CP-ABE Access Control that Block Access of Withdrawn Users in Dynamic Cloud

Yong-Woon Hwang[1] and Im-Yeong Lee[2*]
[1,2]Department of Computer Science and Engineering, Soonchunhyang University
Asan, South Korea
[e-mail: hyw0123@sch.ac.kr, imylee@sch.ac.kr]
*Corresponding author: Im-Yeong Lee

Received April 16, 2020; revised July 23, 2020; accepted September 6, 2020; published October 31, 2020

Abstract

Recently, data can be safely shared or stored using the infrastructure of cloud computing in various fields. However, issues such as data security and privacy affect cloud environments. Thus, a variety of security technologies are required, one of them is security technology using CP-ABE. Research into the CP-ABE scheme is currently ongoing, but the existing CP-ABE schemes can pose security threats and are inefficient. In terms of security, the CP-ABE approach should be secure against user collusion attacks and masquerade attacks. In addition, in a dynamic cloud environment where users are frequently added or removed, they must eliminate user access when they leave, and so users will not be able to access the cloud after removal. A user who has left should not be able to access the cloud with the existing attributes, secret key that had been granted. In addition, the existing CP-ABE scheme increases the size of the ciphertext according to the number of attributes specified by the data owner. This leads to inefficient use of cloud storage space and increases the amount of operations carried out by the user, which becomes excessive when the number of attributes is large. In this paper, CP-ABE access control is proposed to block access of withdrawn users in dynamic cloud environments. This proposed scheme focuses on the revocation of the attributes of the withdrawn users and the output of a ciphertext of a constant-size, and improves the efficiency of the user decryption operation through outsourcing.

Keywords: Attributes Based Encryption, Access Control, Attribute revocation, Constant-size ciphertext, Cloud, Data sharing

1. Introduction

Recently, cloud computing infrastructure is used in various fields, and users can safely share or store data on an external server (cloud). Above all, cloud computing is convenient for users because it can be used over the internet. However, as shown in **Fig. 1**, there are a variety of security threats in the cloud environment. Users can protect their data by safely storing their data in the cloud provided by cloud service providers, but the service provider can use the data at any time. There are also security threats that could cause the data to be leaked or lost due to an attacker. Therefore, security technology is needed to protect the data of the data owner in the cloud environment.

Among many security technologies, Attribute Based Encryption (ABE) is a technology that can encrypt and decrypt data with various attributes of users, and is suitable for cloud environments [1-3]. There are ciphertext policy-attribute-based encryption (CP-ABE) method and a key policy attribute-based encryption (KP-ABE) method. The differences between the two methods are detailed in section 2. CP-ABE is an encryption technology that allows access by multiple users. The proposed scheme in this paper studies CP-ABE. Briefly explaining CP-ABE, when the owner of the data encrypts the message and sends it to the storage server, only users with the attributes contained in the ciphertext can access and decrypt the ciphertext. In other words, the users of the data can be specified by the data owner, not the server, thereby reducing the dependence on the server. However, the disadvantage of attribute-based encryption is that it requires an server(authority) to manage users attributes. Despite these disadvantage, the CP-ABE access control schemes is often used in data sharing environments in the cloud.

To date, research has been conducted on a CP-ABE-based access control system with various requirements in a cloud environment. but some schemes are vulnerable to a variety of security threats or are inefficient. Especially from a security point of view, the CP-ABE scheme should not be accessible after unsubscribing because the user's attributes are

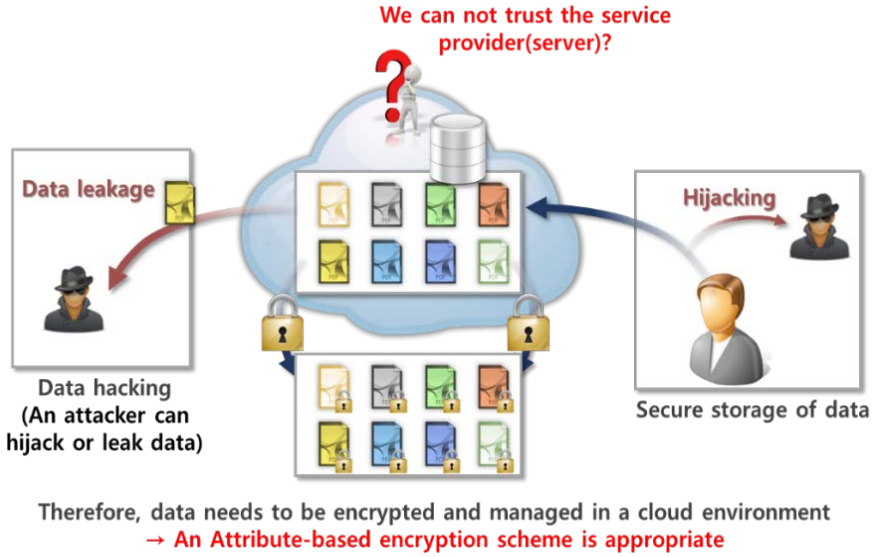

Fig. 1. Security Threats in Cloud Environments

completely removed when the user leaves the existing cloud environment [4-12]. A user who has left creates a problem if he / she can access the cloud environment with their existing attributes and the secret key they had. For example, unauthorized users could access the cloud to download and leak data. It would not protect sensitive document leakage and the data of the enterprise or data owner. In addition, a variety of attacks can occur, such as user-collusion attacks and masquerade attacks, and users who cannot access the cloud can access the cloud with the attributes of other users. From the efficiency point of view, the ciphertext increases in size depending on the number of attributes selected by the data owner. Therefore, storage space may be wasted, and there are many calculations that the user can decrypt the ciphertext, which is inefficient [13-14].

In this paper, we proposed a CP-ABE access control that block access of withdrawn users in dynamic cloud. This allows only authorized users to access the cloud and safely share stored data. The contributions in this paper are as follows. 1) Access control of removed users through a user registration list and a revocation list; A user revocation list is used to block access by users who have been removed. After the initial user registration, a registered user can use the cloud, and a removed user is registered on the revocation list and the cloud cannot be used during the period registered in the revocation list. 2) Output a ciphertext with a constant size; Since the size of the ciphertext is constant generated regardless of the number of attributoperties included in the access structure, it is possible to improve the storage space efficiency in the cloud and reduce the amount of computation required when a user decrypt ciphertext. 3) Reduced amount of decryption computation of user's ciphertext due to partial decryption; Divides existing cloud server into storage and trusted access control (AC) server. In this case, if the accessing user is registered, the AC server authenticates the user and performs partial decryption to determine whether or not it matches the attribute in the access structure included in the ciphertext. The user receives the partially decrypted ciphertext and performs final decryption with the secret key transmitted from the trusted third party (TTP).

The cloud environment proposed in this paper is a dynamic cloud environment, not all cloud environments. In this case, “dynamic cloud” refers to an in-company cloud where registration and withdrawals frequently occur, or a cloud environment that can be used for a purchase period when purchasing space or package files..

This paper is a study of CP-ABE access control to block access of withdrawn users in dynamic cloud environments. Section 2 introduces the mathematical models and definitions used in this study, as well as the CP-ABE scheme previously studied. Section 3 describes the security requirements, and section 4 proposes a new scheme. Section 5 analyzes the security and efficiency of the proposed scheme. Finally, we conclusion in section 6.

2. BACKGROUNDS

In this section, we discuss relevant background materials, attribute-based encryption, and CP-ABE techniques used in traditional cloud environments to address security vulnerabilities in cloud environments.

2.1 PRELIMINARY WORKS

2.1.1 Bilinear Map

Bilinear mapping has been proposed as a tool to attack elliptic curve cryptosystems in the past, but it has been used recently as a cryptographic tool for information security [2].

A bilinear pairing is also called a bilinear map.
· q: Very large prime
· G_1: Additive group on the elliptic curves with a staggered q
· G_2: Multiplication over a finite field with the number of stars
· $P, Q, R \in_R G_2$
· $a, b, c \in_R Z_q^*$

A bilinear map is a function that satisfies the following properties:$G_1 X G_1 \rightarrow G_2$.
· Bilinear: For any P,Q,R, $e(P,Q+R)=e(P,Q)\cdot e(P,R)$ and $e(P+Q,R)=e(P,R)\cdot e(Q,R)$ are established.
· Non-degenerate: For all pairs P, Q of G_1, $e(P,Q)\neq 0$.
· Computable: For any P, Q, there must be an efficient algorithm that can compute $e(P,Q)$.

2.1.2 Complexity Assumption

Bilinear Diffie-Hellman (BDH) Assumption

The deterministic BDH assumption means that when two pairs $(g^a, g^b, g^c, \mathrm{T} = \mathrm{e}(g,g)^{abc})$ and $(g^a, g^b, g^c, \mathrm{W} = \mathrm{e}(g,g)^z)$ are given, Algorithm A, which can distinguish two pairs, has no-negligible probability. Where $a, b, c, z \in Z_p$. In order for Algorithm A that solves the deterministic BDH assumption to benefit from ϵ, Algorithm A must satisfy $|\Pr[\mathrm{A}(g^a, g^b, g^c, \mathrm{T}) = 1] - \Pr[\mathrm{A}(g^a, g^b, g^c, \mathrm{W}) = 1]| \geq \epsilon$.

Bilinear Diffie-Hellman Exponent (BDHE) Assumption

The deterministic BDHE assumption means that given $(\mathrm{h}, g, g^{\alpha} \ldots, g^{\alpha\beta}, g^{\alpha\beta+2}, \ldots g^{\alpha 2\beta})$, there is no algorithm A that can calculate $\mathrm{T} = \mathrm{e}(\mathrm{h}, g)^{\alpha\beta+1}$ with a non-negligible probability. And if this is defined as $g_i = g^{\alpha^i}(\mathrm{i} = 1, \ldots, 2\mathrm{B})$, $g_{\alpha,\beta} = (g_1, \ldots, g_B, g_{B+2}, \ldots, g_{2B})$, the two pairs can be defined as $(h, g, g_{\alpha,\beta}, \mathrm{T} = \mathrm{e}(\mathrm{h}, g)^{\alpha\beta+1})$, $(h, g, g_{\alpha,\beta}, W = \mathrm{e}(\mathrm{h}, g)^z)$. Where $h, g \in G_1$. In order for Algorithm A that solves the deterministic BDH assumption to benefit from ϵ, Algorithm A must satisfy $|\Pr[\mathrm{A}(h, g, g_{\alpha,\beta}, \mathrm{T}) = 1] - \Pr[\mathrm{A}(h, g, g_{\alpha,\beta}, \mathrm{W}) = 1]| \geq \epsilon$.

2.2 Attribute-based Encryption

In 2005, Sahai et al. proposed identity-based encryption [15]. This performs encryption and introduces redundancy based on a set of attributes (address, name) for each entity and the

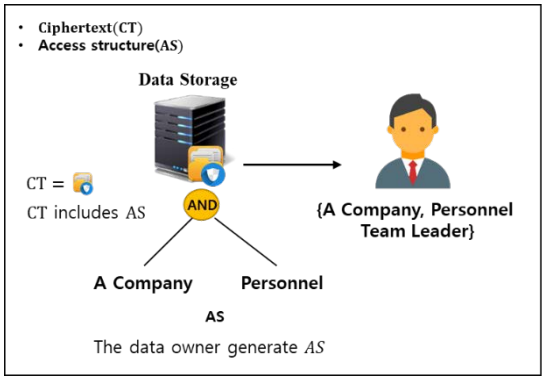

Fig. 2. CP-ABE

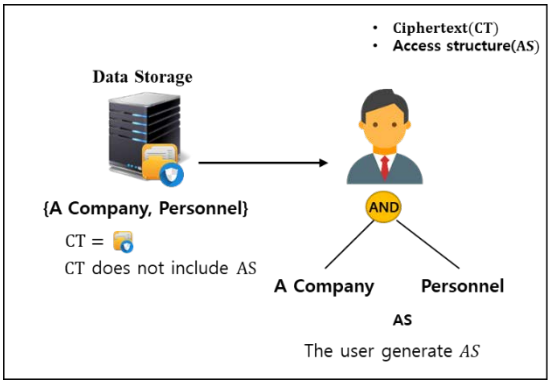

Fig. 3. KP-ABE

access policy (access structure) for accessing a given set of attributes. There are two types of attribute-based encryption: CP-ABE and KP-ABE [1]. **Fig. 2** shows the CP-ABE. When the data owner creates a ciphertext, it creates an access structure with user's attributes, through which the message is encrypted and sends it to the cloud. Thereafter, the user attempts decryption according to the attribute set of the user. As shown in **Fig. 2**, if the user satisfies the [A company, Personnel] attribute, the ciphertext can be decrypted. **Fig. 3** shows the KP-ABE. Data owners use a set of user attributes to encrypt messages and send them to the cloud. Then the user creates an access structure with a set of attributes they have, generates a key that can decrypt the ciphertext, and attempts to decrypt it. As shown in **Fig. 3**, if the data owner encrypts and transmits the data with a user attribute set [A company, personnel]. The user creates an access structure with the [A Company], [Personal] attributes they have, and then creates a secret key. It then decrypts the ciphertext with the secret key. The difference between CP-ABE and KP-ABE depends on the subject who creates the access structure. If the data owner creates the access structure, CP-ABE is the model, and the data user creates the access structure is KP-ABE. The use of attribute-based encryption depends on the environment. However, the CP-ABE method is often used in a cloud environment because it has the advantage that only users with the attributes selected by the data owner can decrypt ciphertexts stored on the server. In this paper, CP-ABE technology is used to control user access to encrypted data [3].

2.3 CP-ABE

Among access control technologies used in the cloud environment, CP-ABE technology has advantages over other one-to-one traditional encryption technologies. First, CP-ABE enables granular access control of data in encrypted form. It can be applied to systems that can control access to data in a system environment in which data is shared in the cloud. Second, it provides a great solution to data confidentiality. Since the plaintext is not disclosed to the cloud, it is possible to safely store the data owner's data on an unreliable cloud provider. In addition, although the generation of the secret key for the user occurs only once, but it can be used to decrypt all applicable ciphertexts stored in the cloud, thereby reducing communication overhead [16]. Third, even if the attributes are the same for each user, the secret key given by

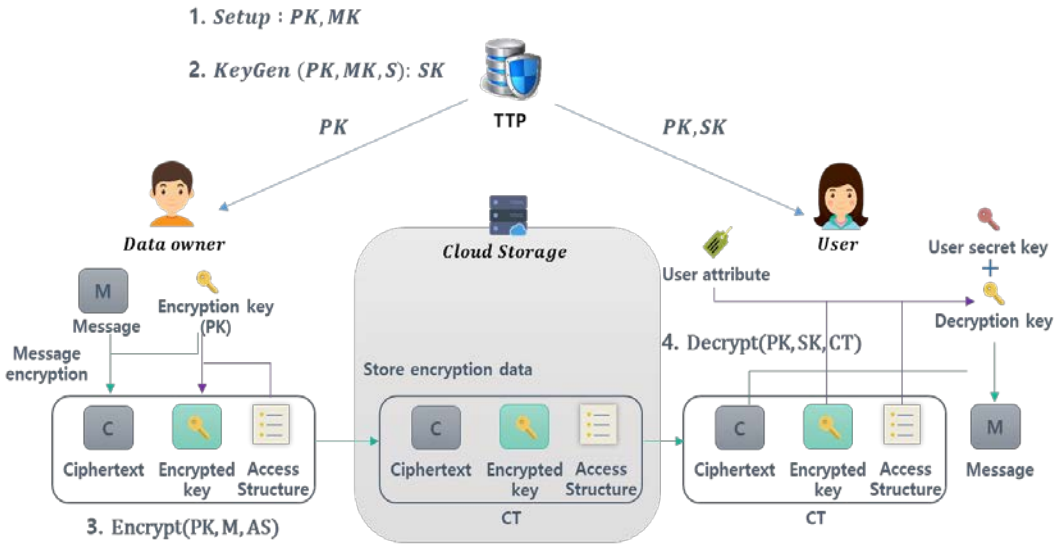

Fig. 4. CP-ABE scheme overall flow

the user contains random numbers or polynomials, so it is safe from collusion or spoofing attacks. Lastly, CP-ABE can be integrated with the proxy re-encryption technology of the cloud, which can be applied to various fields, thereby making it highly scalable [16].

However, there are some weaknesses associated with CP-ABE. First, it is difficult to deal with user property and user revocation issues in a dynamic environment where user attributes can change with time [17]. Trusting the cloud server can lead to collusion attacks. In this attack, a revoked user can collusion with a cloud servers to assemble information to gain unauthorized access to data. Therefore, further improvement is required to adopt CP-ABE [18-19].

2.3.1 System model

The **Fig. 4** shows the basic CP-ABE scheme. It consists of four steps: Setup, Key Generation (KeyGen), Encryption (Encrypt) and Decryption (Decrypt). In a cloud environment, the user and the data owner are provided with the necessary information via each phase as shown in **Fig. 5**. The description of each phase is as follows.

- Setup(k): A trusted third party (TTP) generates the public parameter (PK) and the master key (MK) from the security parameter k.
- KeyGen(MK, S, PK): Use the user attribute set S, generate the secret key (SK) for the attribute value using MK and PK
- Encrypt(PK, M, AS): To proceed with data encryption, the data owner creates a tree-type access structure (AS) from the user's attributes to access the data stored on the server. The message M is then encrypted with PK and AS to generate CT and stored in the cloud storage.
- Decrypt(PK, SK, CT): The user receives the CT from the cloud storage and decrypts the ciphertext using the secret key corresponding to the user's secret key and the set of query attributes to obtain the message M.

2.3.2 The Revocation Problem

The revocation problem is a change of their permissions expire the access rights of a user when a registered user in the cloud environment withdraw. The revocation problem can be divided

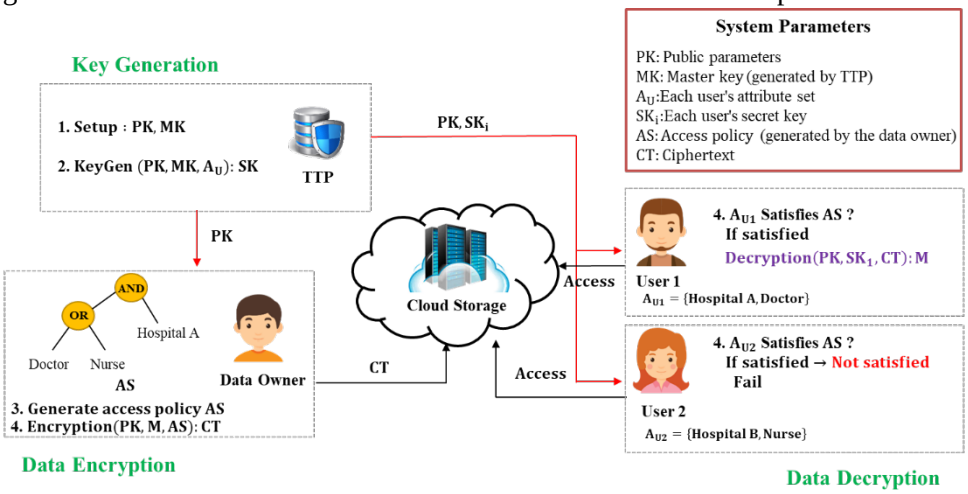

Fig. 5. Access Control Method for CP-ABE in Cloud

into two categories. The first is attribute cancellation, which occurs when a user's attributes are removed. For example, if a company degrades a user's role from manager to general employee, the user's attributes and access rights will be lost. Also, the same set of attributes can be associated with another user's secret key, causing significant computational overhead throughout the revocation process. This occurs because the user attributes that are not revoked and all associated keys must be updated, and the ciphertext associated with the revoked attributes must be re-encrypted [7, 18]. The second is termination of a user, which occurs when the user leaves the system. When the user's registered period expires or the user withdraws, then the data should not be accessible, and the access right must be revoked access rights so that data stored in the cloud cannot be decrypted. Accordingly, when designing a CP-ABE based access control method in a cloud environment, it is necessary to consider a mechanism that can revoke a user's specific access authority. In particular, in a dynamic cloud environment where registration and withdrawal of users frequently occur, handling termination issues is essential, and the following important features are required to solve them:

· Block access of users not registered in the cloud.

· It must be secure against collusion attacks and invalidate the access key of the withdrawn user [7].

· In the process of revocation the withdrawing user, the computational overhead should be minimized.

· Forward security must be supported. In other words, the user with the attribute of the withdrawn user should not be able to access the existing cloud, and should not be able to decrypt the encrypted text stored in the cloud [20].

2.3.3 Previously Proposed CP-ABE Schemes

Research into the CP-ABE data access system in the cloud environments is continuously being studied, but some CP-ABE systems are still exposed to security threats, or the amount of computation is inefficient. In particular, a masquerade attack may occur when a user has left in the Sekhar scheme [4], and in the Zhu scheme [5] attackers can access data using other users' attributes. The Xu scheme [6], Yang scheme [7], and Ramesh scheme [8] apply attribute retraction upon user withdrawal. However, the Xu scheme and Yang scheme are inefficient because when users withdraw, they update both the key of other users participating in the cloud and the ciphertext stored in the cloud. In the Ramesh scheme, there is an attribute authority (AA) that manages attributes when users withdraw, but this introduces inefficiencies, as it is required to continuously update the other users public keys. Therefore, an outsourcing technique that can increase the efficiency of users decryption computations by processing a part of the computation on the cloud server when the user want to decrypt the ciphertext is required. According to the Liu scheme [11] proposed in 2018, the revocation list is included in the ciphertext to terminate it directly. More specifically, by including the timed revocation list in the ciphertext, when the specified time expires, the user cannot decrypt the ciphertext. This has the advantage that the data owner can specify the time for the subject who accesses his data, and this allows immediate user termination. However, since the revocation list is included in the ciphertext, the encryption computation amount and the user's key generation computation amount will increase compared to the existing CP-ABE computation amount. According to the Zhao scheme [12] proposed in 2019, it was satisfied with attribute removal, ciphertext

Table 1. Compare the previously proposed CP-ABE scheme

CP-ABE Scheme	Year	Attribute revocation	Ciphertext size	Support outsourcing decoding operation
Sekhar scheme [4]	2012	○	Increases by the number of attributes	X
Zhu scheme [5]	2015	○	Increases by the number of attributes	X
Xu scheme [6]	2012	○	Increases by the number of attributes	X
Yang scheme [7]	2013	○	Increases by the number of attributes	X
Ramesh scheme [8]	2016	○	Increases by the number of attributes	X
Xia scheme [9]	2016	○	Increases by the number of attributes	X
Liu scheme [11]	2018	○	Increases by the number of attributes	X
Zhao scheme [12]	2019	○	Constant-size ciphertext	○
Hahn scheme [13]	2016	X	Constant-size ciphertext	○
Teng Wei scheme [14]	2017	X	Constant-size ciphertext	X
Goal of our proposed scheme	2020	○	Constant-size ciphertext	○

○: Provide; X: Not provide

length fixation, outsourcing support, verifiable outsourcing, etc. as keywords to solve problems that may occur in CP-ABE. In particular, the attribute removal part has a version for each ciphertext, and the user can decrypt the ciphertext only when there is a version in his key. When the version passes, the encrypted text stored in the cloud is re-encrypted and updated with the new version of the encrypted text. Therefore, if the version of the user key is different, the ciphertext cannot be decrypted. Even in this method, it is inefficient to continuously update the ciphertext when the user is terminated. In addition, since it satisfies various security requirements in addition to the removal of attributes, it is higher than the existing CP-ABE of overall computation. Also, compared to other existing CP-ABE, the size of the ciphertext increases with the number of attributes. As shown in **Table 1**, the proposed CP-ABE method blocks most users' revoked access by updating the key and ciphertext most of the time when the attribute is revoked. In addition, the methods presented in Table 1 waste storage space in the server because the size of the ciphertext increases according to the number of attributes. Therefore, a user who does not support outsourcing is burdened with the amount of computation required to decrypt the ciphertext. The Hahn scheme [13] was proposed in 2015. This is an attribute-based secure data sharing scheme that supports ciphertexts of constanted size and the outsourcing of decryption operations. However, the Hahn scheme is not suitable for cloud environments where data can be shared with others because it is based on a private cloud environment. Tang method [14] is a method proposed in 2017, and proposed an attribute-based access control method that uses a constant sized ciphertext to increase the efficiency of server storage space and user decryption computation. However, when a user leaves the cloud, all ciphertexts and keys must be updated, and the amount of computation required for a user to decrypt the ciphertext is also high.

2.3.4 Security model

The attack model of the proposed scheme is constructed similar to the attack on the constant-size CP-ABE scheme introduced by Zhou and Huang in 2010 [21].

The attack model introduced in this chapter is based on a semantic security game. In this paper, the CP-ABE technique finally derives the probability that the attacker can win the CP-ABE technique game through sending and receiving messages between the probabilistic polynomial time adversaries (PPT adversary) and the challenger for the chosen plaintext attack (CPA) security game described below. The game details consist of the following steps:

· **Init** The adversary selects and provides the ciphertext access structure W to attack the challenger.

· **Setup:** The challenger performs the setup phase, creates system parameters (PK) for the ciphertext attribute set, and transmits it to the adversary.

· **Phase :** The adversary requests a random user private key for access structure L from the challenger. The attacker can request as many secret keys as needed, and the obtained private key is secret key that cannot be decrypted. At this time, $L \nvDash W$ is satisfied.

· **Challenge:** The adversary creates two messages, m_0 and m_1, and sends them to the challenger. The challenger selects a random value $b \in \{0,1\}$, performs encryption, and sends the ciphertext $\{< C_{0,} C_1 >, Key_b\}$ to the attacker.

· **Guess:** The adversary guesses $b' \in \{0,1\}$ for the ciphertext. If $b' = b$ while satisfying $L \nvDash W$, we define the adversary to win. In this game, the definition of the probability an adversary can win the game is equal to $Pr[b' = b] - 1/2$.

The proposed algorithm is semantically secure if the adversary has a negligible benefit in the above game within probabilistic polynomial time [22-26].

3. SECURITY REQUIREMENTS

In this section, we discuss the security requirements that must be met to protect data in a cloud environment.

· **User collusion and masquerade attacks:** A user collusion attack can obtain the decryption keys of other users through collusion between the cloud server and a user who can access the data. Alternatively attacker could try to guess the decryption key using the attributes of a user accessing the data, and thereby attempt a masquerade attack. Therefore, the attacker should not be able to guess the decryption key with users attributes, and should not be able to decrypt the encrypted data of other users even if the cloud server and the user conspire.

· **Access control for unauthorized users**: If the user attempting to access the cloud server is a malicious attacker, the data stored on the cloud server is not secure. Therefore requires user access and authentication techniques because only legitimate users must be able to access and view data in the cloud [27].

· **User attributes revocation:** As shown in **Fig. 6**, even after a user leaves the cloud, they can access the cloud through their existing attributes and private keys. The user can download the ciphertext, perform decryption, and leak the acquired message. To solve this problem, when a user is removed, the attributes of the user must be also removed or the ciphertext and key stored in the existing cloud updated, so that a user cannot decrypt the data even if the user previously accessed it. However, updating all the ciphertexts and keys is computationally inefficient. Therefore, there is a need for a safe and efficient method to deny access to the cloud after removal.

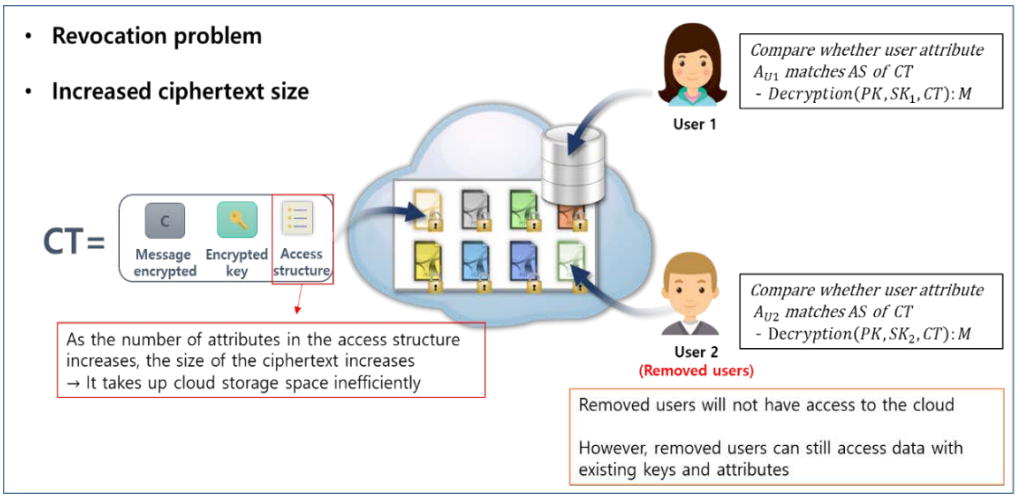

Fig. 6. Security Threats from CP-ABE

· **Efficiency:** As shown in **Fig. 6**, when encrypting data, the size of the ciphertext differs depending on the number of attributes. Also, amount of computation required for users to decrypt the ciphertext is related to the number of attributes of attributes included in the ciphertext. The computational load associated with the decryption of the ciphertext is borne by the user so it is necessary to use an efficient scheme, such as an outsourcing method, to solve this problem.

4. PROPOSED SCHEME

In this section, we propose a CP-ABE access control that block access of withdrawn users in dynamic cloud. As the proposed scheme can only be utilized by users with access rights, we can prevent access by users who have been denied access or are not authorized for access. This scheme is safe from variety of security threats that can occur in cloud environment such as user collusion and masquerade attacks. Above all, the ciphertext, which is proportional in size to the number of attributes in existing CP-ABE schemes, is restricted to a constant size, and the protocol calculations can be done efficiently by supporting an outsourcing technique. In order to reduce the user's decoding operations compared to in the previously proposed CP-ABE decryption phases, in this proposed scheme, decryption is performed by dividing it into a partial decryption and a final decryption. The AC server supports outsourcing technology as a trusted server, and a partial decryption step is performed to determine if the user attributes match the set of attributes specified in the cryptogram access structure. Also, the final decryption can only be performed by the user with the secret key. Therefore, since the user's decrypt operation amount is reduced, the user's efficiency is improved. The proposed scenario is shown in **Fig. 7**.

The proposed scheme consists of a setup phase, a key generation phase, a data encryption phase, a user data access phase, and a data decryption phase. Additionally, as shown in **Fig. 10**, a user attribute revocation phase is added to revoke a user's access rights when the user leaves the system.

The proposed scheme assumes the following; 1) The user is registered with a TTP prior to accessing the cloud. 2) The TTP manages the user's information, and the data sent by the TTP

is communicated over a secure channel. 3) The AC is a server that manages user access control and controls user access based on information provided by the TTP

In this section, we explain each phase of the proposed protocol in detail.

4.1 System Parameters

The description of the system parameters is as follows.

· TTP: Trusted third party that manages user attributes
· Storage: Cloud server that stores and manages shared data in the cloud
· Access Control: Cloud server that performs user access management and partial decryption
· MK, PK:Master key, Public parameter
· SK: Decryption key or User security key
· ID_{user}: User identifiable value
· S, A_U : A set of attribute data, User attribute data
· AS: Access structure
· TK: Tokens that can access the cloud
· AK: Decryption key generated during partial decryption
· Nonce value: Random value given for each user
· CT : Ciphertext
· T: Timestamp

4.2 System Model Object

The roles of system objects of this proposed scheme is described below.

· **Data Owner:** The data owner is the user of the cloud who is responsible for encrypting the data. Create an access structure with user attributes that can access the cloud. The data is then

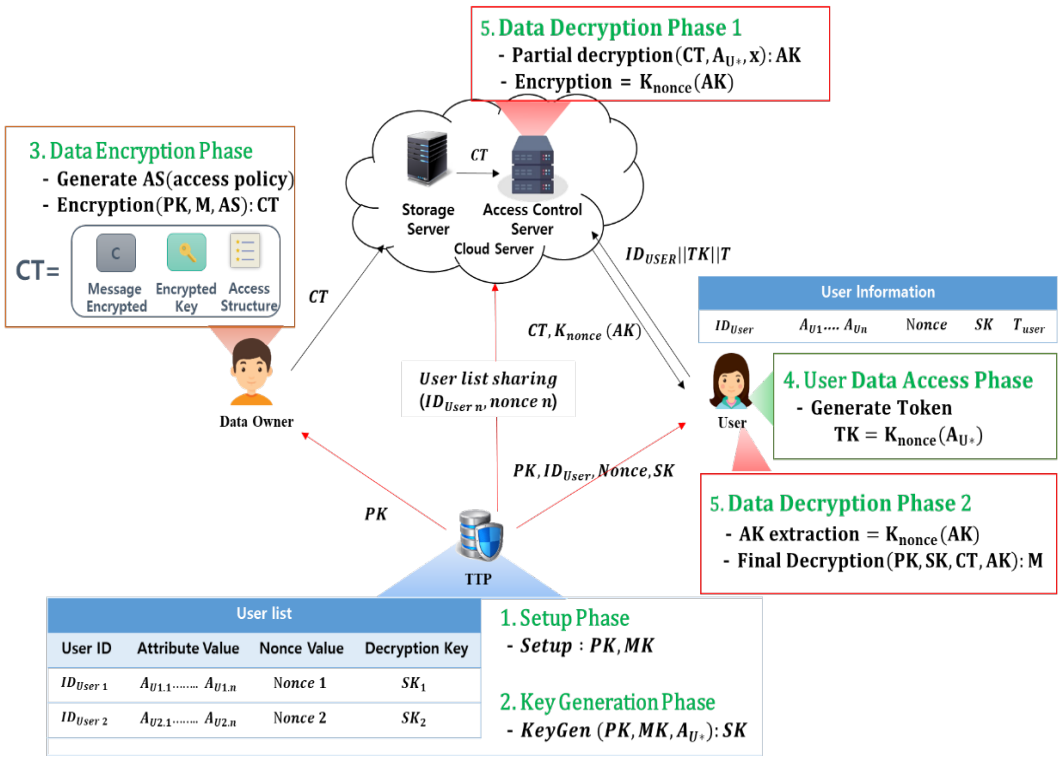

Fig. 7. Scenario of the proposed scheme

encrypted and uploaded to the cloud.

· **Cloud Server:** The cloud server consists of storage server and AC server. The storage server stores and manages encrypted data, and the AC server is an outsourcing server that can perform partial decryption of the ciphertext requested by the user after authentication as a registered user. The role of the AC server is to manage user access. Therefore, it reduces the amount of decryption operations by the user and increases the efficiency of the user's decryption operation.

· **Trusted third party** (**TTP**): Initial setting and key generation process are performed, and it is a trusted server that manages user attributes. When a registered user requests a secret key(decryption key), TTP creates a key to decrypt the ciphertext with the registered user's attributes and sends it safely to the user.

· **Data User:** A user is an entity who accesses encrypted data stored in cloud storage using its attributes. Partial decryption is performed with its attributes, and final decryption is performed using a key issued by TTP to acquire the data.

4.3 Proposed schemes

4.3.1 Setup Phase

Initially, the user registers by sending a registration request message to the TTP to register in the cloud. TTP registers user information in the user list, and generates PK and MK as follows.

< Setup: Generate PK, MK>

The prime order of the bilinear group G is p, and a random constructor is generated $\alpha, \beta \in Z_p$. The symbol i, is an integer representing the value according to the number of attributes and after calculating $h = g^s \in G_0$, the PK and MK are generated.

$$\text{PK} < g, \{g_i\}_{1 \le i \le n}, h, \text{e}(g,g)^{\alpha} > \quad (1)$$

$$\text{MK} < B, g^{\alpha} > \quad (2)$$

4.3.2 Key Generation Phase

The TTP generates a secret key through the user's attribute data, PK, and MK. Then, it sends ID_{User} , $nonce\ value\ and\ SK$ to the data user, the user list ($ID_{User\ n}$, $nonce\ n$) to the AC server and PK to the data owner, over a secure channel.

< $KeyGen(PK, MK, A_U)$: Generate SK corresponding to attribute set A_U >

· $j \in$ S (The symbol j indicates the number of each attribute, and the symbol S indicates a set of attributes.)

· $r_1 \ldots r_j \in Z_p$(The symbols r_1 to r_j denote random values given to each attribute.)

· Then, $D \in g^{\alpha+r}$ is calculated and an SK is generated as follows.

$$SK < D, \forall j \in S, D_j : g^r H(j)^{r_j}, D_j' : g^{r_j} > \quad (3)$$

4.3.3 Data Encryption Phase

The data owners generate access structures based on user attributes. Then the owner encrypts the data with the access structure and the PK received from the TTP, uploads it to the cloud and stores it. At this time, since the ciphertext has a constant size, the cloud storage space can be used efficiently.

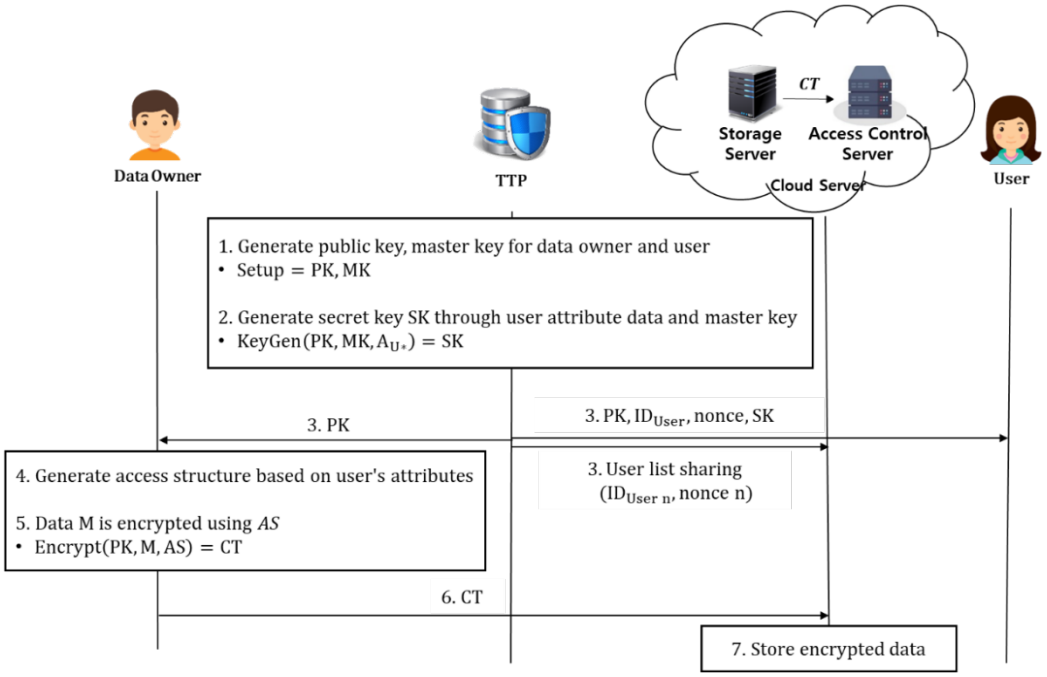

Fig. 8. Setup Phase, Key Generation Phase, Data Encryption Phase

Step 1. The data owners create AS with the attributes of users accessing the cloud. The message M, is encrypted with AS and PK to create a ciphertext, sent to the cloud storage and stored in storage.

< $\boldsymbol{Encrypt(PK, AS, M)}$**: Data encryption step with PK and AS>**

· $s \in Z_p$(Random value generation)

$$C_0: \mathrm{M} \cdot \mathrm{e}(g,g)^{\alpha s}, \qquad C_1: g^s, \qquad C_2: (\mathrm{h} \cdot \prod_{j \in AS} g_j)^S \tag{4}$$

$$\mathrm{CT} < \mathrm{AS}, C_0, C_1, C_2 > \tag{5}$$

4.3.4 User Data Access Phase

The user accesses the cloud storage by generating a token based on the information received from the TTP. On the AC server, the user's attributes are compared to those specified in the user list; then primary decryption is performed and the primary decrypted result and ciphertext are transmitted to the user. The user decrypts the ciphertext using the SK, thus acquiring message M.

Step 1. The user generates a TK by encrypting the attribute set using the nonce value received from the TTP. It then requests access to the cloud and sends the TK to the AC server with the user ID and timestamp.

$$\mathrm{TK} = k_{nonce}(A_{U*}) \tag{6}$$

4.3.5 Data Decryption Phase

In this phase, after user authentication, as a step of decrypting data, partial decryption is performed on the AC server, and the user then performs the final decryption to obtain the message.

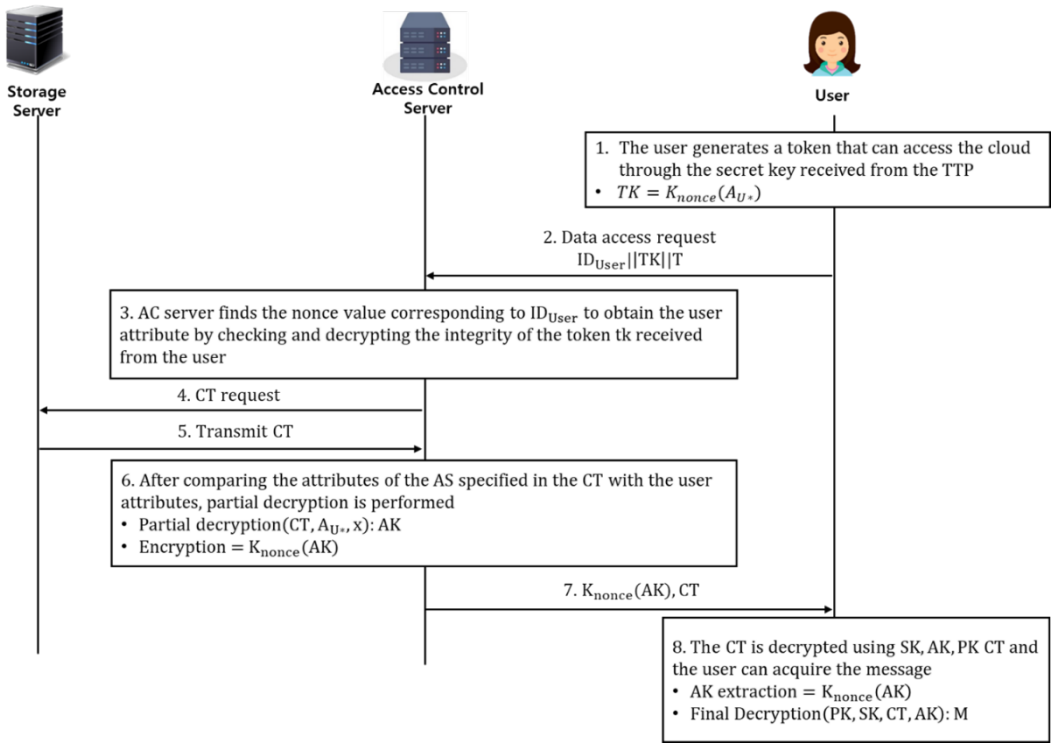

Fig. 9. User Data Access Phase, Data Decryption Phase

Step 1. The AC server decrypts the TK, obtains the user's attributes and compares the attributes of the AS contained in the ciphertext with the user's attribute set. Then, if it's a match, the AC server performs primary decrypting of the ciphertext and generates a key AK required for final decryption of the ciphertext. The AK is then encrypted with the user's nonce value and sent to the user along with the ciphertext CT.

< Partial decryption phase>

· Compare whether user attribute A_{U*} matches AS of CT → AK extraction

· z ∈ Z_p(Random value generation)

$$AK = \frac{\mathrm{e}(g^z, C_2)}{\mathrm{e}\left(C_1, (\prod_{j\in S} g_j)^s\right)} = \frac{\mathrm{e}\left(g^z, (g^s \cdot \prod_{j\in AS} g_j)\right)}{\mathrm{e}\left(g^s, (\prod_{j\in S} g_j)^s\right)} = \mathrm{e(g,g)}^{zs} \tag{7}$$

$$Ak_{nonce}(AK), \mathrm{CT} \tag{8}$$

A key required for final decryption and the ciphertext are transmitted if the user satisfied the attribute structure specification

Step 2. The user decodes $k_{nonce}(AK)$ received from AC. Then, the cipher text CT is decrypted using the SK, AK, and the PK, to acquire message M.

< Final decryption phase>

· AK Extraction = k_{nonce}(AK)

·$Decrypt(AK, PK, \mathrm{SK}, CT)$: Acquire M by decoding data CT

$$\mathrm{M} = \frac{C_0}{e(C_1, \mathrm{D}) \cdot ak^{r/z}} = \frac{\mathrm{M} \cdot \mathrm{e}(g,g)^{\alpha s}}{e(g^s, g^{\alpha+r}) \cdot \mathrm{e}(g,g)^{zs\cdot r/z}} = \frac{\mathrm{M} \cdot \mathrm{e}(g,g)^{\alpha s}}{(g,g)^{\alpha s+r} \cdot \mathrm{e}(g,g)^{s\cdot r}} = \frac{\mathrm{M} \cdot \mathrm{e}(g,g)^{\alpha s}}{e(g,g)^{\alpha s}} \tag{9}$$

4.3.6 Attribute Revocation Upon User Withdraws Phase

In this phase, we remove the attributes of a user who has left the system and block access to the stored data.

To remove a user who has left, TTP registers the user who has left as a attribute revocation list and changes the nonce value of the user. The user list is then synchronized and shared with the AC server. Therefore, the AC server cannot decrypt $k_{nonce}(A_{U*}, T)$ when the withdrawn user creates a TK with a nonce value and requests access to the cloud. The AC server then sends an access failure message to an unregistered (withdrawn) user to block access.

Step 1. The user sends a removal request from TTP for withdrawal.

Step 2. The TTP adds the user to be removed to the user attribute revocation list and changes their nonce value. Then, the user's nonce value in the registered user list is changed to reflect the user cancellation and this information is shared with the AC server.

Step 3. If the unsubscribed user uses their existing nonce value as a symmetric key to create a TK, and requests access to the AC server, the AC server finds the nonce value corresponding to the user's identifier. At this time, the AC server cannot decode the TK received from the user because the AC server key is not the same as that used by the user.

Step 4. The AC server then blocks access by sending the removed user an access failure message. Also, if a timestamp T is registered in the attribute revocation list, the user cannot access data stored during period T. This is because the nonce value of the user who registered the T period and the user nonce value of the ac server are different.

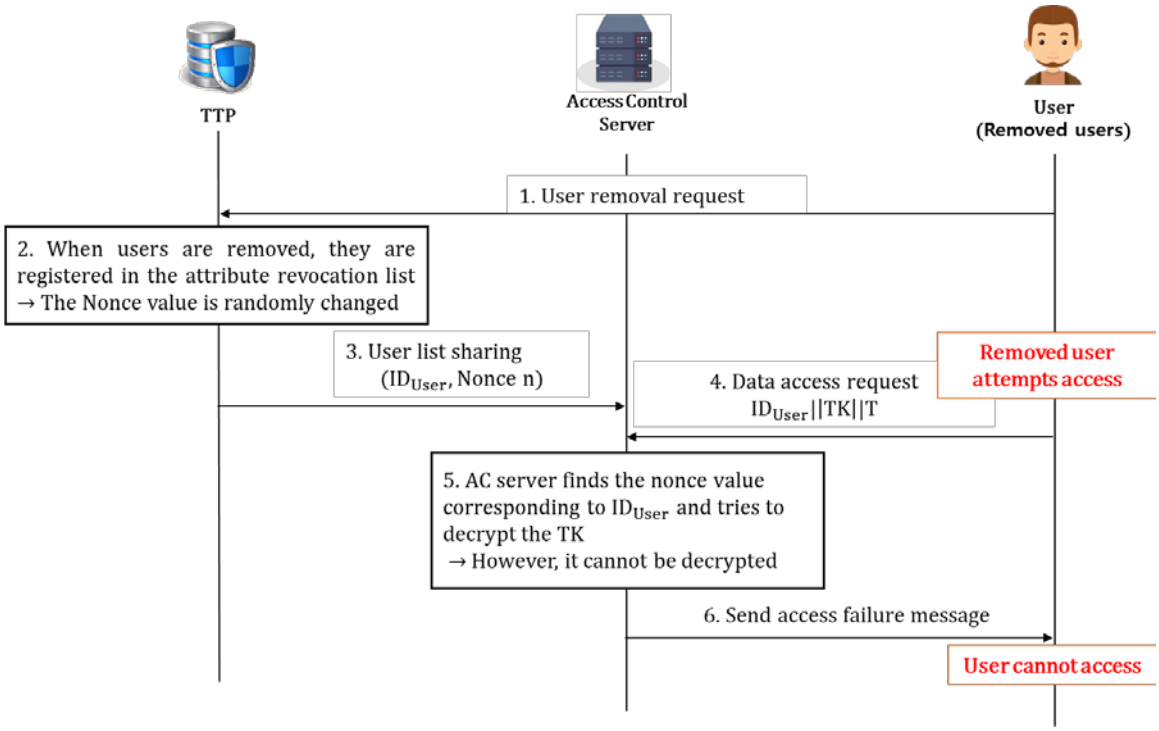

Fig. 10. Attribute Revocation Upon User Withdraws

5. ANALYSIS OF PROPOSED SCHEME

5.1 Security Analysis

The proposed approach prevents unauthorized users from accessing your data and is safe from security threats.

· **User collusion attack:** The proposed scheme can acquire the attributes of users who have permission to access the cloud through a collusion attack among ordinary users who do not have access to the cloud. After that, you can receive PK, $ID_{User}, nonce\ value\ and$ SK from TTP through the acquired attributes. However, the $ID_{User}, nonce\ value\ and$ SK received from TTP are different from existing users because SK generates based on the user's attributes and random nonce values when creating SK in TTP. Accordingly, when a token is created by using the $nonce\ value$ received by the general user as a symmetric key and requested to access the AC server, the AC server cannot find the $nonce\ value$ for the user identifier and cannot decrypt the token. In other words, it cannot be accessed other than the pre-registered user. There are also cases where the cloud may collaborate with users who do not have access to the cloud.The AC server sends the partial decryption text to the general user, and the general user can attempt to decrypt it. However, in this proposed scheme, SK is not known, so data cannot be decrypted. Therefore, the proposed scheme is safe for the collusion attack.

· **masquerade attack:** In the proposed scheme, encrypted data can is accessed and decrypted based on a user's attributes and SK. An attacker cannot decrypt the data even if they eavesdrop on the transmitted data. Also, the TTP registers users in the user attribute revocation list when they leave. This list is then shared with the AC server, which changes their $nonce\ value$. It also uses the user's registration values in the user list as symmetric keys to communicate between the AC server and the user. Thus, even if an unauthorized user uses another user's attributes to access the AC server, the AC server cannot decrypt the access request message. Therefore, the proposed scheme is safe from masquerade attacks, in which other users attributes are used to attempt to access data.

· **User attribute revocation:** In the proposed scheme, as in phase 4.3.6, the withdrawal user's $nonce\ value$ is changed by TTP when registering in the user attribute revocation list. Here, the $nonce\ value$ is a key value used when generating a TK for accessing the cloud. We then share the user revocation list with the AC server. When an access request is made using previous token, the AC server can not decode the token received from the user because the nonce value has changed. Thus, the access of the removed user can be blocked. And attributes for users registered in the attribute revocation list held by TTP are automatically revoked after a certain period of time.

· **Access control for unauthorized(withdrawn) users:** In the proposed scheme, only users registered in the TTP can generate valid tokens and access the AC server, which is connected to the cloud. The AC server compares the user attribute with the attribute in the access structure specified in the ciphertext to determine if it matches, and if it does, proceeds with partial decryption to extract the final decryption key AK. Thereafter, AK and CT are sent only to matching users. Therefore, users without access rights cannot obtain access rights. This ensures the confidentiality and integrity of data stored in the cloud.

Table 2. Security analysis of existing scheme and proposed scheme

Scheme Items	Hahn Scheme [13]	Wei Scheme [14]	Liu Scheme [11]	Zhao Scheme [12]	The Proposed Scheme
User Collusion	○	○	○	○	○
masquerade attack	○	○	○	○	○
Storage server space	Efficient	Efficient	Inefficient	Efficient	Efficient
User revocation	X	X	○	○	○
Attribute revocation	X	X	X	○	○
Key, Ciphertext updates	Requires key, ciphertext update		Requires key, ciphertext update	Re-encrypt the ciphertext	Not required
Ciphertext lenght	Constant-size ciphertext		Increases by the number of attributes	Constant-size ciphertext	
Encryption	$c_e + 3E + (n+1)M$	$6E + (n+2)M$	$c_e + 4(n+1)E$ $+2(n+1)M$	$2nc_e + 2(n+4)E$ $+4M + 2H$+1Enc	$c_e + 2E + nM$
Decryption (server)	$2c_e + 3E + 2nM$	-	-	$4c_e + (2n+2)M$ $+ 4E$	$2c_e + 2E + nM$
Decryption (user)	$c_e + 2E + M$	$4c_e + 2E + (n+3)M$	$4c_e + (n+2)E$ $+ (2n+5)M$	$4c_e + 2E$ $+(n+3)M$+1Dec	$c_e + E + M$
Outsourcing	Provide	Not provide	Not provide	Provide	Provide

c_e : *Pairing operation*; M : *Multiplication operation*; n: *Number of attributes*; E: *Expoentiation operation*

Enc: *Symmetric key encryption*; *Dec*: *Symmetric key decryption*; H: *Hash function*; ○: safe; X: unsafe

○: Safe (Provied); X: Unsafe(Not provied)

5.2 Efficiency

The proposed scheme increases the efficiency of cloud storage space by outputting ciphertext of a constant size, and the AC server increases the computational efficiency of the scheme for the user by processing part of the decryption. The computational power experiments in **Fig. 11**, **Fig. 12** were performed on a Windows system with a 3.50GHz Intel Core i5-4690 processor and 8GB of RAM. For the pairing operation, refer Pairing Based Crypto Library (Lynn, B., "The pairing-based cryptography (PBC) library," Available: http://crypto.stanford.edu/pbc, 2012). Also, multiplication and exponential operations are expressed as 0.001 because they are very fast. And In **Fig. 11**, **Fig. 12** the graphs for the experimental values are expressed on a millisecond basis.

· **Efficiency of cloud storage:** In the existing CP-ABE system, the ciphertext increases with the number of attributes specified in the access structure. In this proposed scheme, since the number of attributes is represented by one operation, such as a number of $C_2 = (\mathrm{h} \cdot \prod_{j \in AS} g_j)^s$, specified in the access structure, the size of the ciphertext is constantly output. In other words, the proposed scheme compared to the CP-ABE method, which does not output constant size of ciphertext in Table 1, enables efficient use of cloud storage.

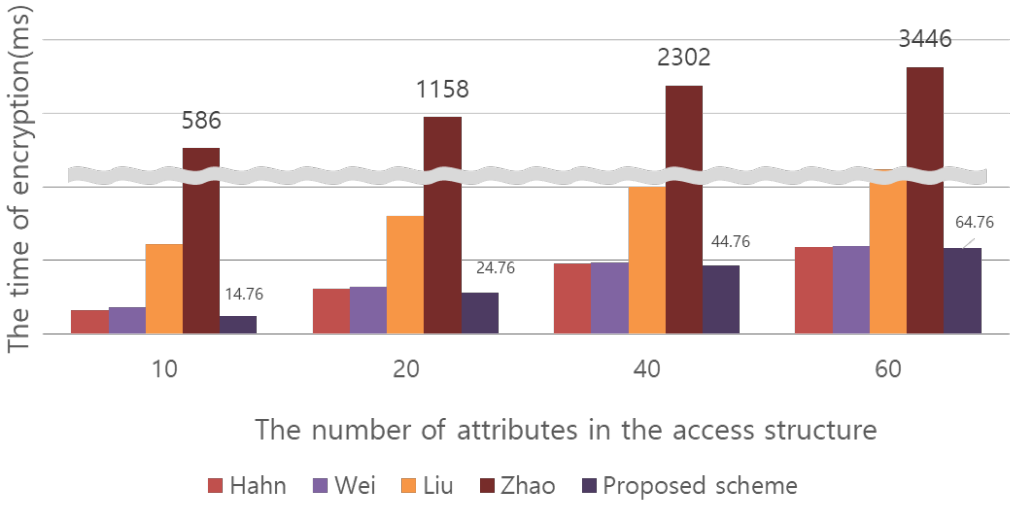

Fig. 11. Comparison of the encryption time

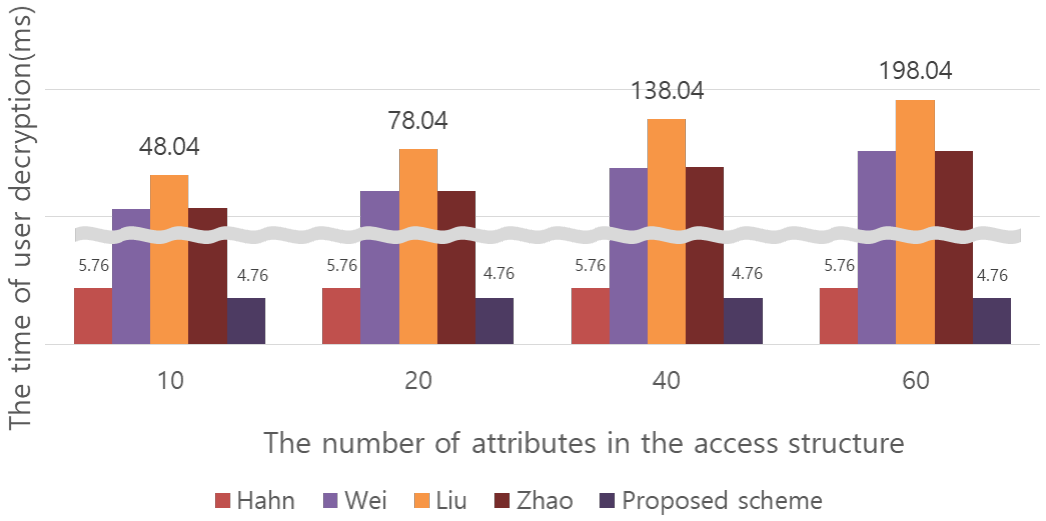

Fig. 12. Comparison of the user decryption time

· **User message encryption and decryption operation efficiency:** In the existing CP-ABE method, since the user receives and decrypts the ciphertext from the cloud server, the user's ciphertext decryption operation was burdensome. However, the proposed scheme supports outsourcing servers and performs part of the decryption operation to be processed by the user. **Fig. 11**, **Fig. 12** show a comparison of the speeds of the encryption and decryption calculations to the existing CP-ABE schemes. Referring to **Table 2** and **Fig. 11**, it can be seen that the proposed scheme is efficient compared to the existing CP-ABE method. Also, referring to the decoding portion of **Fig. 12**, the user's decoding operation amount is more efficient than the CP-ABE method in which outsourcing is supported is not supported. However, the Zhao scheme, which supports outsourcing, includes a verification process, so it performs more than twice the existing amount of encryption and decryption.

In addition, since the user downloads and decrypts the ciphertext in Wei schemes and the Liu scheme, the amount of computation required for the user to decrypt the ciphertext is a burden. On the other hand, the proposed scheme, like the Hahn scheme and Zhao scheme, performs some of the computations needed to decrypt ciphertexts a the trusted server, thereby reducing the computational amount for users to decrypt ciphertexts. This will be applicable to cloud computing environments used in mobile or Internet of Things (IoT) applications.

6. CONCLUSIONS

In this paper, we proposed a CP-ABE access control to block access of withdrawn users in dynamic cloud environments. The proposed scheme responds to security threats such as user collusion attacks, masquerade attacks, and backward security, When a user is withdrawn, it is registered in the attribute revocation list to block the access of the user who left the service, and the user's attribute is removed at a later period. When user removal is requested in the proposed scheme, the nonce value is changed in the user attribute list. If the user creates a token via the previous attribute information and nonce, and tries to request access to the cloud using it, the AC server blocks the user from access because it can not decode the token. This effectively solves the problem of inefficiently updating all keys and ciphertexts when removing attributes under existing access schemes. In addition, an outsourcing method is applied via the AC server that controls user access. The attributes of the access structure contained in the ciphertext are compared against the set of attributes of the accessing user. If they match, a partial decryption is performed and sent to the user, to perform the final decryption. It is efficient because the user performs only a part of the amount of computation required to decrypt the existing ciphertext. In addition, in terms of storage space efficiency, the storage efficiency is improved compared to the existing CP-ABE method because the ciphertext of a constant size is calculated. In terms of user operations, some of the computations for decrypting the ciphertext are processed by the AC using an outsourcing method, so that the efficiency can be improved for the user decrypt operation by reducing their computational load during decryption.

In future research, we will study anonymous CP-ABEs that can protect the access structure and user attributes because CP-ABE access schemes may lead to privacy infringements in environments where user attributes are sensitive.

Acknowledgement

This research was supported by the MSIT(Ministry of Science, ICT), Korea, under the High-Potential Individuals Global Training Program)(2020-0-01596) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation) and was supported by the Soonchunhyang University Research Fund.

References

[1] Bethencourt. J, Sahai. A, Waters. B, “Ciphertext-policy attribute-based encryption,” in *Proc. of Security and Privacy, SP'07. IEEE Symposium on*, pp. 321-324, 2007. Article(CrossRefLink)

[2] Goyal, V., Pandey, O., Sahai, A., & Waters, B., “Attribute-based encryption for fine-grained access control of encrypted data,” in *Proc. of the 13th ACM conference on Computer and communications security, ACM*, pp. 89-98, 2006. Article(CrossRefLink)

[3] Cheung Ling, Nwport Calvin, "Provably secure ciphertext policy ABE," in *Proc. of the 14th ACM conference on Computer and communications security, ACM*, pp. 456-465, 2007. Article(CrossRefLink)
[4] Sekhar. B. R, Kumar. B. S, Reddy. L. S, PoornaChandar. V, "CP-ABE based encryption for secured cloud storage access," *International Journal of Scientific & Engineering Research*, vol. 3, no. 9, pp. 1-5, 2012. Article(CrossRefLink)
[5] Zhu. S, Yang. X, "Protecting data in cloud environment with attribute-based encryption," *International Journal of Grid and Utility Computing*, vol. 6, no. 2, pp. 91-97, 2015. Article(CrossRefLink)
[6] Xu. Z, Martin. K. M, "Dynamic user revocation and key refreshing for attribute-based encryption in cloud storage," in *Proc. of Trust, Security and Privacy in Computing and Communications, 2012 IEEE 11th International Conference on, IEEE*, pp. 844-849, 2012. Article(CrossRefLink)
[7] Yang. K, Jia. X, "Attribute-based Fine-Grained Access Control with Efficient Revocation in Cloud Storage Systems," in *Proc. of the 8th ACM SIGSAC symposium on Information, computer and communications security, ACM*, pp. 523-528, 2013. Article(CrossRefLink)
[8] Ramesh, D, Priya. R, "Multi-authority scheme based CP-ABE with attribute revocation for cloud data storage," in *Proc. of Microelectronics, Computing and Communications (MicroCom), 2016 International Conference on, IEEE*, pp. 1-4, 2016. Article(CrossRefLink)
[9] Xia, Zhihua, Liangao Zhang, and Dandan Liu, "Attribute-based access control scheme with efficient revocation in cloud computing," *China Communications*, vol. 13, no. 7, pp.92-99, 2016. Article(CrossRefLink)
[10] Liu, Zechao, et al., "Practical attribute-based encryption: Outsourcing decryption, attribute revocation and policy updating," *Journal of Network and Computer Applications*, vol.108, pp.112-123, 2018. Article(CrossRefLink)
[11] Liu, Joseph K., et al., "Time-based direct revocable ciphertext-policy attribute-based encryption with short revocation list," in *Proc. of International Conference on Applied Cryptography and Network Security. Springer, Cham*, pp. 516-534, 2018. Article(CrossRefLink)
[12] Zhao, Yang, et al., "A revocable storage CP-ABE scheme with constant ciphertext length in cloud storage," *Mathematical biosciences and engineering: MBE*, vol. 16(5), pp. 4229-4249, 2019. Article(CrossRefLink)
[13] Hahn Changhee, Junbeom Hur, "Constant-Size Ciphertext-Policy Attribute-Based Data Access and Outsourceable Decryption Scheme," *Journal of KIISE*, vol. 43, no. 8, pp. 933-945, 2016. Article(CrossRefLink)
[14] Teng, Wei, et al., "Attribute-based access control with constant-size ciphertext in cloud computing," *IEEE Transactions on Cloud Computing*, vol. 5, no. 4, pp. 617-627, 2017. Article(CrossRefLink)
[15] A. Sahai and B. Waters, "Fuzzy Identity Based Encryption," *Advances in Cryptology, Eurocrypt, volume 3494 of LNCS, Springer*, pp 457–473, 2005. Article(CrossRefLink)
[16] Lee C.C., Chung P.S., Hwang M.S, "A Survey on Attribute-based Encryption Schemes of Access Control in Cloud Environments," *IJ Netw. Secur.*, vol. 15, pp. 231-240, 2013. Article(CrossRefLink)
[17] Kumar V., Kumar P.V, "A lterature Survey on Revocable Multiauthority Cipher Text-Policy Attribute-Based Encryption (CP-ABE) Scheme for Cloud Storage," *Int. J. Adv. Res. Electron. Commun. Eng.*, vol. 3 pp. 1723-1728, 2014. Article(CrossRefLink)
[18] X. Liang, Z. Cao, H. Lin, J. Shao, "Attribute-based proxy re-encryption with delegating capabilities," in *Proc. of the 4th International Symposium on Information, Computer, and Communications Security*, pp. 276-286, 2009. Article(CrossRefLink)
[19] H. Seo, 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. Article(CrossRefLink)
[20] Liu C.W., Hsien W.F., Yang C.C., Hwang M.S, "A Survey of Attribute-based Access Control with User Revocation in Cloud Data Storage," *Int. J. Netw. Secur.*, vol. 18, pp. 900–916, 2016. Article(CrossRefLink)

[21] Zhou, Z. & Huang, D, "On efficient ciphertext-policy attribute based encryption and broadcast encryption," in *Proc. of the 17th ACM conference on Computer and communications security*, pp. 753-755, 2010. Article(CrossRefLink)
[22] R. Canetti, S. Halevi, and J. Katz, "A Forward-Secure Public-Key Encryption Scheme," *Advances in Cryptology, Eurocrypt, volume 2656 of LNCS. Springer*, pp. 255-271, 2003. Article(CrossRefLink)
[23] R. Canetti, S. Halevi, and J. Katz, "Chosen Ciphertext Security from Identity Based Encryption," *Advances in Cryptology, Eurocrypt, volume 3027 of LNCS, Springer*, pp. 207–222, 2004. Article(CrossRefLink)
[24] D. Boneh and X. Boyen, "Efficient Selective-ID Secure Identity Based Encryption Without Random Oracles," *Advances in Cryptology, Eurocrypt, volume 3027 of LNCS, Springer*, pp. 223–238, 2004. Article(CrossRefLink)
[25] D. Boneh and M. Franklin, "Identity Based Encryption from the Weil Pairing," *Advances in Cryptology, CRYPTO, volume 2139 of LNCS, Springer*, pp. 213–229, 2001. Article(CrossRefLink)
[26] D. Boneh and X. Boyen, "Efficient Selective-ID Secure Identity Based Encryption Without Random Oracles," *Advances in Cryptology – Eurocrypt, volume 3027 of LNCS, Springer*, pp. 223–238, 2004. Article(CrossRefLink)
[27] Chung. P. S, Liu. C. W, Hwang. M. S, "A Study of Attribute-based Proxy Re-encryption Scheme in Cloud Environments," *IJ Network Security*, vol. 16, no. 1, pp. 1-13, 2014. Article(CrossRefLink)

Yong-Woon Hwang received the M.S.degrees in Depart of Computer Science Engineering from Soonchunhyang University, Korea, in 2017, respectively. He is now a Ph.D. candidate in Department of Computer Science and Engineering from Soonchunhyang University, Korea. His research interests include Cloud Storage Security, Cryptography,Attribute-based Encryption, Data Sharing, etc.

Im-Yeong Lee is corresponding author. He received the B.S. degrees in Department of Electronic Engineering from Hongik University, Korea, in 1981 and the M.S. and Ph.D. degrees in Department of Communication Engineering from Osaka University, Japan, in 1986 and 1989, respectively. From 1989 to 1994, he had been a senior researcher at ETRI (Electronics and Telecommunications Research Institute), Korea. Now he is a professor in Department of Computer Software Engineering from Soonchunhyang University, Korea. His research interests include Cryptography, Information theory, Computer & Network security.