A Study on Data Sharing Scheme using ECP-ABSC that Provides Data User Traceability in the Cloud

Yong-Woon Hwang[1], Taehoon Kim[2], Daehee Seo[3] and Im-Yeong Lee[4*]
[1,2,4] Department of Software Convergence, Soonchunhyang University
Asan 31538, South Korea
[e-mail: hyw0123@sch.ac.kr, 20134101@sch.ac.kr, imylee@sch.ac.kr]
[3] Department of Faculty of Artificial Intelligence and Data Engineering, Sangmyung University
Seoul 03016, South Korea
[e-mail: daehseo@smu.ac.kr]
[*]Corresponding author: Im-Yeong Lee

Received October 4, 2022; accepted November 28, 2022;
published December 31, 2022

Abstract

Recently, various security threats such as data leakage and data forgery have been possible in the communication and storage of data shared in the cloud environment. This paper conducted a study on the CP-ABSC scheme to solve these security threats. In the existing CP-ABSC scheme, if the data is obtained by the unsigncryption of the data user incorrectly, the identity of the data owner who uploaded the ciphertext cannot be known. Also, when verifying the leaked secret key, the identity information of the data user who leaked the secret key cannot be known. In terms of efficiency, the number of attributes can affect the ciphertext. In addition, a large amount of computation is required for the user to unsigncrypt the ciphertext. In this paper, we propose ECP-ABSC that provides data user traceability, and use it in a cloud environment to provide an efficient and secure data sharing scheme. The proposed ECP-ABSC scheme can trace and verify the identity of the data owner who uploaded the ciphertext incorrectly and the data user who leaked the secret key for the first time. In addition, the ciphertext of a constant size is output and the efficiency of the user's unsigncryption computation were improved.

Keywords: Cloud Access Control; Data Sharing; Attribute-based Signcryption; Traceability; Constant-Size Ciphertext; Outsourcing;

1. Introduction

It has recently become possible to securely store and share data between users in the cloud. Because it can manage and share a lot of big data used in hospitals, companies, and public institutions, the development of cloud computing technology gives users many advantages. However, some security threats may occur when data is shared in a cloud environment [1-2]. First, providers of cloud services cannot be fully trusted. If you use an externally provided cloud, you can securely protect your stored data from external threats. The stored data can be secure from external threats if an externally provided cloud is used. However, the provider knows the content of the data stored and used. Shared data can be leaked or deleted by an attacker (a malicious user). The attacker accesses the server, leak the data to the outside and tamper with the stored data. If it is a cloud server that manages and stores users' personal (sensitive) information, it becomes a serious security threat [3-4]. Accordingly, cloud data encryption is required, along with data access control.

Attribute-based encryption allows for both data access control and encryption/decryption. This access and encryption are particularly applicable to the cloud environment [5-6]. However, attribute-based encryption does not allow the data user to trust the data obtained by the decrypted ciphertext. That is, it is impossible to verify data reliability. Questions arise such as who uploaded the data? Are the decrypted data those uploaded by the owner? And has the data been tampered with? To solve this problem, many scholars have researched attribute-based signcryption scheme [7-9]. Attribute-based signcryption is a security technology that combines the characteristics of attribute-based signature and attribute-based encryption. In Chapter 2, attribute-based signcryption was explained in detail. Our scheme is a variation of ciphertext-policy attribute-based signcryption (CP-ABSC).

Several CP-ABSC schemes have been researched, but the security threat still remains and or is inefficient in terms of the amount of computation. First, the data user must ask the owner of the uploaded data to take responsibility if the decrypted ciphertext is unclear. However, it is possible to know what attributes of the data owner, but not the identity of the data owner [10-11]. In addition, an external (unregistered) user can access the cloud server using a registered user’s secret key (i.e., Leaked unsigncryption key). If data leakage is detected, the user's identity information is checked by verifying the user's secret key. However, the secret key contains only the user attribute value of the initially issued data, and the identity of the data user is unknown [12-13]. The above problem occurs because the anonymity between each object is provided by using the CP-ABSC scheme. Therefore, while providing anonymity, it should be possible to verify the identity of data users (owners/users) in the event of a problem. Here, the problem means that the acquired data is wrong or the leaked secret key is first distributed. Second, a solution is needed because the number of attributes included in the ciphertext affects the size of the ciphertext by the data owner during the encryption phase [14-15]. Finally, since the amount of operation required for a user to unsigncryption the ciphertext is large. Therefore, it is limited for users who are burdened with device performance among users who try to access through a device in an IoT-Cloud or a mobile computing environment [16-17].

In this paper, we propose the Efficient Ciphertext-Policy Attribute-based Signcryption (ECP-ABSC) scheme that provides data user traceability, and use it to provide efficient and secure data sharing in the cloud. Specifically, the proposed ECP-ABSC scheme possible to verify the identity information of a data owner or user via cooperation between a Trace Authority (TA) and Attribute Authority (AA) when a leaked secret key or shared data is unclear. In addition, the number of attributes does not affect the size of the ciphertext, and is

output in a constant size. Finally, the cloud server supports some amounts of unsigncryption computation to improve the computation efficiency when the user unsigncryption the ciphertext.

1.1 Contribution

The contributions of the proposed ECP-ABSC are as follows.

- **Data user traceability and identification:** In the proposed scheme, a trusted entity called TA is used. The TA does not participate in data-sharing; it only registers the identities of data users and issues pseudonyms. In detail, anonymity is provided because data users perform data sharing by registering an identity from the TA and issuing a pseudonym. If the data obtained from the ciphertext is unclear, or when the identity of the data user who leaked the secret key is to be verified, the TA and the AA can cooperate to check the identity of the requested data user. Compared to the traditional CP-ABSC scheme, this provides anonymity to data users and can trace and identify data users when a problem arises.
- **Ciphertext output of a constant size:** In the proposed scheme aggregates the attribute values included in the ciphertext, and calculates them as a single value. Then it is included in the ciphertext to provide a ciphertext output of a constant size. Compared to the existing scheme, the number of attributes contained in a ciphertext does not affect the size of the ciphertext.
- **Efficient user unsigncryption operation:** In the proposed scheme, performs a partial unsigncryption phase that can compare and calculate ciphertext and user attributes in the cloud server. Then, the user receives the unsigncrypted ciphertext and performs the final unsigncryption to obtain data. This process can reduce the user's ciphertext unsigncryption operation amount compared to the existing scheme for the user to unsigncrypt the entire ciphertext. As a result, even a user with a performance burden can sufficiently perform ciphertext unsigncryption through the outsourced server.

1.2 Organizaion

Each chapter of the paper is as follows. Chapter 2 is the background and describes attribute-based signcryption, the existing CP-ABSC scheme, and related works. Chapter 3 describes the requirements(security, efficiency) of the cloud, and Chapter 4 describes the proposed ECP-ABSC scheme. Chapter 5 focuses on the efficiency and security aspects of our scheme. Chapter 6 presents our conclusion.

2. Backgrounds

This chapter describes attribute-based signcryption and the existing CP-ABSC scheme and related works.

2.1 Preliminaries

2.1.1 Bilinear Map

Recently, bilinear mapping is used as a cryptography tool for information security. The bilinear pairing function is also called bilinear mapping. Below is a description of the notation:

Assume that there are multiplicative group $G_1$and G_2 with the same order p. Suppose a discrete log problem is difficult to solve within a group. Let g be a generator group of G_1, and

let $e: G_1 X G_1 \rightarrow G_2$ be a bilinear mapping that satisfies the following properties:
1. Bilinearity: For all $P, Q \in G_1$ and all $a, b \in \mathrm{Z}_p$, $e(P^a, Q^b) = e(P, Q)^{ab}$.
2. Non-degeneracy: For all $Q \in G_1$, if $e(P, Q) = 1$, then $P = 0$.
3. Computability: There is an algorithm that computes $e(P, Q) \in G_2$ for all $P, Q \in G_1$,.

2.1.2 Bilinear Diffie Hellman (BDH) Assumption

The deterministic BDH assumption is that two pairs $(g^a, g^b, g^c, \mathrm{W} = \mathrm{e}(g, g)^z)$ and $(g^a, g^b, g^c, \mathrm{T} = \mathrm{e}(g, g)^{abc})$, have no algorithm A to distinguish two pairs with meaningful probability. Here it is $a, b, c, z \in Z_p$. If algorithm A to solve the deterministic BDH assumption that if $|\Pr[\mathrm{A}(g^a, g^b, g^c, \mathrm{T}) = 1] - \Pr[\mathrm{A}(g^a, g^b, g^c, \mathrm{W}) = 1]| \geq \epsilon$ is satisfied, then algorithm A has a profit of ϵ [18].

2.1.3 Bilinear Diffie Hellman Exponent (BDHE) Assumption

The deterministic BDHE assumption is 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 can compute $\mathrm{T} = \mathrm{e}(\mathrm{h}, g)^{\alpha\beta+1}$ with meaningful probability. Here is $h, g \in G_1$. As defined by $g_i = g^{\alpha^i}(\mathrm{i} = 1, \ldots, 2\mathrm{B})$ and $g_{\alpha,\beta} = (g_1, \ldots, g_B, g_{B+2}, \ldots, g_{2B})$, when the next two pairs are $(h, g, g_{\alpha,\beta}, W = \mathrm{e}(\mathrm{h}, g)^z)$, $(h, g, g_{\alpha,\beta}, \mathrm{T} = \mathrm{e}(\mathrm{h}, g)^{\alpha\beta+1})$ the algorithm A to solve the deterministic BDHE assumption that if $|\Pr[\mathrm{A}(h, g, g_{\alpha,\beta}, \mathrm{T}) = 1] - \Pr[\mathrm{A}(h, g, g_{\alpha,\beta}, \mathrm{W}) = 1]| \geq \epsilon$ is satisfied, then algorithm A has a profit of ϵ [18].

2.1.4 Elliptic Curve Discrete Logarithm Problem (ECDLP) Assumption

Elliptic curve cryptography is a public key encryption method that uses the mathematical properties (discrete logarithm problem for elliptic curve are difficult) of elliptic curves in a finite field. To use elliptic curve cryptography, an elliptic curve is a set of solutions *(X, Y)* to the equation $y^2 = x^3 + ax + b\ (mod\ p)$ defined for arbitrary integers a, b. The fact that there is a point *P = (X, Y)* on the elliptic curve means that the above equation is satisfied. $Q = x \cdot P$ can be expressed as a definition of an arbitrary integer *x* for two points *P* and *Q*. The solution to find x is a problem with discrete logarithm elliptic curves. That is, assuming that $Q = x \cdot P$, it is easy to ind *Q* using $x \cdot P$. However, it is very difficult to infer the value of *x* even if *Q* and *P* are known [18].

2.2 Attribute Based Signcryption

Signcryption is a cryptographic signature/encryption tool that guarantees confidentiality, integrity, and non-repudiation. In particular, it has the advantage of effectively reducing operating costs and communication overhead compared to the existing encryption method after signing. In 1997, the first signcryption was proposed by Zheng et al., followed by various attribute-based technologies [19]. ABSC performs signcryption/unsigncryption based on a set of attributes (affiliation, job, etc.) of each participant and the access structure created based on it [7-9]. That is, it has the characteristics of ABE and ABS [5-6][20].

The ABSC consists of CP-ABSC and KP-ABSC and this proposed scheme conducted research on CP-ABSC. In the CP-ABSC scheme, the ciphertext contains an access structure created by collecting user attributes. A user with the attributes specified in the access structure can access and decrypt the ciphertext. For example, suppose the access structure is {{Nurse AND Doctor} OR Hospital A}, and is included in the ciphertext. Only nurse and doctor from hospital A can access the ciphertext and decrypt.

The CP-ABSC is widely used in a data-sharing environment between users in a public cloud. In particular, it is widely used in 1:N (where N indicates “many”) cloud environments because it has the attributes of sharing data by accessing the ciphertext uploaded by the data owner.

2.2.1 Access Structure

ABE performs encryption/decryption with an access structure created with a set of attributes (e.g., job, affiliation) for each entity. The ABSC is also a scheme of performing signcryption/unsigncryption using an access structure. In detail, two access structures are used in ABSC. The access structure γ_e is similar to the access structure used in ABE. Another access structure γ_s, creates the ciphertext from the attributes of the data owner. In the access structure, denoted by T, each non-leaf node can represent a threshold gate such as an AND gate or an OR gate depending on the threshold. In general, for all nodes $x \in$ T, we use the notations num_x and k_x to represent the threshold of x and the number of children. For a non-leaf node x, if $k_x = num_x$, then x represents an AND gate. If $k_x = 1$, it represents an OR gate. If $1 < k_x < num_x$, then x is a threshold gate. We define $k_x = 1$ and $num_x = 0$ for leaf node x.

2.2.2 CP-ABSC Scheme

Fig. 1 shows the basic structure of CP-ABSC. A CP-ABSC scheme has an AA, cloud server, and data owners and users (see Chapter 4). The overall scenario of CP-ABSC consists of a total of four phases: setup, key generation, data signcryption/unsigncryption.

First, AA creates public parameters and master keys through the setup phase. Second, when a data user(owner/user) requests a secret key (signature secret key, ciphertext unsigncryption key) from AA, the AA generates a signature secret key/ciphertext unsigncryption key. In detail, the AA generates a signature secret key is based on the data owner's attributes and sends it to the data owner (with the *PPs*). It then generates a ciphertext unsigncryption key with the attributes s of the users, which is sent to the user (with the *PPs*). Third, the data owner generates access structure (γ_S, γ_e). γ_S is an access structure that can represent the data owner, and is generated with the data owner's attributes. γ_e is generated with the attributes of the data users who needs to access the ciphertext. Then, the owner signcrypt the using the access structures, *PPs*, and signature secret key. Then sends the ciphertext to the cloud server. Finally, the data user requests a ciphertext from the cloud server. And it performs unsigncryption by comparing its own attribute with the attribute value of the access structure in the ciphertext. This requires the user's unsigncryption key, the ciphertext, and *PPs*. User with the required ciphertext attributes can

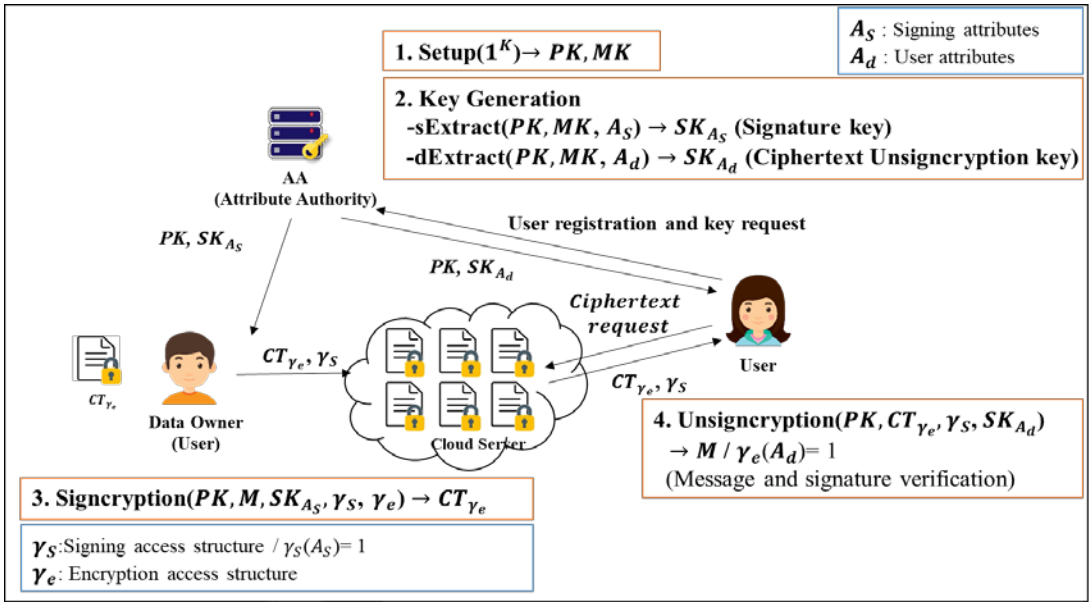

Fig. 1. CP-ABSC scheme structure

unsigncrypt. Through the data verification process, the attribute value of the data owner and the integrity of the acquired data can be verified.

2.3 Related Work

2.3.1 The need for user tracing and identification in the CP-ABSC scheme

The existing CP-ABSC scheme communicates shared data by signcrypting and unsigncrypting user attributes. Accordingly, anonymity is provided between entities that want to participate in data sharing. This can protect privacy between users. However, this several problems occur as follows.

First, the shared ciphertext includes the attributes of the data owner. Assumed that the user acquires data after performing unsigncryption. If the data are unclear, the identity of the owner who uploaded them cannot be verified because the owner information in the ciphertext is exclusively attributes [10].

For example (**Fig. 2**), assume that Teams A and C of a company collaborate on a project. Team A encrypts and uploads data ("Project version 5") with user attributes they want to share with team C. Data with Team A attributes are shared by insiders or other data ("Project version 3") can be modulated by the addition of Team C attributes and then re-uploaded (re-encrypted). However, Team C cannot know whether the data obtained ("Project Version 3") are inaccurate, or, if they recognize that the data are inaccurate, they cannot know the identity of the owner who uploaded them [10-11].

Another problem arises (**Fig. 2**) when a leaked secret key is detected. It is assumed that one of the users in Team C leaks secret key (i.e., gives the secret key that accesses the cloud to a third party for money). The third party can obtain data by accessing the cloud with the key and attempting to decrypt the ciphertext. At this time, when the cloud server detects the risk of data leakage and requests an investigation from a trusted authority, the trusted authority verifies the secret key of the user who can access to the data. However, the secret key lacks user identity information and it is impossible to trace the leaked user [12-13].

In the existing ABE scheme, research to find a user who has leaked an abused secret key is continuously being conducted. When a legitimate user privately provides a secret key to another user, the user who has been provided with the secret key can access the cloud and decrypt. The

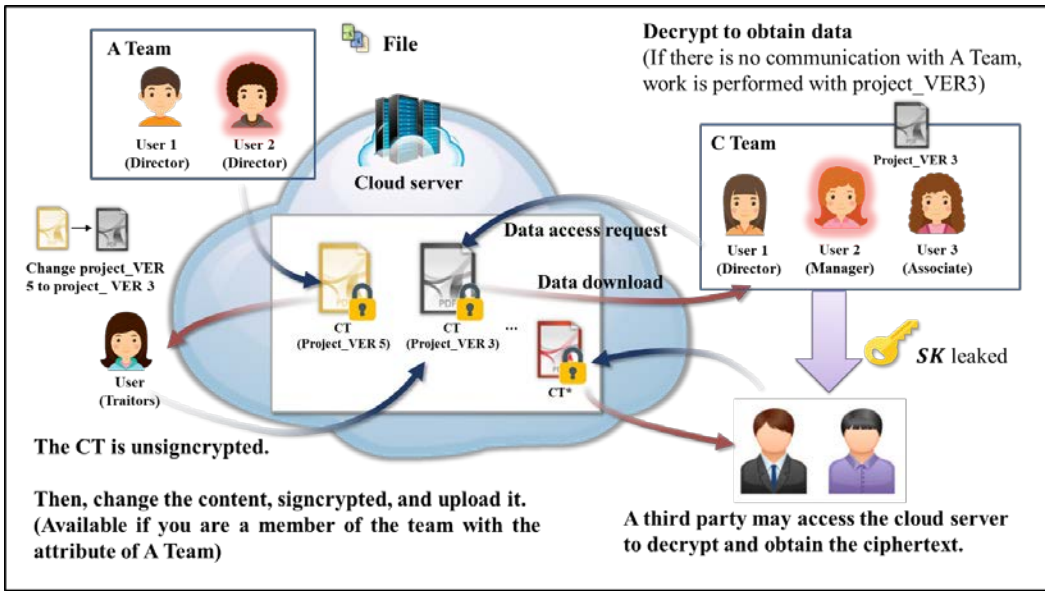

Fig. 2. Requires data user tracing and identity verification

secret key provided to other users is referred to as an abused secret key. For example, there are cases in which an authorized user publicly leaks or sells an attribute secret key through an e-commerce platform. At this time, after obtaining an abused key from KGC or a trusted authority, the key is verified through the key sanity check process to verify the user's identity [22-24].

The above two problems can occur sufficiently in the CP-ABSC environment. Therefore, depending on the circumstances, it should be possible to track and verify the identity of the data user.

2.3.2 The need for efficiency in the CP-ABSC scheme

In the traditional CP-ABSC schemes, the attributes included in the ciphertext affect the size of the ciphertext. Since ciphertext is communicated, stored, shared, and processed, the ciphertext size has a significant impact. To solve this problem, the attributes to be included in the ciphertext are collected and aggregated into a single value. In this case, ciphertext may be output in a constant size. However, it only affects the size of the ciphertext and does not affect the amount of computation required for signcryption/unsigncryption. Rather, since the aggregate operation is performed, the amount of operation is added compared to the CP-ABSC scheme that does not provide a ciphertext of a constant size. To efficiently process the amount of computation, it is necessary to support the amount of computation of the server such as outsourcing.

In the existing CP-ABSC schemes, the user receives the requested ciphertext from the cloud server and performs unsigncryption. However, if the user performing unsigncryption with a device or terminal may have limited performance, there are cases where the user cannot properly unsigncrypt the ciphertext. To solve this problem, research should be conducted to utilize an outsourced server to provide a part of the computational amount of the user's ciphertext unsigncryption.

2.3.3 Previously researched CP-ABSC scheme

The first signcryption was proposed by Zheng and based on this scheme, attribute-based signcryption was then proposed by Gangé et al. in 2010 [7]. Since then, the attribute-based signcryption that provides various requirements has been continuously researched based on Gangé et al.

The schemes of Deng et al. and Sana et al. are the CP-ABSC schemes that provide the requirements for outputting a ciphertext of a constant size [14-15]. Both schemes aggregate the attributes into a single value. This is one of the important requirements to provide because the size of the shared ciphertext affects communication and storage.

The scheme of Sana et al., Hundera et al., and Deng et al., are the CP-ABSC schemes that provides requirements for outsourcing server support to improve computational efficiency [16-17, 22-23]. In terms of data owners and users, signcryption and unsigncryption processes require a lot of operation. Therefore, operational support from a reliable outsourced server is required to handle this efficiently. In particular, partial unsigncryption by the server reduces the user burden.

The schemes of Lu et al. and Wei et al. are the CP-ABSC schemes that provides a requirement for identify by tracing the data owner [10-11]. Thus, as mentioned above, if the data obtained are incorrect, the identity of the owner can be verified. Identities are initially registered with the AA. If a problem occurs, the user requests the owner’s identity from the AA. However, the default anonymity of traditional CP-ABSC is then lost. Therefore, it must be able to work with a separate trusted server such as a TA to verify the identity of the owner.

The schemes of Rohit et al. and Hong et al. are the CP-ABSC schemes that provide traceability requirements that enable identity verification by tracing the data user who was issued the secret key when a leaked secret key is detected [12-13]. All data user identities are initially registered with the AA. If a leaked secret key is detected, the identity information can be checked by tracing

the data user who first issued the key can be verified through AA. However, as mentioned above, anonymity between AA and data users is not provided.

Our ECP-ABSC scheme was proposed as a model that satisfies the three requirements by analyzing the above-mentioned CP-ABSC schemes. The requirements provided include data user tracing and identity verification, ciphertext of a constant size, partial unsigncyrption performed on a cloud server

3. Security Requirements

An CP-ABSC scheme shares signcrypted data using only the attributes of the data owner and users. Therefore, anonymity basically provided for the data user and owner. In the secure CP-ABSC scheme provides confidentiality and integrity for shared data. The user access control function for accessing the ciphertext is also a basic requirement. In addition, in order to build a secure data sharing system using CP-ABSC, various requirements should be considered. The requirements are efficiency in CP-ABSC scheme, data owner and user identity trace, attribute revocation of withdrawn users, multi-AA for key escrow problem solving, and ensures that user attribute management is secure etc. The CP-ABSC scheme, which provides all requirements, is the most secure and efficient scheme. However, as the requirements apply, the CP-ABSC system becomes heavy (inefficient). Therefore, research that can apply the necessary requirements according to the environment is required.

Our proposed ECP-ABSC scheme is to focus on the requirements for tracing data users and verifying the identity, and for outputting a constant size of ciphertext and performing partial unsigncryption to provide efficiency. The description of the requirements to be provided are as follows.

- **Data user traceability and identification:** In the CP-ABSC scheme, if the data obtained during unsigncryption are incorrect, the identity of the owner who uploaded the ciphertext cannot be known. If a problem arises, it is essential to identify the data owner who uploaded the ciphertext using the AA or a cloud server, and takes appropriate action. In traditional CP-ABSC schemes, the secret key issued by the data user does not contain a value that can identify the data user. Therefore, such a user may leak the key and attribute values to others for profit or out of malice. That third party can access the cloud, obtain ciphertext, and perform unsigncryption. This can cause data leakage. Therefore, it is necessary to verify the identity information of the user who was issued the secret key for the first time. This means that the data user cannot provide the secret key to other users in advance [25].
- **Output constant-sized ciphertext:** In some CP-ABSC schemes, the size of the ciphertext is affected by the number of attributes in the ciphertext. If the size of the ciphertext increases, it affects the cloud storage where the ciphertext is stored. To solve this problem, research that can output the ciphertext of a constant size without affecting the number of attributes is required.
- **Efficient user unsigncryption operation:** In general, partial unsigncryption is performed by comparing attribute values when decrypting ciphertext, and final unsigncryption is performed through a secret key to obtain data. However, the amount of operation required when the user decrypts the ciphertext makes users who lack computing power feel a burden. To solve this problem, research should be conducted to utilize an outsourced server that can handle part of the operational amount of ciphertext (partial unsigncryption) [26-27].

- **Security (confidentiality, integrity) and access control for shared data:** If the data shared and stored in the cloud is plaintext, attackers can create various security threats such as data leakage and forgery with shared data as targets. Therefore, it is important to provide security and access control for data. Data must be encrypted and shared, and confidentiality must be provided so that only legitimate users with authority can decrypt it. By verifying the unsigncrypted data, it is necessary to verify the integrity of the data uploaded by the data owner. In order for the user to access the stored ciphertext, an element for access control is required so that only the user with the attributes specified in the access structure by the data owner can access.

4. The Proposed Scheme

In this chapter, we propose the ECP-ABSC scheme that provides data user traceability, and use it to provide efficient and secure data sharing in a cloud environment. The feature of the proposal scheme is to provide anonymity between each participating entity. If a problem arises (If there is a problem with the data obtained by the data user decrypting the signature, or if it is necessary to verify the identity of the first leaker of the leaked user's secret key), the TA and AA cooperate to identify the owner or user involved. In addition, the size of the ciphertext proportional to the number of existing attributes was minimized by outputting the ciphertext size as a constant. Finally, the cloud server partially unsigncrypts the ciphertext requested by the user. As a result, the user's ciphertext unsigncryption amount is reduced, thereby increasing the user's operation amount efficiency.

Fig. 3 shows the overall scenario of the proposed scheme. Participants consist of the AA, TA, cloud server, data users (owners and users). The detailed descriptions follow.

4.1 System Model

4.1.1 System Entity

The description of the role of each entity in the ECP-ABSC scheme are as follows.

- **Attribute Authority:** In this proposed scheme, AA is a semi-trusted server that manages the attributes of users. Here, the expression “semi-trust” refers to an entity that can be trusted by users but, since service providers are curious subjects, they have the right to obtain user information whenever they want. Accordingly, when requesting a secret key from the AA, the data owner and the user request the secret key using a pseudonym. AA has the attributes of owner and user and plays the role of creating a signature secret key and a secret key (ciphertext unsigncryption key) and sending it to the user.
- **Trace Authority (TA):** The TA is a trusted server outside the data-sharing environment. Even though there is AA, the reason for having TA is that if AA has the identity information of data users to provide data traceability, problems such as key escrow may occur with the information of data users in AA. In order not to cause this problem, TA was needed to divide the authority and role of AA. The TA manages the identities and pseudonyms of owners and users. When the AA requests an identity for the pseudonym value in the future, it serves to confirm and inform it.

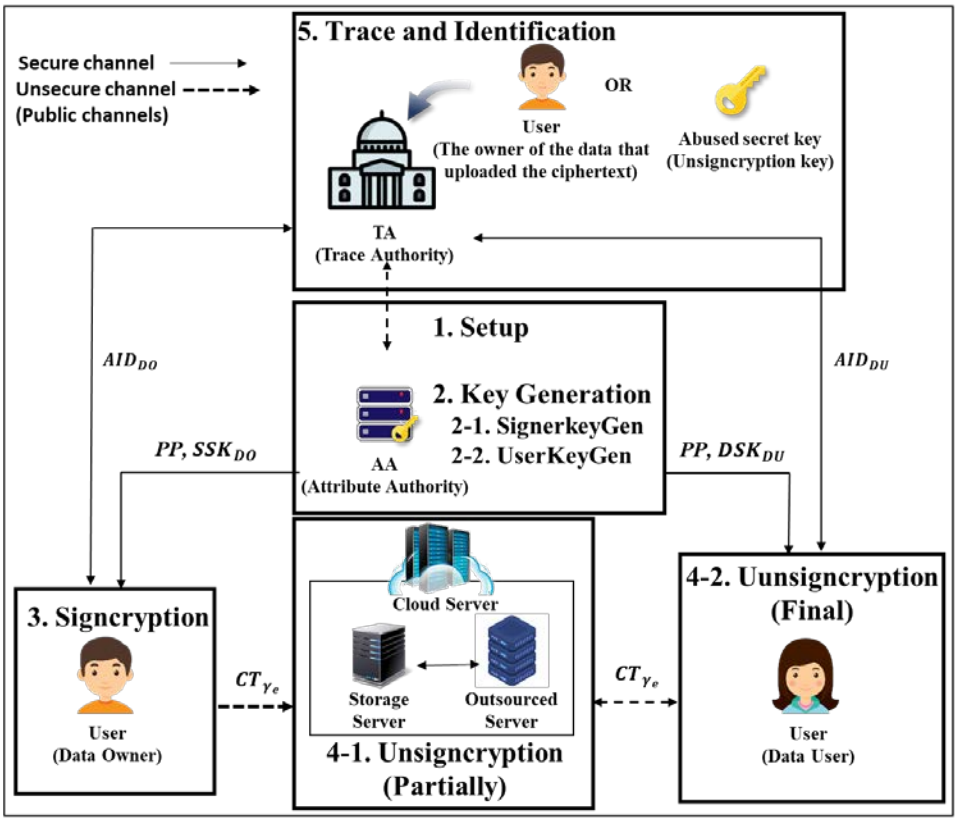

Fig. 3. Overall Scenario of proposed ECP-ABSC scheme

- **Cloud Server:** It usually consists of storage where files are server that performs operations and stored. This proposed scheme is expressed as a cloud server. It performs storage and management of shared data. When the user asks for a ciphertext, it performs partial unsigncryption by comparing the attributes in the ciphertext access structure with the user's attributes and the result is sent to the user.
- **User(Data Owner):** It means the user who signcrypts data and uploads it. The attributes of a user who needs to access the data, an access structure is created based on the attributes of the owner. Then, using the secret signature key and public parameters received from the AA, signcryption is performed, and ciphertext is created and uploaded.
- **User(Data User):** It refers to a user who receives partially unsigncrypted ciphertext from the cloud and performs final unsigncryption to obtain data. After that, you can verify that the data is correct. If there is a problem with the data(the ciphertext), the user can verify by ask the AA and TA to identity who uploaded the data.

4.1.2 System Parameter

The proposed ECP-ABSC method uses the following system parameters (see **Table 1**).

Table 1. System parameter notation ABSC schemes

Symbol	Definition
*	Entities that participate in the scenario (For example, the data owner is denoted as *DO*, and the data user is denoted as *DU*)
PP, MK, PK_{AA}	Public parameter, Master key, AA's public key
SSK_{DO}	Owner's signature secret key
DSK_{DU}	Ciphertext unsigncryption key
AID_*, ID_*	The data owners and user's pseudonym *ID*, the data owners and user's *ID* (For example, the data owner pseudonym and *ID* is denoted as AID_{DO}, ID_{DO}. Data user pseudonym and *ID* is denoted as AID_{DU}, ID_{DU}.)
PSK_*, TI_*	Values that can be traced to data owners and users (For example, the data owner traced value is denoted as PSK_{DO}, TI_{DO}. Data user traced value is denoted as PSK_{DU}, TI_{DU}.
$Att_{i(*)}, A_*$	Attribute data of user's *, attribute data Set (For example, the data owner attribute is denoted as $Att_{i(DO)}$.The attribute set is A_{DO}. Data user attribute is denoted as $Att_{i(DU)}$. The attribute set is A_{DU}.)
γ_s	Access structure for signers (Data owner's)
γ_e	Access structure for ciphertext
CT_{γ_e}	Cipher text of access structure γ_e
C	Result value after performing partially unsigncrypted ciphertext (Access structure γ_e and user attribute comparison operation)
SS	Parameter value generated after signature verification
$H_1(\cdot)$	Cryptographic hash function $\left(\{0,1\}^* \rightarrow Z_p^*\right)$
$H_2(\cdot)$	Cryptographic hash function $(\{0,1\}^* \rightarrow G)$
$H_3(\cdot)$	Cryptographic hash function $\left(\{0,1\}^* \times G \rightarrow Z_p^*\right)$

4.1.3 Procedure

Our CP-ABSC data-sharing scheme identifies data ownership and prevents key leakage. It is possible to verify owner/user identity. In addition, the output of a constant size of the ciphertext and it improves the efficiency of the user's unsigncryption calculation amount through partial unsigncryption. The cloud server and the user perform the unsigncryption process in this proposed scheme separately. Each phase proceeds as follows.

- **Setup phase:** The data owner or user transmits an *ID* to the TA, requests registration, and receives the AID_* corresponding to the ID_*. In addition, *PP* and *MK* are generated by inputting security parameter k in AA.
- **SignerKeyGen phase:** The data owner sends the AID_{DO} to the AA to request the SSK_{DO}. AA generates the data owner's SSK_{DO} with PP, MK, AID_{DO}, A_{DO} and securely transmits PP, SSK_{DO} to the data owner.
- **UserKeyGen phase:** The data user sends the AID_{DU} to the AA and requests the DSK_{DU} (ciphertext unsigncryption key) corresponding to the attribute. AA generates the data user's DSK_{DU} with PP, MK, AID_{DU}, A_{DU} and securely transmits PP, DSK_{DU} to the data user.
- **Signcryption phase:** The data owner creates the γ_e of the ciphertext as the attributes of users and the γ_s based on his/her own attributes. Then, with *PP*, SSK_{DO}, the message *M* is signcrypted. The CT_{γ_e} and γ_s are transmitted together to the cloud server, which stores them.
- **Unsigncryption (Partial) phase:** A user generates a token and requests a CT_{γ_e} from the cloud. The cloud server performs partial unsigncryption by comparing the attribute value

included in the access structure γ_e to those of the user. After partial unsigncryption, the result values *C* and *SS* are transmitted to the user with CT_{γ_e}.

- **Unsigncryption (Final) phase:** The user performs final unsigncryption of SS, C, CT_{γ_e} received from the cloud server with his/her DSK_{DU}. If unsigncryption is successful, the user obtains M. The integrity of the M can then be verified.
- **Trace and Identification phase:** If a problem arises with CT_{γ_e} or leaked DSK_{DU}, it is possible to identify the owner who uploaded CT_{γ_e} or the user who was issued DSK_{DU}. The AA employs the CT_{γ_e} or PSK_*, TI_* included in DSK_{DU} to calculate the pseudonym *ID* (i.e., AID_*) of the owner or user. The AA sends this to the TA and requests identification; the TA accepts the AID_*value and engages in calculation employing this value, and the user attribute values, and extracts the user *ID* and sends it to the AA.

4.2 Description of the Proposed ECP-ABSCE Scheme

In this ECP-ABSC scheme, two cycle multiplication groups G, G_T of prime number p are generated, and a bilinear mapping map $e: G \times G \rightarrow G_T (\forall i, j \in G, e(i,j) = v, v \in G_T)$ is generated. Let g be the generator of *G*. AA creates a subgroup G_2 of elliptic curve points of fractional order q and chooses the generator P of G_2. Suppose there are n attributes in the universe where the universal set is $A_* = \{Att_{1(*)}, Att_{2(*)}, Att_{3(*)}, \dots, Att_{n(*)}\}$. γ_e is a tree-type access structure created with the attributes of the user who needs to access the ciphertext. γ_s is a tree-type access structure composed of the attributes of the data owner. Both access structures are expressed as $\gamma_* = \{\gamma_{1(*)}, \gamma_{2(*)}, \gamma_{3(*)}, \dots, \gamma_{n(*)}\}$. $Att_{n(*)}$ is included in $\gamma_{*(n)}$.

4.2.1 Setup Phase

First, *PP* and *MK* are created through the setup phase in AA. When the prime order of a bilinear group *G* is *p*, the AA generates random value $\alpha, s \in Z_p{}^*, tr_i \in G$, and then generates *PP*, *MK*, and public key as follows.

$$PP < e,\ g, G, G_T, G_2\ \{TR_i = g^{tr_i}\}_{i\in[1,n]}, h = g^s,\ e(g,g)^\alpha, H_1, H_2, H_3 > \quad (1)$$

$$MK < \alpha,\ \{tr_i\}_{i\in[1,n]} >, PK_{AA} =< \alpha \cdot P > \quad (2)$$

The data user or owner sends the *ID* to the TA to request registration. The TA calculates a corresponding pseudonym *ID* value AID_* using the user *ID* and attribute values, and transmits this to the data owner or user. The calculated value *F* is used as *PP*.

- Random number $f \in Z_p{}^*$, $F=g^f$
- Create data owner pseudonym: $AID_{DO} = ID_{DO} \oplus H_2(h^f) \oplus f$
- Create data user pseudonym: $AID_{DU} = ID_{DU} \oplus H_2(h^f) \oplus f$

4.2.2 Key Generation Phase

The data owner and user transmit their pseudonym AID_{D*} values and attributes to the AA when requesting a secret key. In detail, the data owner requests a SSK_{DO} to be used for signcryption, and the data user requests a DSK_{DU}. The AA generates a SSK_{DO} based on the data owner's attributes, and securely transmits PP, SSK_{DO} to the owner. In addition, the AA generates a DSK_{DU} based on the data user's attributes, and securely transmits PP, DSK_{DU} to the user.

- Random number $\beta \in Z_p{}^*$.

$\boldsymbol{< SignerKeyGen(PP,\ MK, AID_{DO}, A_{DO}) >}$

- Random number $r_{DO_i} \in \mathrm{Z}_p{}^*$, $\{r_{DO_{i,n}} \in \mathrm{Z}_p{}^*\}_{i\in[1,n]}$, $r_{DO} = \sum_{i=1}^{n} r_{DO_i}$

$$K = g^{\alpha - r_{DO_i}},\ K_i' = g \cdot H_1(Att_i)^{r_{DO_{i,n}}}, K_i'' = H_1(Att_i)^{\frac{r_{DO_{i,n}}}{tr_i}} \quad (3)$$

$$PSK_{DO} = \alpha + H_1(AID_{DO}) \cdot \alpha \quad (4)$$

$$TI_{DO} = AID_{DO} \oplus \beta \oplus PSK_{DO} \quad (5)$$

$$SSK_{DO} = \{PSK_{DO}, TI_{DO},\ K, \{K_i', K_i''\}_{Att_i \in [A_{DO}]}\} \quad (6)$$

$<UserKeyGen(PP,\ MK, AID_{DU}, A_{DU})>$

- Random number $r_{DU_i} \in Z_p{}^*, \{r_{DU_{i,n}} \in Z_p{}^*\}_{i \in [1,n]}, r_{DU} = \sum_{i=1}^{n} r_{DU_i}$

$$D_i = g^{\alpha - r_{DU_i}}, D_i' = g^{\frac{r_{DU_i}}{tr_i}}, D_i'' = g^{s \cdot r_{DU_i}/tr_i} \quad (7)$$

$$PSK_{DU} = \alpha + H_1(AID_{DU}) \cdot \alpha \quad (8)$$

$$TI_{DU} = AID_{DU} \oplus \beta \oplus PSK_{DU} \quad (9)$$

$$DSK_{DU} = \{PSK_{DU}, TI_{DU},\ D, \{D_i', D_i''\}_{Att_i \in [A_{DU}]}\} \quad (10)$$

4.2.3 Data Signcryption Phase

The data owner signcrypt the data, and sends it to the cloud server. In detail, the data owner creates the access structure γ_e of the ciphertext based on the attributes of users, and creates the access structure γ_s with his/her own attributes. The owner selects a message and signcrypt it using the *PP*, secret signature key, and access structures. Then, send the CT_{γ_e} to the cloud server. The server stores the received CT_{γ_e} and γ_s.

- Random number $o, o' \in Z_p{}^*$

Access structure as $\gamma_e = \{\gamma_{e1}, \gamma_{e2}, \gamma_{e3}, \ldots, \gamma_{en}\}$.

- If $Att_{i,1} \in \gamma_{ei}$, computes $C_i = (g^{tr_i})^o$
- If $Att_{i,1} \notin \gamma_{ei}$, computes $C_i = (g^{tr_{i+1}})^o$

$$C_0 = M \cdot e(g,g)^{\alpha \cdot o},\ C_1 = g^o, C_2 = h^o \cdot \prod_{i \in n} C_i \quad (11)$$

$$V = e(C_1,\ g^{o'}), \qquad VA = H_3(M||V) \quad (12)$$

$$R = g^{o'} \cdot K^{VA},\ R_i' = (K_i')^o \cdot TR_i, R_i'' = (K_i'')^o \cdot TR_i \quad (13)$$

$$CT_{\gamma_e} = <\gamma_e, \gamma_s,\ PSK_{DO}, TI_{DO},\ C_0,\ C_1,\ C_2,\ VA,\ R,\ \{R_i', R_i''\}> \quad (14)$$

4.2.4 Data Unsigncryption (Partial and Final) Phase

The user generates a token and sends it to the cloud server to request the ciphertext. The server performs partial unsigncryption by calculating and comparing the attributes specified in the ciphertext access structure requested by the user with the attributes of the user.

$< Partial\ unsigncryption(CT_{\gamma_e}, A_{DU})>$

$$C = \frac{e(C_2, \prod_{i \in \gamma_e} D_i')}{e(C_1, (\prod_{i \in \gamma_e} D_i''))} = \frac{e(h^o \cdot \prod_{i \in n} C_i, \prod_{i \in \gamma_e} g^{r_{DU_i/tr_i}})}{e(g^o, \prod_{i \in \gamma_e} g^{s \cdot r_{DU_i/tr_i}})} = e(g,g)^{r_{DU_i} \cdot o}$$

$$SS = \frac{\prod_{i \in A_{DO}} e(R_i', g)}{\prod_{i \in A_{DO}} e(TR_i,\ R_i'')} = \frac{\prod_{i \in A_{DO}} e(g \cdot H_1(Att_i)^{r_{DO_{i,n}} \cdot o} \cdot g^{tr_i}, g)}{\prod_{i \in A_{DO}} e(g^{tr_i}, (g^{tr_i} \cdot H_1(Att_i)^{\frac{r_{DO_{i,n}}}{tr_i} o}))} = e(g,g)^{r_{DO_i} \cdot o} \quad (15)$$

After performing partial unsigncryption *C*, *SS*, CT_{γ_e} are sends to the user from the cloud server. The user performs final unsigncryption with DSK_{DU} and *PP*. The obtained message content and integrity are then verified.

$$< \boldsymbol{Final\ Unsigncryption}(\boldsymbol{PP},\ \boldsymbol{DSK_{DU}}, \boldsymbol{C},\ \boldsymbol{SS}, \boldsymbol{CT_{\gamma_e}}) >$$

$$M = \frac{C_0}{e(C_1, D) \cdot C} = \frac{M \cdot e(g,g)^{\alpha \cdot o}}{e(g^o, g^{\alpha - r_{DU_i}}) \cdot e(g,g)^{r_{DU_i} \cdot o}}$$

$$V' = \frac{e(C_1,\ R)}{(e(g,g)^{\alpha \cdot o} \cdot\ SS^{-1})^{VA}} \ \rightarrow\ V'? = V(true\ or\ false)\,, \qquad VA' = VA \tag{16}$$

4.2.5 Trace and Identification Phase

The trace and identification phase verifies the identity of the data owner who uploaded ciphertext, and that of the data user who was first issued a leaked secret key for the first time. If the integrity of data obtained by the user on decrypting the ciphertext is compromised or the wrong data are uploaded, the data owner must be traced to verify their identity. It is also to prevent unauthorized third parties from obtaining keys from someone and accessing the cloud. If there is a problem with a leaked key, the identity information of the data user who was first issued the key is verified. The AA uses PSK_*, TI_* included in CT_{γ_e} or DSK_{DU} to calculate the pseudonym $ID(AID_*)$ of the data owner or user.

$$< \boldsymbol{Trace}(\boldsymbol{PSK_*}, \boldsymbol{TI_*}) >$$

$$PSK_{D*} \cdot P = PK_{AA} + H_1(AID_{D*}) \cdot\ PK_{AA} \tag{17}$$

$$AID_{D*} = \beta \oplus TI_{D*} \oplus PSK_{D*} \tag{18}$$

The AA sends $AID_{D*} \oplus H_2(F^s)$ to the TA to request the identity of the pseudonym. The TA uses the AID_* and the corresponding user attribute value, to perform calculations, and extracts the user *ID* and gives it to the AA.

$$ID_{D*} = AID_{D*} \oplus\ H_2(F^s) \oplus f \tag{19}$$

5. Analysis of proposed scheme

Our proposed ECP-ABSC scheme analyzed the security and efficiency to satisfy the requirements presented in Chapter 3. **Table 2** compares and analyzes the existing CP-ABSC scheme based on the requirements to be provided by CP-ABSC scheme. **Table 3** and **Fig. 4** shows the amount of operation required to be performed in the signcryption phase and the unsigncryption phase of the existing CP-ABSC schemes.

5.1 Security Analysis

- **Data user traceability and identification:** In this proposed scheme, the data owner and user initially receive the pseudonym AID_{D*} after registration from the TA, and perform data sharing. Therefore, anonymity between each entity participating in sharing is provided. In addition, when the data obtained by the data user unsigncryting the ciphertext is unclear(incorrect), the ciphertext may be transmitted to the AA to request the identity of the data owner who uploaded the ciphertext. The AA extracts the pseudonym value $AID_{DO} = \beta \oplus TI_{DO} \oplus PSK_{DO}$ from the ciphertext with the value PSK_{DO}, TI_{DO} that can trace the owner of the data. XOR operation is performed on the extracted pseudonym value with $H_2(F^s)$, and it is transmitted to the TA to request the identity of the data owner. The TA verifies the identity of the data owner corresponding to the pseudonym through operation ($ID_{DO} = AID_{DO} \oplus\ H_2(F^s) \oplus f$), and transmits the information to the AA.

Table 2. Comparison of security requirements between the existing CP-ABSC scheme and the proposed scheme

<table>
<tr><th rowspan="2">ABSC Scheme</th><th rowspan="2">Anonymity</th><th colspan="2">Trace and Identification</th><th rowspan="2">Ciphertext size</th><th rowspan="2">Partial unsigncryption</th></tr>
<tr><th>Data owner</th><th>Data user</th></tr>
<tr><td>Sana et al. scheme</td><td rowspan="4">Provides full anonymity</td><td colspan="2" rowspan="4">Traceability is not taken into account as full anonymity is provided</td><td>Constant-size</td><td>Not provided</td></tr>
<tr><td>Fuhu Deng et al. scheme</td><td>Affected by number of attributes</td><td rowspan="3">Support for outsourced server operation</td></tr>
<tr><td>Hundera et al. scheme</td><td>Constant-size</td></tr>
<tr><td>Ningzhi Deng et al. scheme</td><td rowspan="3">Affected by number of attributes</td></tr>
<tr><td>Lu et al. scheme</td><td>Anonymity provided only by data users</td><td>Traceable (Tracing users with user identifiable values in TA)</td><td>Not traced</td><td rowspan="2">Not provided</td></tr>
<tr><td>Rohit et al. scheme</td><td>Anonymity provided only by data owners</td><td>Not traced</td><td>Traceable (User identity verification through key verification in TA)</td></tr>
<tr><td>The proposed scheme</td><td>Provide anonymity
(Using data owner/user pseudonym ID)</td><td colspan="2">Traceable (Owner/user traceable in TA and AA)</td><td>Constant-size</td><td>Support for outsourced server operation</td></tr>
</table>

From another perspective, when a leaked secret key is detected, the leaked secret key is transmitted to the AA. Then, the first secret key is issued and the identity of the leaked data user is requested. AA extracts the pseudonym value $AID_{DU} = \beta \oplus TI_{DU} \oplus PSK_{DU}$ with PSK_{DU}, TI_{DU} included in the secret key. In the above method, the identity of the data user ID_{DU} can be verified by requesting the TA.

- **Security (confidentiality, integrity) and access control for shared data:** This proposed scheme encrypts, stores, and shares data using CP-ABSC scheme, confidentiality and integrity of data are provided from third parties. In detail, the CT_{γ_e} with γ_e generated based on user attribute $A_{DU} = \{Att_1, Att_2, Att_3, \dots, Att_n\}$ the only users who want to decrypt it are users with attribute $A_{DU} = \{Att_1, Att_2, Att_3, \dots, Att_n\}$ and secret key DSK_{DU} included in the ciphertext. Confidentiality of data is provided because data cannot be obtained even when unsigncryption is attempted by taking the CT_{γ_e} other than that. In addition, the user can check whether the data uploaded by the data owner is correct through the data verification process and the integrity of the shared data through operation ($V' = \frac{e(C_1, R)}{(e(g,g)^{\alpha \cdot o} \cdot SS^{-1})^{VA}} \rightarrow V' ? = V$). In the proposed scheme, when a user requests a ciphertext from the cloud server, the ciphertext access structure is calculated by comparing with the user's attributes. If they match, partial unsigncryption is performed and the ciphertext and partial unsigncrypted result are sent to the user. That is, when a user who does not have an

attribute and a secret key request a ciphertext, the server cannot perform partial unsigncryption. Accordingly, an access control function for the ciphertext is provided by the data owner, and access from unauthorized users can be blocked.

5.2 Efficiency

The computational amount of signcryption/unsigncryption of the proposed scheme was measured on a Windows machine with a 3.50GHz Intel Core i5-4690 processor and 8GB of RAM. The pairing operations, was performed with reference to the pairing-based cryptographic library [28]. The ECC implementation used a Koblitz elliptic curve $y^2 = x^3 + ax + b\ (mod\ p)$ with a=1 and b=1 with a 163-bit random prime efined in $F_{2^{163}}$. In **Fig. 4**, the amount of user unsignryption operation shown is compared only with CP-ABSC schemes that provide partial unsigncryption requirements in **Table 3**.

Table 3. Comparison of calculation amount between the existing CP-ABSC scheme and the proposed scheme

ABSC Scheme	Signcryption	Unsigncryption (server)	Unsigncryption (Data user)
Sana et al. scheme	$(n+8)T_E + (n+5)T_M$	-	$4P + T_E(n+8) + 7T_E$
Fuhu Deng et al. scheme	$(3n+4)T_E$	$(2n+2)P + (3n+2)T_E$	$P + (2n+2)T_E$
Hundera et al. scheme	$(2n+11)T_E + 2H$	$(3n+2)P + T_M + T_E$	$3T_M$
Ningzhi Deng et al. scheme	$(4n+5)T_E + P$	$(4nP)T_E$	$3T_E$
Lu et al. scheme	$(2n+4)T_E + 2nH + (3n+2)T_M + 3M$	-	$(2n+3)P + (n+3)T_E$
Rohit et al. scheme	$5(n+1)T_E + (n+2)T_M$	-	$(2n+2)P + 2M + (n+3)T_E + (n+2)T_M$
The proposed scheme	$P + (2n+5)T_E + H + (n+1)T_M + E$	$(2n+2)P + (2n+3)T_E$	$P + T_E + 2T_M$

P: Paring operation, *E*: Exponentiation, *M*: Multiplication, *n*: Nmber of attributes; *H*: hash operation; T_E: Exponential operation of group *G*; T_M: Multiplication of group *G*;

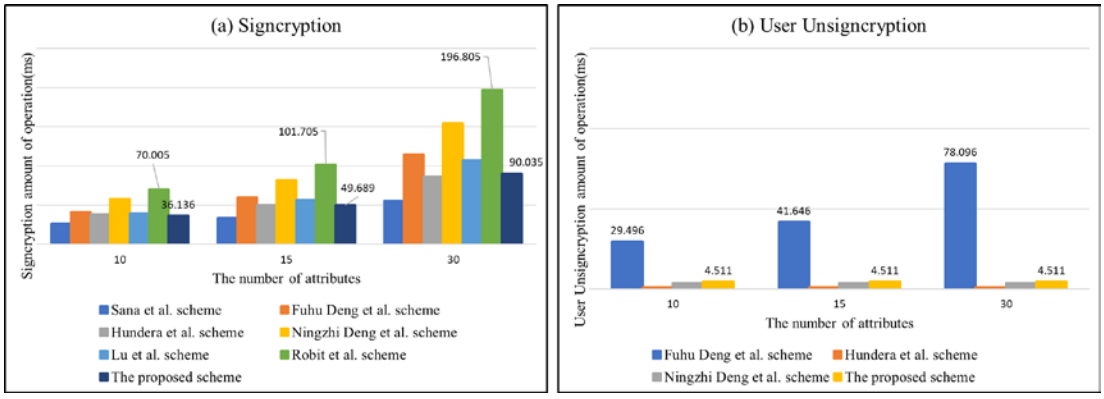

Fig. 4. Comparison of operational amount when performing signcryption and Unsigncryption of user ciphertext

- **Provide a constant-size ciphertext output:** In the proposed ECP-ABSC scheme, an aggregate operation is used to solve the problem that C_i included in the ciphertext increases according to the attribute in the existing CP-ABSC scheme. In detail, C_i is expressed as a single number C_2 by performing an aggregate operation as in $C_2 = h^o \cdot \prod_{i \in n} C_i$. This does not mean that the amount of computation for the ciphertext generation process and the decryption process is reduced, but simply means that ciphertext of a constant size can be output without affecting of the number of attributes. By outputting a ciphertext of a constant size, it effectively affects the ciphertext that is communicated, stored, and shared.
- **Efficient user unsigncryption operation:** This proposed scheme supports a part of the amount of operation required for the user to unsigncrypt the existing ciphertext through partial unsigncryption($C = \frac{e(C_2, \prod_{i \in \gamma_e} D_{i'})}{e(C_1, (\prod_{i \in \gamma_e} D_i''))}$, $SS = \frac{\prod_{i \in A_{DO}} e(R_i', g)}{\prod_{i \in A_{DO}} e(TR_i, R_i'')}$) in the cloud server. That is, since the user can acquire the *M* only by receiving the result *C* and *SS*, the ciphertext CT_{γ_e} from the partial unsigncryption from the cloud server and performing the final unsigncryption. Therefore, compared to the data user's unsigncryption of the ciphertext in the existing CP-ABSC scheme, the amount of user ciphertext unsigncryption operation in the CP-ABSC scheme that provides partial unsigncryption is greatly reduced. As shown in **Table 3**, the efficiency of the amount of unsigncryption operation is high in terms of the user compared to the scheme of Sana et al., Lu et al., Rohit et al. that do not perform partial unsigncryption. However, the CP-ABSC scheme providing partial unsigncryption has a similar amount of unsigncryption operation compared to the scheme of Hundera et al., Ningzhi Deng et al. Since the scheme of Hundera et al., Ningzhi Deng et al. process a large amount of computation by partial unsigncryption in the cloud server, the amount of computational amount of unsigncryption from the user side is more efficient compared to the proposed scheme. However, in this proposed scheme, the requirements (data user tracing and identification) not provided by the scheme of Hundera et al., Ningzhi Deng et al. are provided. the efficiency is good.

6. Conclusion

In this paper, we propose the ECP-ABSC scheme that provides data user traceability, and use it to provide efficient and secure data sharing in a cloud environment. The ECP-ABSC scheme has the following advantages. First, it guarantees the integrity and confidentiality of shared data and blocks access by unauthorized users. Data are securely managed and protected. Second, the anonymity of the user is maintained, and in some cases the data owner or user can be traced to verify their identity. For example, if ciphertext obtained by a user is unclear(incorrect), the identity of the owner who uploaded the ciphertext may be verified. It can also verify the identity information of the user who was initially issued a unsigncryption key obtained by an illegal user. This prevents an authenticated user from exposing the key. In addition, cloud storage can be used efficiently because it provides constant output of the number of attributes without affecting the size of the ciphertext. Finally, the cloud server supports some unsigncryption operations of a user, which aids a user with relatively limited computing resources.

This proposed scheme targets a cloud environment that is used by a large number of data users and owners. In particular, it can be applied in a cloud environment that manages data communicated between nurses/doctors and patients in healthcare, IoT-Cloud environments, and

data shared within companies/public enterprises. This proposed scheme conducted research with the goal of performing security on shared data and authentication of users who access data.

In the future research, damage occurs due to the leakage of personal information due to the sensitive attributes of the access structure. Therefore, it is considered that research that can hide or anonymize sensitive attribute values in the access structure is also needed.

Acknowledgement

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (RS-2022-00167197, Development of Intelligent 5G/6G Infrastructure Technology for The Smart City) and this word was funded by BK21 FOUR (Fostering Outstanding Universities for Research)(No.:5199990914048) and this work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2022R1A2B5B01002490) and the Soonchunhyang University Research Fund.

References

[1] R. Kumar, and R. Goyal, "On cloud security requirements, threats, vulnerabilities and countermeasures: A survey," *Computer Science Review*, vol. 33, pp. 1-48, Aug. 2019. Article(CrossRefLink)

[2] A. Singh and K. Chatterjee, "Cloud security issues and challenges: A survey," *Journal of Network and Computer Applications*, vol. 79, pp. 88- 115, Feb. 2017. Article(CrossRefLink)

[3] X. Zheng, Z. Cai, and Y. Li, "Data Linkage in Smart IoT Systems: A Consideration from Privacy Perspective," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 55-61, Sep. 2018. Article(CrossRefLink)

[4] Z. Cai, Z. He, X. Guan, and Y. Li, "Collective Data-Sanitization for Preventing Sensitive Information Inference Attacks in Social Networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 577-590, July-Aug. 2018. Article(CrossRefLink)

[5] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. of 2007 IEEE symposium on security and privacy (SP'07)*, Berkeley, CA, USA, pp. 321-334, 2007. Article(CrossRefLink)

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

[7] M. Gagné, S. Narayan, and R. Safavi-Naini, "Threshold attribute-based signcryption," in *Proc. of International Conference on Security and Cryptography* for *Networks*, Springer, Berlin, Heidelberg, pp. 154-171, 2010. Article(CrossRefLink)

[8] C. Hu, J. Yu, X. Cheng, Z. Tian, and L. Sun, "CP_ABSC: An attribute-based signcryption scheme to secure multicast communications in smart grids," *Mathematical foundations of computer science*, vol. 1, no. 1, pp. 77-100, Feb. 2018. Article(CrossRefLink)

[9] H. Zheng, J. Qin, J. Hu, and Q. Wu, "Threshold attribute-based signcryption and its application to authenticated key agreement," *Security and Communication Networks*, vol. 9, no. 18, pp. 4914-4923, Oct. 2016. Article(CrossRefLink)

[10] Y. Lu, X. Wang, C. Hu, H. Li, and Y. Huo, "A traceable threshold attribute-based signcryption for mHealthcare social network," *International Journal of Sensor Networks*, vol. 26, no. 1, pp. 43-53, Jan. 2018. Article(CrossRefLink)

[11] J. Wei, X. Hu, and W. Liu, "Traceable attribute-based signcryption," *Security and Communication Networks*, vol. 7, no. 12, pp. 2302-2317, Dec. 2013. Article(CrossRefLink)

[12] H. Hong, X. Zhou, B. Hu, and Z. Sun, "A ciphertext policy attribute based signcryption scheme with secure and flexible key evolving," in *Proc. of 2019 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)*, Xiamen, China, pp. 413-418, 2019. Article(CrossRefLink)

[13] R. Ahuja, S. K. Mohanty, and K. Sakurai, "A traceable signcryption scheme for secure sharing of data in cloud storage," in *Proc. of 2016 IEEE International Conference on Computer and Information Technology (CIT)*, Nadi, Fiji, pp. 524-531, 2016. Article(CrossRefLink)

[14] L. Z. Deng, S. Li, and X. Wang, "Attribute based signcryption with constant ciphertext length," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 16, no. 6, pp. 337-347, 2013. Article(CrossRefLink)

[15] S. Belguith, N. Kaaniche, M. Laurent, A. Jemai, and R. Attia, "Constant-size threshold attribute based signcryption for cloud applications," in *Proc. of SECRYPT 2017: 14th International Conference on Security and Cryptography*, Madrid, Spain, pp. 212-225, 2017. Article(CrossRefLink)

[16] N. Deng, S. Deng, C. Hu, and K. Lei, "An efficient revocable attribute-based signcryption scheme with outsourced unsigncryption in cloud computing," *IEEE Access*, vol. 8, pp. 42805-42815, Dec. 2019. Article(CrossRefLink)

[17] W. H. Negalign, H. Xiong, A. A. Addis, Y. G. Ashenafi, and D. M. Geresu, "Outsourced attribute-based signcryption in the cloud computing," in *Proc. of 2018 15th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, Chengdu, China, pp. 40-44, 2018. Article(CrossRefLink)

[18] Y. W. Hwang and I. Y. Lee, "A study on CP-ABE-based medical data sharing system with key abuse prevention and verifiable outsourcing in the IoMT environment," *Sensors*, vol. 20, no. 17, p. 4934, 2020. Article(CrossRefLink)

[19] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of Computation*, vol. 48, no. 177, pp. 203-209, 1987. Article(CrossRefLink)

[20] Y. Zheng, "Digital signcryption or how to achieve cost (signature & encryption)≪ cost (signature)+ cost (encryption)," in *Proc. of Annual international cryptology conference*, Springer, Berlin, Heidelberg, pp. 165-179, 1997. Article(CrossRefLink)

[21] S. Guo, Y. Zeng, "Attribute-based signature scheme," in *Proc. of International Conference on Information Security and Assurance (ISA 2008)*, Busan, Korea (South), pp. 509–511, 2008. Article(CrossRefLink)

[22] F. Deng, Y. Wang, L. Peng, H. Xiong, J. Geng, and Z. Qin, "Ciphertext-policy attribute-based signcryption with verifiable outsourced designcryption for sharing personal health records," *IEEE Access*, vol. 6, pp. 39473-39486, 2018. Article(CrossRefLink)

[23] R. Zhang, L. Hui, S. Yiu, X. Yu, Z. Liu, and Z. L. Jiang, "A traceable outsourcing cp-abe scheme with attribute revocation," in *Proc. of 2017 IEEE Trustcom/BigDataSE/ICESS.*, Sydney, NSW, Australia, pp. 363-370, 2017. Article(CrossRefLink)

[24] Y. Yang, X. Liu, R. H. Deng, and Y. Li, "Lightweight sharable and traceable secure mobile health system," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 1, pp. 78-91, 2020. Article(CrossRefLink)

[25] Y. Yang, X. Liu, X. Zheng, C. Rong, and W. Guo, "Efficient traceable authorization search system for secure cloud storage," *IEEE Transactions on Cloud Computing*, vol. 8, no. 3, pp. 819-832, 2020. Article(CrossRefLink)

[26] S. Belguith, N. Kaaniche, M. Hammoudeh, and T. Dargahi, "Proud: Verifiable privacy-preserving outsourced attribute based signcryption supporting access policy update for cloud assisted IoT applications," *Future Generation Computer Systems*, vol. 111, pp. 899-918, 2020. Article(CrossRefLink)

[27] N. W. Hundera, C. Jin, M. U. Aftab, D. Mesfin, and S. Kumar, "Secure outsourced attribute-based signcryption for cloud-based Internet of Vehicles in a smart city," *Annals of Telecommunications*, vol. 76, no. 9, pp. 605-616, 2021. Article(CrossRefLink)

[28] B. Lynn, "The pairing-based cryptography (PBC) library," 2012. [Online]. Available: http://crypto.stanford.edu/pbc

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 Security, Cryptography, Attribute-based Encryption, Data Sharing, etc.

TaeHoon Kime received the M.S degrees in Department of Software Convergence from Soonchunhyang University (SCH), Asan, South Korea, in 2021, respectively. He is now a Ph.D. candidate in Department of Software Convergence from Soonchunhyang University (SCH), Asan, South Korea. His research interests include Information Security, Blockchain, Key Management, Privacy, Decentralized Identifier, etc.

DaeHee Seo received the B.S. degree in electronic and electrical engineering from Dongshin University, Naju, South Korea, in February 2001, and the M.S. degree in computer science and engineering and the Ph.D. degree in computer science from Soonchunhyang University (SCH), Asan, South Korea, in February 2003 and February 2006, respectively. He is currently an Assistant Professor with the Faculty of Artificial Intelligence and Data Engineering, SangMyung University (SMU), Seoul, South Korea.

Im-Yeong Lee is corresponding author. He received the B.S. degree in electronic engineering from Hongik University, Seoul, in 1981, and the M.S. and Ph.D. degrees in information and communication engineering from Osaka University, Osaka, Japan, in 1986 and 1989, respectively. He is currently a Professor with the Department of Computer Software Engineering, Soonchunhyang University (SCH), Asan, South Korea. His research interests include information security, cryptographic protocol, information theory, and data communication.