D_{i,1}) = e(T_{n+i}^s, h^{\frac{r_i}{t_{n+i}}}) = e(g,h)^{sr_i}$;
     c. Otherwise, compute $E_i = e(C_i, D_{i,2}) = e(T_{2n+i}^s, h^{\frac{r_i}{t_{2n+i}}}) = e(g,h)^{sr_i}$.

     Compute

     $$\frac{\tilde{C}}{e(\hat{C}, \hat{D}) \cdot \prod_{i\in A_{system}} E_i} = \frac{M \cdot e(g,h)^{ys}}{e(g^s, h^{y-r}) \cdot e(g,h)^{sr}}$$

     and the message M can be obtained.

  2) Else if CT' is a re-encrypted well-formed data, $\mathbb{C}$ is decrypted by using SK to obtain $\mathcal{D} = g^d$. Then compute $\frac{\tilde{C} e(\mathcal{D}, \breve{C})^n}{\bar{C}} = \frac{M \cdot e(g,h)^{ys} \cdot e(g^d, h^s)^n}{e(g,h)^{ys+nds}}$ to obtain the message M.

2.3 Luo et al.'s Scheme

In 2010, Luo et al. proposed a cipertext policy attribute-based proxy re-encryption scheme [38]. This scheme not only maintains the properties of Liang et al.'s scheme [33], but adds multi-valued on the positive attributes. This paper also introduced the property of re-encryption control. This property allows the data owner to determine whether the encrypted data can be re-encrypted or not. Besides, the access policy in this scheme is based on the ciphertext policy, the access policy is in the encrypted data to control a user's access, and the access policy is AND-gate policy. Apart from this, the policy of this scheme supports multi-value attributes, negated attributes (note that means these attributes the user doesn't have) and wildcards (note that it means the attributes don't appear in the AND-gate, so these attributes are not considered in decryption algorithm). The main construction of this paper is described as follows.

- Setup(1^k): Let $\mathcal{U} = \{att_1, att_2, \cdots, att_n\}$ be a set of attributes. For $\mathcal{U}$, let $\bar{\mathcal{U}} = \{\neg att_1, \neg att_2, \cdots, \neg att_n\}$ be a set of negated attributes and for each attribute att_i, let $S_i = \{v_{i,1}, \cdots, v_{i,n_i}\}$ be a set of possible values, where $\mid S_i \mid = n_i$. Let $W = [W_1, ..., W_n]$ be an access policy, where $W_i \in S_i \cup \{\neg att_i, *\}$. Let G denote a bilinear group of prime order p, and $e : G \times G \rightarrow G_T$ denote a bilinear map. Choose random numbers $y, t_{i,j}, a_i, b_i$ from Z_p, where $1 \leq i \leq n$ and $1 \leq j \leq n_i$, and generate generators $g, g_2, g_3 \in G$ at random. Compute $g_1 = g^y$, $Y = e(g_1, g_2)$, $\{\{T_{i,j} = g^{t_{i,j}}\}_{1 \leq j \leq n_i}, A_i = g^{a_i}, B_i = g^{b_i}\}_{1 \leq i \leq n}$. The public key is $PK = (e, g, g_1, g_2, g_3, Y, \{\{T_{i,j} = g^{t_{i,j}}\}_{j \in [1,n_i]}, A_i, B_i\}_{i \in [1,n]})$. The master key is $MK = (y, \{\{t_{i,j}\}_{j \in [i,n_i]}, a_i, b_i\}_{i \in [1,n]})$.

- KeyGen(MK, L): For a data user who can obtain the corresponding secret key, let $L = [L_1, ..., L_n]$ be an attribute list. Choose random numbers r_i, r_i', r_i'' from Z_p, where $1 \leq i \leq n$, and set $r = \sum_{i=1}^{n} r_i$. Compute $D_0 = g_2^{y-r}$. For $1 \leq i \leq n$, possible conditions are described as follows.

  1) If $L_i = v_{i,k_i}$, compute $D_i = (g_2^{r_i} T_{i,k_i}^{r_i'}, g^{r_i'}), F_i = (g_2^{r_i} B_i^{r_i''}, g^{r_i''})$;
  2) If $L_i = \neg att_i$, compute $D_i = (g_2^{r_i} A_i^{r_i'}, g^{r_i'}), F_i = (g_2^{r_i} B_i^{r_i''}, g^{r_i''})$.

  Output the secret key $SK_L = (L, D_0, \{D_i, F_i\}_{i \in [1,n]})$.

- Encrypt(PK, M, W): Choose a random number s form Z_p to encrypt the message $M \in G_T$. Compute $\tilde{C} = M \cdot Y^s$, $C_0 = g^s$, and $C_0' = g_3^s$. For $1 \leq i \leq n$, possible conditions are described as follows.

  1) If $W_i = v_{i,k_i}$, compute $C_i = T_{i,k_i}^s$;
  2) If $W_i = \neg att_i$, compute $C_i = A_i^s$;
  3) If $W_i = *$, compute $C_i = B_i^s$.

  Output the encrypted data $CT = (W, \tilde{C}, C_0, C_0', \{C_i\}_{i \in [1,n]})$.

- ReKeyGen(SK_L, W): Let SK_L be a valid secret key, W be an access policy. Choose a random number d from Z_p, and set $\mathcal{D} = g^d$ and $D_0' = D_0$. Compute $D_i' = (D_{i,1} g_3^d, D_{i,2}), F_i' = (F_{i,1} g_3^d, F_{i,2})$. Encrypt $\mathcal{D} = g^d$ with the access policy W and get the ciphertext of $\mathcal{D}$, $\mathbb{C}$. Then, Output the re-encryption key $rk = (L, W, D_0', \{D_i', F_i'\}_{i \in [1,n]}, \mathbb{C})$.

- ReEncrypt(rk, CT): Let rk be a valid re-encryption key, and CT be a well-formed encrypted data. Check whether L satisfy W or not, if not, output $\perp$. If it satisfies, for $1 \leq i \leq n$, possible conditions are described as follows.

  1) If $W_i' \neq *$, compute $E_i = \frac{e(C_0, D_{i,1}')}{e(C_i, D_{i,2}')}$, where $*$ is wildcard;
  2) If $W_i' = *$, compute $E_i = \frac{e(C_0, F_{i,1}')}{e(C_i, F_{i,2}')}$.

  Next, compute

$$\begin{aligned} \prod_{i=1}^{n} E_i &= e(g, g_2)^{s r_i} e(g, g_3)^{sd} \\ \bar{C} &= e(C_0, D_0') \prod_{i=1}^{n} E_i \\ &= e(g^s, g_2^{y-r}) e(g, g_2)^{sr} e(g, g_3)^{sd}. \end{aligned}$$

  Output the re-encrypted data $CT' = (W', \tilde{C}, C_0', \bar{C}, \mathbb{C})$.

- Decrypt(CT, SK_L): Let SK_L be a valid secret key. Check whether L satisfies W or not. If it satisfies, the data user can use his secret key to decrypt CT by executing these computation.

  1) If CT is an original well-formed encrypted data, for $1 \leq i \leq n$, possible conditions are described as follows.
     a. If $W_i \neq *$, compute $D_i' = D_{i,1}$ and $D_i'' = D_{i,2}$;
     b. If $W_i = *$, compute $D_i' = F_{i,1}$ and $D_i'' = F_{1,2}$.
     Compute $\frac{\tilde{C} \prod_{i=1}^{n} e(C_i, D_i'')}{e(C_0, D_o) \prod_{i=1}^{n} e(C_0, D_i')}$, and obtain the message M.
  2) Else if CT is a re-encrypted well-formed encrypted data, $mathbbC$ can be decrypted by using the data user's secret key and get $\mathcal{D} = g^d$. Then compute $\frac{\tilde{C} \cdot e(C_0', g^d)^n}{\bar{C}}$ and obtain the message M.

2.4 Yu et al.'s Scheme

In 2010, Yu et al.'s proposed an attribute-based proxy re-encryption scheme [57] whose access policy is based

on KP-ABE [12]. They point out that the data confidentiality is not only the basic security requirement, but also a juristic concern when users out source their data in the cloud or share their data to other users via the cloud. Therefore, this proposed scheme can achieve fine-grained, scalability and data confidentiality at the same time, and it doesn't need more complexity key management and higher encryption computation cost than previous works. Besides, its encryption complexity is related to the number of attributes and the number of the encrypted data, not the number of users in the system. In data file creation/deletion and new grant operations, the current data file can be affected, but the system-wide data updating and re-key aren't. These operations of Yu et al.'s scheme work as follows.

- System Setup: Let $\mathcal{U} = \{1, 2, ..., n\}$ denote the universe of attributes. And, let k be a security parameter selected by the data owner and the algorithm level interface $ASetup(k)$ is called by the data owner. Next, y, t_i are chosen as two random numbers from Z_p, where each attribute $i \in \mathcal{U}$. It outputs the system public parameter $PK = (T_1 = g^{t_1}, ..., T_n = g^{t_n}, Y = e(g,g)^y)$, and the system master key $MK = (t_1, ..., t_n, y)$. Besides, each component of PK can be signed by the data owner, PK and these signatures can be sent to the cloud server.

- New File Creation: This operation is executed before a file is uploaded to the cloud server. The data file is processed by the data owner.

  1) Choose a unique ID for this data file;
  2) Choose a symmetric data encryption key DEK at random from the key space, and encrypt data with DEK, i.e. $\{DataFile\}_{DEK}$;
  3) Define a set of attributes A_{CT} for the data file and encrypt DEK with A_{CT} using KP-ABE, i.e.

     $$(\tilde{E}, \{E_i\}_{i \in A_{CT}}) \leftarrow AEncrypt(A_{CT}, DEK, PK),$$

     where $\tilde{E} = DEK \times Y^s$ and $\{E_i = T_i^s\}_{i \in A_{CT}}$;
  4) Divide the data file into ID, header, and body. Header stores the encrypted DEK, and body stores the encrypted data, $DataFile_{DEK}$. These are the format of the data file and it is stored on the cloud server.

- New User Grant: The data owner assigns the new user a unique identity w and access structure A_{U-KP} when he wants to join the system. And this operation is executed as follows.

  1) For each new user w, the component of user's secret key $sk_j = g^{\frac{q_j(0)}{t_j}}$ is generated and a secret key $SK = \{sk_j\}_{j \in L_{A_{U-KP}}}$ is created, where $L_{A_{U-KP}}$ is the set of attributes connected to leaf nodes of A_{U-KP};
  2) Compute $C = (A_{U-KP}, SK, PK, \delta_{o,(A_{U-KP},SK,PK)})_{PK_w}$, where PK_w is user w's public key and $\delta_{o,(A_{U-KP},SK,PK)}$ is the data owner's signature on message (A_{U-KP}, SK, PK);
  3) Send $(T, C, \delta_{o,(T,C)})$ to cloud server, where $T = (w, \{j, sk_j\}_{j \in A_{U-KP} \setminus Att_D})$ and Att_D is the dummy attribute.

  When the cloud server receives this tuple $(T, C, \delta_{o,(T,C)})$, the cloud executed this operation as follows.

  1) Check whether $\delta_{o,(T,C)}$ is correct or not;
  2) If correct, T is stored in the system user list UL;
  3) Send C to the data user.

  When the data user receives C, he processes as follows.

  1) Check whether $(\delta_{o,(A_{U-KP},SK,PK)})_{PK_w}$ is correct or not;
  2) If correct, accept (A_{U-KP}, SK, PK) as his access structure, secret key, and the system public key.

- User Revocation: When the data user v is revoked from the system, the data owner executes this operation as follows.

  1) Determine a minimal set of attributes D, and there is no leaving user's access structure.
  2) Choose the new components of the system master key t_i' at random from Z_p, where $i \in D$;
  3) Compute $T_i' \leftarrow g^{t_i'}$ and $rk_{i \hookrightarrow i'} \leftarrow \frac{t_i'}{t_i}$ to update the components of the system public key;
  4) Send $Att = (v, D, \{i, T_i', \delta_{o,(i,T_i')}, rk_{i \hookrightarrow i'}\}_{i \in D})$ to the cloud server, where $\delta_{o,(i,T_i')}$ is the data owner's signature on message (i, T_i').

  When the cloud server receives Att, it processes as follows.

  1) Remove leaving user's identity v from the system user list UL;
  2) Store the tuple $\{i, T_i', \delta_{o,(i,T_i')}\}_{i \in D}$ and join $rk_{i \hookrightarrow i'}$ to attribute i history list AUL_i for recording the PRE key of the latest version in AHL.

- File Access: When the data user wants to access file, this operation can be executed as follows. In this operation, the data owner doesn't need to join, only the cloud server responds the requirement of the data file access of the data user. If necessary, the cloud server can update the data user's secret keys and re-encrypt the data file which the data user needs.

1) The data user sends a access request to the cloud server;
2) The cloud checks whether $u \in UL$ or not, if $u \in UL$, gets $(u, \{j, sk_j\}_{j \in L_{A_{U-KP}} \setminus Att_D})$;
3) Check whether sk_j is the latest version or not, if not, update sk_j with sk'_j by $(rk_{i \hookrightarrow i^{(n)}})^{-1}$, where $i^{(n)}$ is the latest version of attribute i;
4) Check whether the requested data file is the latest version or not, if not, update $rk_{i \hookrightarrow i^{(n)}} = \frac{t_i^{(n)}}{t_i}$ and output $E_i^{(n)}$ by computing $E_i^{(n)} \leftarrow (E_i)^{rk_{i \hookrightarrow i^{(n)}}} = g^{t_{i^{(n)}} s}$;
5) Forward the respond message $(\{j, sk'_j, T'_j, \delta_{o,(j,T'_j)}\}_{j \in L_{A_{U-KP}} \setminus Att_D}, FL)$ to the data user, where FL is the requested data file.

When the data user receives the respond message, he processes as follows.

1) Check whether the data owner's signature $\delta_{o,(j,T'_j)}$ is correct or not, if correct, replace each sk_j with sk'_j, and update T_j with T'_j;
2) Compute

$$\prod_{j \in L_{A_{U-KP}}} e(g,g)^{q_j(0)s} = e(g,g)^{q_r(0)s} = e(g,g)^{ys};$$

3) Decrypt $DEK = \tilde{E} Y^{-s} = DEK \times Y^s \times e(g,g)^{-ys}$ and decipher the requested data file with DEK.

2.5 Yu et al.'s Scheme

In addition, in the same year, Yu et al's also proposed the other attribute based proxy re-encryption scheme [58]. This scheme is different with [57], the access policy of this proposed scheme is based on CP-ABE [3]. The authors point out that if these existing CP-ABE schemes are directly applied in the real environment, some issue will be found out: firstly, the access structure in CP-ABE is monotonic, and it can't express negative word and so on; CP-ABE is not able to realize provable security and the efficient construction; besides, CP-ABE can't simultaneously achieve them; secondly, CP-ABE is not able to provide an efficient manner in the user revocation operation. Because of the above issues, this paper proposes a scheme of combining proxy re-encryption with CP-ABE, and it reduces the burden of the authority revoking any user's attribute. Therefore, this scheme can let the authority freely remove any user's attribute at any time. The construction of this paper is illustrated as follows.

- Setup(1^λ): Let $\mathcal{U} = \{1, 2, ..., n\}$ denote the universe of attributes, let G denote a bilinear group of prime order p, let g denote a generator, and let $e : G \times G \rightarrow G_T$ be a bilinear map. Then random numbers $y, t_1, ..., t_{3n}$ are selected from Z_p. Output the public parameter $PK = (e, g, Y = e(g,g)^y, T_1 = g^{t_1}, ..., T_3 = g^{t_3})$, and the master key $MK = (y, t_1, ..., t_{3n})$. Besides, let $ver = 1$ be initialize version number, (ver, PK) is published, and (ver, MK) is kept by the authority.
- KeyGen(MK, A_U): Let A_U be a set of attributes in data user U private key. Next, let $r_1, ..., r_n$ be random numbers which are chosen from Z_p, such that $r = \sum_{i=1}^{n} r_i$. It computes $D = g^{y-r}$, and for each $i \in \mathcal{U}$, $F_i = g^{\frac{r_i}{t_{2n+i}}}$ and $D_i = g^{\frac{r_i}{t_i}}$ if $i \in A_U$, or $D_i = g^{\frac{r_i}{t_{n+i}}}$ otherwise. Output the user's secret key $SK = (ver, A_U, D, \bar{D} = \{D_i, F_i\}_{i \in \mathcal{U}})$. Note that i is the negative attribute if i does not appear in A_U.
- Encrypt(M, AS, PK): Let AS be a single AND gate and also be the access structure of CT, and $AS = \bigwedge_{\tilde{i} \in I} \tilde{i}$, where $\tilde{i}$ is the literal meaning of attribute i, which may be the positive attribute or the negated attribute, and I is the set of attributes of interest. A random number s is chosen from Z_p to encrypt the message $M \in G_T$ by computing $\tilde{C} = M \cdot Y^s$. Compute $\hat{C} = g^s$. Besides, let $+i$ be a positive attribute, and $-i$ be a negative attribute. For each $i \in I$, possible conditions are described as follows.
  1) If $\tilde{i} = +i$, compute $C_i = T_i^s$.
  2) If $\tilde{i} = -i$, compute $C_i = T_{n+i}^s$.
  3) If $i \in \mathcal{U} \setminus I$, compute $C_i = T_{2n+i}^s$.

  Output the encrypted data $CT = \{ver, AS, \tilde{C}, \hat{C}, \{C_i\}_{i \in \mathcal{U}}\}$, where ver is the current version number.
- ReKeyGen(γ, MK): Denote the range of each $i \in \gamma$ being $[1, 2n]$, also the range of each $i \in \beta$ being $[1, 2n]$. And $1 \leq value \leq n$ represents the positive attribute, and $value \geq n$ means the attribute $i - n$ being negative. For each $i \in \gamma$, t'_i is chosen randomly from Z_p, and $rk_i = \frac{t'_i}{t_i}$ is generated. For each $i \in 1, ..., 2n \setminus \gamma$, $rk_i = 1$. Output the re-key $rk = (ver, \{rk_i\}_{1 \leq i \leq 2n})$. Besides, the system version number is initialized that ver is equal to 1 when everything is done.
- ReEncrypt(CT, rk, β): Denote the range of each $i \in \beta$ being $[1, 2n]$. And the version number of CT and rk is different, this algorithm directly outputs the encrypted data CT. Otherwise, CT can be re-encrypted. For each $i \in \beta$, if i satisfies one of these conditions which are described as follows, it will execute the computation.
  1) $1 \leq i \leq n$, compute $C'_i = C_i^{rk_i}$;
  2) $n \leq i \leq 2n$, compute $C'_{i-n} = (C_{i-n}^{rk_i})$.

  For each $i \in \mathcal{U}$, if $i \notin \beta$ and $i + n \notin \beta$, or $i \notin I$, compute $C'_i = C_i$. Output the re-encrypted data $CT' = (ver + 1, AS, \tilde{C}, \hat{C}, \{C'_i\}_{i \in \mathcal{U}})$, where ver is the version number of the encrypted data.

- ReKey($\bar{D}, rk, \theta$): Denote the range of each $i \in \theta$ being $[1, 2n]$. And the version number of $\bar{D}$ and rk is different, this algorithm directly returns $\bar{D}$. Otherwise, $\bar{D}$ can be updated. For each $i \in \theta$, possible conditions are described as follows.

  1) If $1 \leq i \leq n$, compute $D'_i = D_i^{rk_i^{-1}}$;
  2) If $n \leq i \leq 2n$, compute $D'_{i-n} = (D_{i-n})^{rk_i^{-1}}$.

  For each $i \in \mathcal{U}$, if $i \notin \theta$ and $i + n \notin \theta$, $D'_i = D_i$ is computed. Output $\bar{D}' = \{D'_i, F_i\}_{i \in \mathcal{U}}$. Besides, *ver* of the corresponding SK is added to 1.

- Decrypt(CT, PK, SK): If randomly choose two from CT, PK, SK, and the version number of any two are different, it will output " $\perp$ ". Otherwise, decrypt $T = \{ver, AS, \tilde{C}, \hat{C}, \{C_i\}_{i \in \mathcal{U}}\}$. For each $\tilde{i} \in I$, possible conditions are described as follows.

  1) If $\tilde{i} = +i$ and $i \in A_U$, compute
  $e(C_i, D_i) = e(g^{t_i s}, g^{\frac{r_i}{t_i}}) = e(g, g)^{r_i s}$;
  2) If $\tilde{i} = -i$ and $i \notin A_U$, compute
  $e(C_i, D_i) = e(g^{t_{n+i} s}, g^{\frac{r_i}{t_{n+i}}}) = e(g, g)^{r_i s}$;
  3) If $\tilde{i} \notin I$, compute
  $e(C_i, D_i) = e(g^{t_{2n+i} s}, g^{\frac{r_i}{t_{2n+i}}}) = e(g, g)^{r_i s}$.

  Therefore, CT is decrypted, and M is obtained by computing $\frac{\tilde{C}}{e(\hat{C}, \hat{D}) \prod_{i=1}^{n} e(g,g)^{r_i s}} = M$.

2.6 Do et al.'s Scheme

In 2011, Do et al. proposed an attribute based proxy re-encryptions scheme for protecting data confidentiality [10], and the scenario of this paper is in the healthy cloud environment. Do et al. point out some disadvantages in Yu et al.'s scheme [57]: firstly, the collusion attack would occur in their scheme when the data user colludes with the cloud server; secondly, this scheme can't provide the selective delegation of the level of the trust; for example, the agencies can be chose to know the partial user's information. Thus, to improve these disadvantages, Do et al.'s scheme divides the data file into a header and a body which are same as [57]. But, a header is stored in the privilege manager group which is a trusted authority in the cloud, and a body is stored in the cloud server in this scheme. Besides, the privilege management model uses the concept of Type-based proxy re-encryption to manage the data access. Now, these operations are introduced as follows.

- System Setup: Let $\mathcal{U} = \{1, 2, ..., N\}$ denote the universe of attributes. And y, a_i are chosen as two random numbers from Z_p, where each attribute $i \in \mathcal{U}$. It outputs the system public parameter $PK = (A_1 = g^{a_1}, ..., A_N = g^{a_N}, Y = e(g, g)^y)$. Besides, the type information $T = (t_1, ...t_N)$ is generated.

- New File Creation: The data owner creates a symmetric data encryption key $DEK = g^{H(r\|t)}$, and uses the DEK to encrypt the data file M, $M \cdot g^{H(r\|t)}$. Next, header $(i, \bar{E}_i, \{E_i\}_{i \in A_{CT}})$ and body $(M \cdot g^{H(r\|t)})$ are generated by the data owner, and the header is stored in the privilege manager group, and body is stored in the cloud server.

- New User Grant: The data owner assigns the new user a unique identity w and access structure A_{U-KP} when he wants to join the system.

- User Revocation: When the data user is revoked by the data owner, this operation can be executed. By updating the system master key and public key, the data user can be removed.

- File Access: When the data user wants to access the requested data, he can send a access request to the privilege manager group. Privilege manager executes this operation as follows.

  1) Check whether the data user is in UL or not, if it is, send proxy re-encryption key R and $\{sk_i\}_{i \in A_{U-KP}}$ to the cloud server.
  2) The cloud server receives the message and generates secret key sk'_i for the data user by using $AUpdateSK(i, sk_i, AHL_i)$ algorithm. The secret key is attached to $(M \cdot g^{H(r\|t)})$ and then it sent to the data user.
  3) Privilege manager group uses the algorithm $AUpdateAtt4File(i, E_i, AHL_i)$ to generate $(i, \bar{E}_i, \{E_i\}_{i \in A_{U-KP}})$ which corresponds to sk'_i, and send it.
  4) The data user receives the message, and use the algorithm $ADecryptH(L_{A_{U-KP}}, SK, E)$ to obtain DEK, where SK is user's secret key and E is the encrypted DEK.
  5) Execute the algorithm $ADecryptB(M \cdot DEK, DEK)$, the data M can be obtain by decrypting $(M \cdot g^{H(r\|t)})$ with DEK.

2.7 Seo et al.'s Scheme

In 2012, Seo et al. proposed an attribute-based proxy re-encryption with a constant number of paring operations [44]. Since the computation cost of the previous attribute based proxy re-encryption schemes are according to the number of attributes of the scheme to compute, there are not a constant ciphertext length and number of paring operation in these schemes. Therefore, the concept of a constant ciphertext length based CP-ABE was proposed by Emura et al. [11] in 2009; the length of ciphertext and the computation cost were signally diminished compared to other attribute-based encryption schemes. Seo et al. extended this concept to provide a constant length of message and number of paring operations based attribute-based proxy re-encryption scheme. It is different from

other attribute-based proxy re-encryption schemes because of the lower computation cost. Next, we introduce the main construction of this scheme as follows.

- Setup(1^k): Let A_{system} be a set of attribute in system, where $A_{system} = \{1, 2, ..., n\}$. Besides, a index i is given for each attribute $a_i \in A_{system}$, where $1 \leq i \leq n$. Let G be a bilinear group of prime order p, and $e : G \times G \rightarrow G_T$ denote a bilinear map. Next, four random numbers k, y, z, t_i are chosen from Z_p, where $1 \leq i \leq 3n$, and two generators of G, g and h are selected at random . In addition, for each $1 \leq i \leq 3n$, let $Y = e(g, h)^y$ and $T_i = g^{t_i}$. The public parameter is $PK = \langle e, g^z, h, Y^{k \cdot z}, \{T_i, \frac{t_i}{k \cdot z}\}_{1 \leq i \leq 3n} \langle$. The master key is $MK = \langle k, y, z, \{t_i\}_{1 \leq i \leq 3n} \rangle$.

- KeyGen(A_U, MK): Let A_U be a set of attributes of data user U. Next, let $r_1, ..., r_n$ be random numbers which are chosen from Z_p, such that $r = \sum_{i=1}^{n} r_i$. It computes $\hat{D} = (h^{y-r})^k$, and for each $i \in A_{system}$, $\{D_{i,1} = h^{r_i}\}_{i \in A_{system}}$. Output the user's secret key $SK = \langle A_U, \{D_{i,1}\}_{i \in A_{system}}, \hat{D}, k \cdot z \rangle$.

- Encrypt(M, A_{CT-CP}): Let A_{CT-CP} be an access structure, and choose a random number $s \in Z_p$ to encrypt a message $M \in G_T$ by computing $\tilde{C} = M \cdot Y^{s \cdot k \cdot z}$. Compute $\hat{C} = g^{s \cdot z}$ and $\breve{C} = h^{s \cdot k \cdot z}$. Beside, let $+d_i$ be a positive attribute, and $-d_i$ be a negative attribute. For $i \in A_{system}$, possible conditions are described as follows.

  1) If $+d_i \in A_{CT-CP}$, compute $C_i = T_i^s$.
  2) If $-d_i \in A_{CT-CP}$, compute $C_i = T_{n+i}^s$.
  3) Otherwise, compute $C_i = T_{2n+i}^s$.

  Output the encrypted data $CT = \langle A_{CT-CP}, \tilde{C}, \hat{C}, \breve{C}, (C_i)_{i \in A_{system}} \rangle$.

- ReKeyGen(SK, A_{CT-CP}): Let SK be a valid user secret key, and A_{CT-CP} be an access structure. Choose a random number d from Z_p ,and set $\mathcal{D} = g^d$ and $\hat{D}' = \hat{D}$. For $i \in A_{system}$, $D'_{i,1} = D_{i,1} \cdot h^d$. $\mathcal{D}$ is encrypted with the access structure A'_{CT-CP} that is the ciphertext of $\mathcal{D}$, $\mathbb{C}$. Output a re-key $rk = \langle A_U, A'_{CT-CP}, \{D'_{i,1}\}_{i \in A_{system}}, \hat{D}', k \cdot z, \mathbb{C} \rangle$.

- ReEncrypt(rk, CT): Let rk be a valid re-key, and CT be a well-formed encrypted data. Check whether A_U satisfies A_{CP-CT} or not, if not, output "$\perp$"; otherwise, for $i \in A_{system}$, possible conditions are described as follows.

  1) If $+d_i \in A_{CP-CT}$, compute $T_i = \frac{t_i}{k \cdot z}$;
  2) If $-d_i \in A_{CP-CT}$, compute $T_i = \frac{t_{n+i}}{k \cdot z}$;
  3) Otherwise, $T_i = \frac{t_{2n+i}}{k \cdot z}$.

  Next, compute $C = \prod_{i \in A_{system}} C_i = g^{s \cdot \sum_{i \in A_U} t_i} = g^{s \cdot t}$, $D = \prod_{i \in A_{system}} D_i = h^{d + \sum_{i \in A_U} r_i} = h^{n \cdot d + r}$, and $E = e(C, D^T) = e(g, h)^{(n \cdot d + r)(k \cdot s \cdot z)}$. Then, compute $\bar{C} = e(\hat{D}, \hat{C}') \cdot E = e(g^{s \cdot z}, h^{k \cdot (y-r)}) \cdot e(g, h)^{(n \cdot d + r)(k \cdot s \cdot z)} = e(g, h)^{(k \cdot s \cdot z \cdot y) + (n \cdot d \cdot k \cdot s \cdot z)}$. Output the re-encrypted data $CT_{re} = \langle A'_{CT-CP}, \tilde{C}, \bar{C}, \breve{C}, \mathbb{C} \rangle$.

- Decrypt(CT, SK): Let SK be a valid user's secret key. And, check whether A_U satisfies A_{CT-CP} or not, if not, output "$\perp$"; otherwise, do

  1) If CT is an original well-formed encrypted data, for $i \in A_{system}$, possible conditions are described as follows.
     a. If $+d_i \in A_{CP-CT}$, compute $T_i = \frac{t_i}{k \cdot z}$;
     b. If $-d_i \in A_{CP-CT}$, compute $T_i = \frac{t_{n+i}}{k \cdot z}$;
     c. Otherwise, compute $T_i = \frac{t_{2n+i}}{k \cdot z}$.

     Next, compute

$$\begin{aligned} C &= \prod_{i \in A_{system}} C_i \\ &= g^{s \cdot \sum_{i \in A_U} t_i} \\ &= g^{s \cdot t} \end{aligned}$$

     and

$$\begin{aligned} D &= \prod_{i \in A_{system}} D_i \\ &= h^{\sum_{i \in A_{system}} r_i} \\ &= h^r. \end{aligned}$$

     Then, compute $E = e(C, D^T) = e(g, h)^{k \cdot r \cdot s \cdot z}$. Decrypt $\frac{\tilde{C}}{e(\hat{C}, \hat{D}) \cdot e(g,h)^{k \cdot r \cdot s \cdot z} = M}$ and obtain the message M.
  2) Otherwise, CT' is a re-encrypted well-formed data, and $\mathbb{C}$ is decrypted by using SK to obtain $\mathcal{D} = g^d$. Then, decrypt $\frac{\tilde{C} e(\mathcal{D}, \breve{C})^n}{\bar{C}} = \frac{M \cdot e(g,h)^{k \cdot s \cdot y \cdot z} \cdot e(g,h)^{n \cdot d \cdot s \cdot k \cdot z}}{e(g,h)^{(k \cdot s \cdot z \cdot y) + (n \cdot d \cdot k \cdot s \cdot z)}} = M$ to obtain the message M.

3 Comparisons

In this section, we list the comparison of these schemes which we survey. Firstly, we use the criteria to compare these attribute-based proxy re-encryption schemes, and these criteria are introduced in Section 1. Secondly, we compare the properties and performance of these schemes. And the performance comparisons are classified according to the policy.

3.1 Security Requirement Analysis

These schemes which we survey were compared by the criteria that we listed in Section 1. The criteria contain C1- unidirectionality, C2- data confidentiality, C3- non-interactive, C4- non-transitive, C5- multi-use, C6-

re-encryption control, C7- master key security, and C8-collusion resistant. This comparison table is listed in Table 2.

On one hand, almost schemes can achieve these basic security requirement including unidirectionality, data confidentiality, non-interactive, non-transitive, master key security and collusion resistant, except Yu et al.'s scheme [57], because their scheme cannot withstand the collusion attack by using the cloud and the data user. On the other hand, the access policy of Yu et al.'s scheme [57] and Do et al.'s scheme [10] are based on the key policy and focused on updating the version of the user's secret key, hence their schemes are hard to satisfy the property of multi-use and re-encrypt control.

3.2 Performance Analysis

C_e denotes a pairing operation, E denotes an exponentiation group operation, M denotes a multiplication group operation, S denotes a signature operation, n denotes the number of attributes, and $|*|$ denotes the number of the element $*$. We neglect some computation operations which just need less computation cost, ex. symmetric encryption operation. Tables 3, 4, 5 and 6 list the comparison results. And we compare these attribute-based proxy re-encryption schemes, but not including [4]. In addition, we divide the performance comparison into two parts. Because the main construction can be categorized as two various, from these schemes, we can find out that when the number of attributes increases, the computation of these schemes increase. After Seo et al. [44] proposed a scheme with a constant number of paring operations, the computation cost is less than other schemes.

4 Conclusions

Attributed based proxy re-encryption scheme is suitable on the cloud environment, because ABPRE can let the data owner delegate the re-encryption right to the cloud for data sharing, and the data owner doesn't always be online. In existing ABPRE scheme, we survey several attribute-based proxy re-encryption schemes including two varied access policies: key policy and ciphertext policy. Moreover, we list eight security requirements to compare these schemes. The future work will focus on the development of the more security and efficient ABPRE, and let ABPRE can be applied in more application environments.

References

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," *Communications of the ACM*, vol. 53, pp. 50–58, 2010.

[2] G. Ateniese, K. Fu, M. Gree, and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," *Proceedings of the Network and Distributed System Security Symposium*, 2005.

[3] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," *Proceedings of IEEE Symposium on Security and Privacy*, pp. 321V-334, 2007.

[4] M. Blaze, G. Bleumer, and M. Strauss, "Divertible protocols and atomic proxy cryptography," *Proceedings of EUROCRYPT*, pp. 127–144, 1988.

[5] R. Canetti and S. Hohenberger, "Chosen-ciphertext secure proxy re-encryption," *Proceedings of the 14th ACM conference on Computer and Communication Security*, pp. 185–194, 2007.

[6] L. Cheung and C. Newport, "Provably Secure Ciphertext Policy ABE," *Proceedings of the ACM conference on Computer and communications security*, pp. 456–465, 2007.

[7] M. Y. Chen, and C. C. Yang, and M. S. Hwang, "Privacy Protection Data Access Control," *International Journal of Network Security*, vol. 15, no. 6, pp. 391–399, 2013.

[8] C. K. Chu and W. G. Tzeng, "Identity-based proxy re-encryption without random oracles," *Information Security*, vol. 4779 of LNCS, pp. 189–202, 2007.

[9] M. D. Dikaiakos, D. Katsaros, P. Mehra, G. Pallis, and Athena Vakali, "Cloud computing: Distributed internet computing for it and scientific research," *IEEE Internet Computing*, vol. 13, pp. 10 –13, 2009.

[10] J. M. Do, Y. J. Song, and N. Parko, "Attribute Based Proxy Re-encryption for Data Confidentiality in Cloud Computing Environments," *Proceedings of the IEEE Conference on Computers, Networks, Systems, and Industrial Engineering*, pp. 248–251, 2011.

[11] K. Emura, A. Miyaji, A. Nomura, K. Omote, and M. Soshi, "A ciphertext-policy attribute-based encryption scheme with constant ciphertext length," *Proceedings of the Information Security Practice and Experience*, pp. 13–23, 2009.

[12] V. Goyal, O. Pandey, A. Sahai, and B. 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.

[13] M. Green and G. Ateniese, "Identity-Based Proxy Re-encryption," *Applied Cryptography and Network Security*, vol. 4521 of LNCS, pp. 288–306, 2007.

[14] M. Green, S. Hohenberger, and B. Waters, "Outsourcing the decryption of ABE ciphertexts," *Proceedings of the 20th USENIX conference on Security*, 2011.

[15] S. Guo, Y. Zeng, J. Wei, and Q. Xu, "Attribute-based re-encryption scheme in the standard model," *Wuhan University Journal of Natural Sciences*, vol. 13, no. 5 pp. 621–625, 2008.

Table 2: The criteria of an ideal attribute-based proxy re-encryption scheme

Item	Liang et al. [33]	Luo et al. [38]	Yu et al. [57]	Yu et al. [58]	Do et al. [10]	Seo et al. [44]
C1	Y	Y	Y	Y	Y	Y
C2	Y	Y	Y	Y	Y	Y
C3	Y	Y	Y	Y	Y	Y
C4	Y	Y	Y	Y	Y	Y
C5	Y	Y	N	N	N	Y
C6	Y	Y	N	N	N	Y
C7	Y	Y	Y	Y	Y	Y
C8	Y	Y	N	Y	Y	Y

Table 3: Properties comparison

Item	Liang et al. [33]	Luo et al. [38]	Yu et al. [57]	Yu et al. [58]	Do et al. [10]	Seo et al. [44]
Policy	Ciphertext	Ciphertext	Key	Ciphertext	Key	Ciphertext
Assumption	CTDH and ADBDH	DBDH	–	DBDH	–	CTDH and ADBDH

Table 4: Expressiveness comparison [38]

Schemes	Expressiveness
Liang et al. [33]	AND gates on positive and negative attributes with wildcards
Luo et al. [38]	AND gates on multi-valued and negative attributes with wildcards
Yu et al. [57]	Monotonic access structure
Yu et al. [58]	AND gates on positive and negative attributes with wildcards
Do et al. [10]	Monotonic access structure
Seo et al. [44]	AND gates on positive and negative attributes with wildcards

Table 5: Performance comparison: Part one

Item	Liang et al. [33]	Luo et al. [38]	Yu et al. [58]	Seo et al. [44]
Encryption	$(n+2)E+2M$	$(n+2)E+2M$	$(n+1)E+2M$	$(n+2)E+2M$
Decryption	$(n+2)C_e+2M$	$(2n)C_e+3M$	–	$2C_e+(3n+2)E+2M$
Re-encryption	$(n+1)C_e+M$	$(2n+1)C_e+(n+1)M$	$(n)E$	$2C_e+(3n)E+M$
Re-decryption	$(n+3)C_e+4M$	$(2n+1)C_e+5M$	$(n+1)C_e+2M$	$3C_e+(3n)E+4M$

Table 6: Performance comparison: Part two

Item	Yu et al. [57]	Do et al. [10]
New File Creation	$\|A_{CT}\|E+M$	$\|A_{CT}\|E+M$
File Access	$\|L_{A_{U-KP}}\|C_e+E$	$\|L_{A_{U-KP}}\|C_e+E$
User Revocation	$\|D\|E+\|D\|S$	$\|D\|E+\|D\|S$

[16] S. Hohenberger, G. N. Rothblum, A. Shelat, and V. Vaikuntanathan, "Securely Obfuscating Re-encryption," *Theory of Cryptography*, vol. 4392 of LNCS, pp. 233–252, 2007.

[17] S. T. Hsu, C. C. Yang, and M. S. Hwang, "A study of public key encryption with keyword search," *International Journal of Network Security*, vol. 15, no. 2, pp. 71–79, 2013.

[18] J. Hur and D. K. Noh, "Attribute-Based Access Control with Efficient Revocation in Data Outsourcing Systems," *IEEE transaction on parallel and distributed systems*, vol. 22, no. 7, pp. 1214–1221, 2011.

[19] M. S. Hwang, C. C. Lee, S. J. Hwang, "Cryptanalysis of the Hwang-Shi proxy signature scheme", *Fundamenta Informaticae*, vol. 53, no. 2, pp. 131-134, Nov. 2002.

[20] M. S. Hwang, C. C. Lee, S. F. Tzeng, "A new proxy signature scheme for a specified group of verifiers," *Information Sciences*, vol. 227, pp. 102–115, 2013.

[21] M. S. Hwang, I. C. Lin, Eric J. L. Lu, "A secure non-repudiable threshold proxy signature scheme with known signers", *Informatica*, vol. 11, no. 2, pp. 137-144, Apr. 2000.

[22] M. S. Hwang, Eric J. L. Lu, I. C. Lin, "A practical (t, n) threshold proxy signature scheme based on the RSA cryptosystem", *IEEE Transactions on Knowledge and Data Engineering*, vol. 15. no. 6, pp. 1552-1560, Nov./Dec. 2003.

[23] M. S. Hwang, S. F. Tzeng, S. F. Chiou, "A non-repudiable multi-proxy multi-signature scheme," *Innovative Computing, Information and Control Express Letters*, vol. 3, no. 3, pp. 259–264, 2009.

[24] M. S. Hwang, S. F. Tzeng, C. S. Tsai, "Generalization of proxy signature based on elliptic curves", *Computer Standards & Interfaces*, vol. 26, no. 2, pp. 73-84, Mar. 2004.

[25] W. Itani, A. Kayssi, and A. Chehab, "Privacy as a Service: Privacy-Aware Data Storage and Processing in Cloud Computing Architectures," *Proceedings of the 8th IEEE Conference on Dependable, Autonomic and Secure Computing*, pp. 711–716, 2009.

[26] S. Kamara and K. Lauter, "Cryptographic cloud storage," *Proceedings of the 14th international conference on Financial cryptograpy and data security*, pp. 136–149, 2010.

[27] H. Koshutanski, "A Survey on Distributed Access Control Systems for Web Business Processes," *International Journal of Network Security*, vol. 9, no. 1, pp. 61–69, 2009.

[28] C. C. Lee, P. S. Chung, and M. S. Hwang, "A survey on attribute-based encryption schemes of access control in cloud environments," *International Journal of Network Security*, vol. 15, no. 4, pp. 231–240, 2013.

[29] C. C. Lee, S. T. Hsu, and M. S. Hwang, "A Study of Conjunctive Keyword Searchable Schemes," *International Journal of Network Security*, vol. 15, no. 5, pp. 311–320, 2013.

[30] C. C. Lee, T. C. Lin, M. S. Hwang, "Generalization of proxy signature based on factorization," *International Journal of Innovative Computing, Information and Control*, vol. 7, no. 3, pp. 1039–1054, 2011.

[31] J. Li, K. Ren, B. Zhu, and Z. Wan, "Privacy-aware attribute-based encryption with user accountability," *Information on Security*, vol. 5735 of LNCS, pp. 347–362, 2009.

[32] L. H. Li, S. F. Tzeng, M. S. Hwang, "generalization of proxy signature based on discrete logarithms", *Computers & Security*, vol. 22, no. 3, pp. 245-255, 2003.

[33] 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*, pp. 276–286, 2009.

[34] B. Libert and D. Vergnaud, "Unidirectional chosen-ciphertext secure proxy re-encryption," *Public Key Cryptography*, vol. 4939 of LNCS, pp. 360–379, 2008.

[35] Q. Liu, C. C. Tan, J. Wu, and GuojunWang, "Reliable re-encryption in unreliable clouds," *Proceedings of the IEEE Global Telecommunications Conference*, pp. 1–5, 2011.

[36] Q. Liu, G. Wang, and J. Wu, "Time-based proxy re-encryption scheme for secure data sharing in a cloud environment," *Information Sciences. In Press*, 2012.

[37] Eric J. L. Lu, M. S. Hwang, and C. J. Huang, "A new proxy signature scheme with revocation," *Applied Mathematics and Computation*, vol. 161, no. 3, pp. 799–806, 2005.

[38] S. Luo, J. Hu, and Z. Chen, "Ciphertext Policy Attribute-Based Proxy Re-encryption," *Information and Communications Security*, vol. 6476 of LNCS, pp. 401–415, 2010.

[39] M. Mambo and E. Okamoto, "Proxy Cryptosystems: Delegation of the Power to Decrypt Ciphertexts," *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E80-A, no. 1, pp. 54–63, 1997.

[40] T. Matsuo, "Proxy re-encryption systems for identity-based encryption," *Pairing-Based Cryptography*, vol. 4575 of LNCS, pp. 247–267, 2007.

[41] T. Mizuno and H. Doi, "Hybrid Proxy Re-encryption Scheme for Attribute-Based Encryption," *Information security and cryptology*, vol. 6151 of LNCS, pp. 288–302, 2011.

[42] D. Nali, C. Adams, and A. Miri, "Using Threshold Attribute-based Encryption for Practical Biometric-based Access Control," *International Journal of Network Security*, vol. 1, no. 3, pp. 173–182, 2005.

[43] A. Sahai and B. Waters, "Fuzzy identity based encryption," *Advances in Cryptology*, vol. 3494 of LNCS, pp. 457–473, 2005.

[44] H. Seo and H. Kim, "Attribute-based Proxy Re-encryption with a Constant Number of Pairing Operations," *Journal of Information and Communication Convergence Engineering*, vol. 10, no. 1, pp. 53–60, 2012.

[45] H. Seo and H. Kim, "Zigbee security for visitors in home automation using attribute based proxy re-encryption," *Proceedings of the IEEE 15the International Symposium on Consumer Electronic*, pp. 304–307, 2011.

[46] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[47] J. S. Su, D. Cao, X. F. Wang, Y. P. Su, and Q. L. Hu, "Attribute-Based Encryption Schemes," *Journal of Software*, vol. 6, no. 22, pp. 1299–1315, 2012.

[48] S. Subashini and V. Kavitha, "A survey on security issues in service delivery models of cloud computing," *Journal of Network and Computer Applications*, vol. 34, no. 1, pp. 1–11, 2011.

[49] X. Tian, X. Wang, and A. Zhou , "DSP Re-encryption Based Access Control Enforcement Management Mechanism in DaaS," *International Journal of Network Security*, vol. 15, no. 1, pp. 28–41, 2013.

[50] C. S. Tsai, S. F. Tzeng, M. S. Hwang, Improved non-repudiable threshold proxy signature scheme with known signers, *Informatica*, vol. 14, no. 3, pp. 393-402, 2003.

[51] S. F. Tzeng, C. C. Lee, M. S. Hwang, "A batch verification for multiple proxy signature," *Parallel Processing Letters*, vol. 21, no. 1, pp. 77–84, 2011.

[52] S. F. Tzeng, C. C. Lee, and T. C. Lin, "A novel key management scheme for dynamic access control in a hierarchy," *International Journal of Network Security*, vol. 12, no. 3, pp. 178–180, 2011.

[53] S. F. Tzeng, M. S. Hwang, C. Y. Yang, "An improvement of nonrepudiable threshold proxy signature scheme with known signers", *Computers & Security*, vol. 23, no. 2, pp. 174-178, Apr. 2004.

[54] S. F. Tzeng, C. Y. Yang, M. S. Hwang, "A nonrepudiable threshold multi-proxy multi-signature scheme with shared verification", *Future Generation Computer Systems*, vol. 20, no. 5, PP. 887-893, June 2004.
[55] J. Wang, Y. Zhao, S. Jiang, and J. Le, "Providing privacy preserving in cloud computing," *Proceedings of the IEEE Conference on Test and Measurement*, vol. 2, pp. 213–216, 2009.
[56] C. Y. Yang, S. F. Tzeng, M. S. Hwang, "On the efficiency of nonrepudiable threshold proxy signature scheme with known signers", *The Journal of Systems and Software*, vol. 73, no. 3, pp. 507-514, Nov. 2004.
[57] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing," *Proceedings of IEEE INFOCOM*, pp. 534–542, 2010.
[58] S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute based data sharing with attribute revocation," *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, pp. 261–270, 2010.
[59] M. Zhou, R. Zhang, W. Xie, W. Qian, and A. Zhou, "Security and Privacy in Cloud Computing: A Survey," *Proceedings of the 6th International Conference on Semantics Knowledge and Grid (SKG)*, pp. 105–112, 2010.

Pei-Shan Chung received her B. M. in information Management from Chung Yuan Christian University, Jungli, Taiwan, ROC, in 2010. She is currently pursuing the M.S. degree with the Department of Management Information Systems from National Chung Hsing University. Her research interests include information security, cloud computing, and cryptography.

Chih-Wei Liu received his M.S. in Soil And Water Conservation from National Chung Hsiung University, Taichung, Taiwan, ROC, in 2008. He is currently pursuing the Ph.D. degree from Computer Science & Information Engineering Department of Asia University, Wufeng, Taiwan. His research interests include information security, cloud computing, and information law.

Min-Shiang Hwang received the B.S. in Electronic Engineering from National Taipei Institute of Technology, Taipei, Taiwan, Republic of China (ROC), in 1980; the M.S. in Industrial Engineering from National Tsing Hua University, Taiwan, in 1988; and the Ph.D. in Computer and Information Science from National Chiao Tung University, Taiwan, in 1995. He also studied Applied Mathematics at National Cheng Kung University, Taiwan, from 1984-1986. Dr. Hwang passed the National Higher Examination in field "Electronic Engineer" in 1988. He also passed the National Telecommunication Special Examination in field "Information Engineering", qualified as advanced technician the first class in 1990. From 1988 to 1991, he was the leader of the Computer Center at Telecommunication Laboratories (TL), Ministry of Transportation and Communications, ROC. He was also a chairman of the Department of Information Management, Chaoyang University of Technology (CYUT), Taiwan, during 1999-2002. He was a professor and chairman of the Graduate Institute of Networking and Communications, CYUT, during 2002-2003. He is currently a professor of the department of Management Information System, National Chung Hsing University, Taiwan, ROC. He obtained 1997, 1998, 1999, 2000, and 2001 Outstanding Research Awards of the National Science Council of the Republic of China. He is a member of IEEE, ACM, and Chinese Information Security Association. His current research interests include electronic commerce, database and data security, cryptography, image compression, and mobile computing. Dr. Hwang had published 170+ articles on the above research fields in international journals.