sensors

MDPI

Article

A Study on CP-ABE-based Medical Data Sharing System with Key Abuse Prevention and Verifiable Outsourcing in the IoMT Environment

Yong-Woon Hwang and Im-Yeong Lee *

Department of Computer Science and Engineering, Soonchunhyang University, Asan 31538, Korea; hyw0123@sch.ac.kr

* Correspondence: imylee@sch.ac.kr; +82–41–530–1323

Received: 29 July 2020; Accepted: 29 August 2020; Published: 31 August 2020

Abstract: Recent developments in cloud computing allow data to be securely shared between users. This can be used to improve the quality of life of patients and medical staff in the Internet of Medical Things (IoMT) environment. However, in the IoMT cloud environment, there are various security threats to the patient's medical data. As a result, security features such as encryption of collected data and access control by legitimate users are essential. Many studies have been conducted on access control techniques using ciphertext-policy attribute-based encryption (CP-ABE), a form of attribute-based encryption, among various security technologies and studies are underway to apply them to the medical field. However, several problems persist. First, as the secret key does not identify the user, the user may maliciously distribute the secret key and such users cannot be tracked. Second, Attribute-Based Encryption (ABE) increases the size of the ciphertext depending on the number of attributes specified. This wastes cloud storage, and computational times are high when users decrypt. Such users must employ outsourcing servers. Third, a verification process is needed to prove that the results computed on the outsourcing server are properly computed. This paper focuses on the IoMT environment for a study of a CP-ABE-based medical data sharing system with key abuse prevention and verifiable outsourcing in a cloud environment. The proposed scheme can protect the privacy of user data stored in a cloud environment in the IoMT field, and if there is a problem with the secret key delegated by the user, it can trace a user who first delegated the key. This can prevent the key abuse problem. In addition, this scheme reduces the user's burden when decoding ciphertext and calculates accurate results through a server that supports constant-sized ciphertext output and verifiable outsourcing technology. The goal of this paper is to propose a system that enables patients and medical staff to share medical data safely and efficiently in an IoMT environment.

Keywords: internet of medical things; cloud storage; access control; CP-ABE; traceability; verifiable outsourcing; user privacy protection

1. Introduction

Cloud computing has become very popular, and users can safely store and share data via the cloud. With these advantages, it is possible to apply cloud environments to many fields, including Business to Consumer (B2C), enterprise, public, and Internet of Things (IoT) services. Recently, IoT and the cloud have been combined to collect patient data as via wearable devices in the medical field through Internet of Medical Things (IoMT), and to store it in a hospital cloud environment so doctors and nurses can remotely monitor it to check the patients' health. Figure 1 shows a flow diagram of a medical data sharing system for patients in a cloud-based IoMT environment. Here, the data owner

is assumed to be a patient, and the cloud server is assumed to be a medical environment server. A data user refers to a user who has the authority to check the data of the data owner, such as a doctor or a nurse, regardless of gender. For example, it means a doctor or nurse in charge who is examining a patient. As shown in Figure 1, the patient's medical data (physical signal) collected by healthcare wearables is stored on a smartphone, and if it is transmitted to the hospital cloud server, a doctor or nurse in the department in charge of the patient can remotely monitor the patient's data. By examining the patient's medical data and sending prescription data to the cloud server, the patient can check the prescription results smart devices. Therefore, since the introduction of IoMT enables safe data sharing between patients and doctors or nurses in hospitals, patients can manage their health in a fast and convenient environment. However, there are some security threats when sharing data in an IoMT cloud. First, service providers cannot be completely trusted. If a user employs a cloud provided by a large enterprise, stored data are safe from external threats, but the cloud provider can infringe user privacy because the provider has access to the data. Especially in the medical cloud environment, the stored data is very sensitive. Therefore, security of stored data is important in a cloud environment. In addition, access control technology and user authentication technology for accessing encrypted data are required. Various security technologies exist, among which Attribute-Based Encryption (ABE) is popular. ABE has the advantage of ensuring the user's anonymity because the data is encrypted with the user's attributes. This is used to protect privacy in an environment where sensitive data such as healthcare information is sent and received. ABE includes both ciphertext-policy (CP)-ABE and key-policy (KP)-ABE. The differences are described in Section 2 and the proposed method considers the CP-ABE method [1,2]. The CP-ABE method is an encryption that allows 1:N data sharing. When a data owner encrypts and uploads data, any user meeting owner-defined attributes can decrypt the ciphertext (CT). However, an attribute authority (AA) is required to manage user attributes. Nevertheless, CP-ABE-based access control techniques are often used to access data in cloud environments [3,4]. Various CP-ABE based access control technologies have been studied to date. The existing CP-ABE access control methods do not identify the user who issues the secret key. Accordingly, a user can leak the secret key for his own benefit and that user cannot be traced. For example, in Figure 1, if a doctor gives another user the secret key to access the medical cloud for malicious purposes, the other user can access the medical cloud with that key. If the secret key sent to another person is misused and you want to track the user (doctor) who issued that secret key, the secret key does not reveal the user who was issued the secret key because there is no information about the user [5]. This is a situation that can occur due to the use of CP-ABE. To solve this problem, when the attribute authority, which verifies the attribute and issues the key, issues the key for each user, it records the user's ID value or user signature value in the key parameter [6–11]. If the key is misused after the key is delegated or delivered to someone, research is being conducted to verify the identity by tracking the user who initially issued the key with the user ID value or user signature value included in the key parameter. However, since user attributes and information are exposed to Attribute Authority (AA), it may infringe on the privacy of users. Most of all, attribute-based encryption is characterized by not having to reveal the identity of multiple participants because it accesses the user's attributes. Therefore, if the user ID value is managed by the AA, the anonymity of the cloud user will not be ensured. Another problem is that because data is encrypted with attribute-based encryption, the size of the ciphertext increases with the number of attributes contained in the access structure [12–14]. This can waste space in cloud storage, and when the user decrypts the ciphertext the amount of computation increases with the number of attributes, which affects the user's ciphertext decryption speed. Lastly, since the amount of computation is large when the user decrypts the ciphertext, an outsourcing server is needed to support the decryption operation during distribution. Additionally, the user should know whether the result value calculated by the outsourcing server is a properly calculated result value according to the result value after the final decryption. Therefore, a verification process is needed to prove that the result calculated by the outsourcing server is a properly calculated result [15–23]. In this paper, we propose a system that can safely share medical data among multiple users who use the cloud in an IoMT environment and

provide protection for user privacy and from key abuse. In detail, based on CP-ABE access control techniques where various security requirements have been studied, such as outsourcing operations,

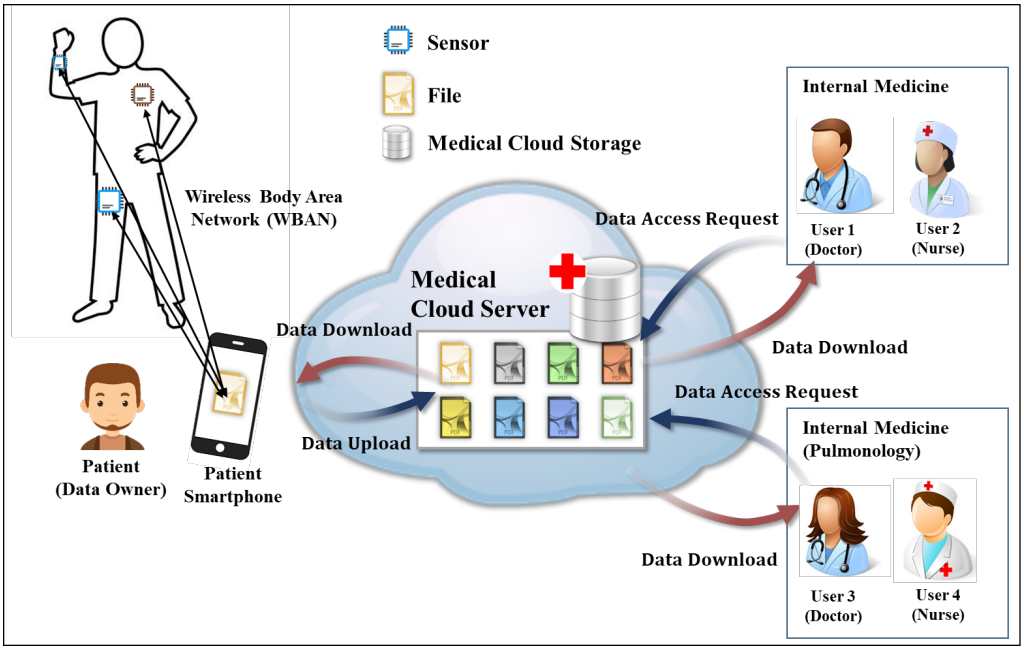

Figure 1. Data sharing system in a medical cloud environment.

partial decoding, and ciphertext constant-size output, we will add techniques that can track the identity of the user issued the key, as well as verification outsourcing techniques.

The contributions in this paper are as follows. (1) User anonymous ID usage and traceability. Users use anonymous IDs in the cloud environment and there is an object called Trace Authority (TA) in the cloud to track down and identify the user who has been issued the key in the event of a problem with a user's key [24]. (2) Output of constant size of ciphertext, regardless of the number of attributes. The efficiency of the storage space of the cloud storage can be improved. (3) Provide verifiable outsourcing techniques. The Access Control (AC) server can reduce the ciphertext decryption computations for user decryption by performing partial decryption on outsourcing servers. In addition, verification allows users who have performed final decryption to verify the integrity of outsourced ciphertext transformations and whether the data was uploaded by the data owner.

The scheme proposed in this paper is not intended for all cloud environments. It will be applicable to cloud environments focused on security of sensitive data such as military and medical and protects the privacy of users accessing the data. Each chapter of this paper is as follows. Section 2 is a related study that describes the formulas used for attribute-based encryption (bilinear map and complexity assumptions), data security technologies in cloud environments, and content on attribute-based encryption (definition, key abuse problem, need for verifiable outsourcing technology, previously proposed studies), and security models. Section 3 describes the security requirements in the IoMT environment, and Section 4 describes the proposed scheme. Section 5 explains the analysis of the security and efficiency of the proposed scheme, and Section 6 concludes with a conclusion.

2. Related Works

This section describes bilinear mapping and attribute-based encryption, and explores the need for traceability and security models in the existing CP-ABE methods.

2.1. Bilinear Map

Consider an additive group G_1 and a multiplicative group G_2 having the same constant q. Assuming that it is difficult to solve the discrete logarithm problem in groups, let P be the

constructor of the additive group G_1. Let $e: G_1 \times G_1 \rightarrow G_2$ be a folded bilinear map satisfying the following properties:

- Bilinearity: For all $P, Q \in 1$ and all $a, b \in Z$, $e(aP, bQ) = e(P, Q)ab$.
- Non-Degeneracy: For all $Q \in G_1$ if, $e(P, Q) = 1$, then $P = 0$.
- Computability: There exists an efficient algorithm computing $e(P, Q)$ for all $P, Q \in G_1$.

2.2. Complexity Assumption

2.2.1. Bilinear Diffie Hellman (BDH) Assumption

Given two pairs$(g^a, g^b, g^c, T = e(g, g)^{abc})$ and $(g^a, g^b, g^c, W = e(g, g)^z)$, the deterministic BDH assumption means that there is no algorithm A that can distinguish the two pairs with a meaningful probability. Here it is $a, b, c, z \in Z_p$. If algorithm A, which solves the deterministic BDH assumption, satisfies $|\Pr[A(g^a, g^b, g^c, T) = 1] - \Pr[A(g^a, g^b, g^c, W) = 1]| \geq \epsilon$, it is said that algorithm A has a profit of ϵ.

2.2.2. Bilinear Diffie Hellman Exponent (BDHE) Assumption

The deterministic BDHE assumption means that, given $(h, g, g^{\alpha} \ldots., g^{\alpha\beta}, g^{\alpha\beta+2}, \ldots g^{\alpha 2\beta})$, there is no algorithm A that can compute $T = e(h, g)^{\alpha\beta+1}$ with a meaningful probability. Here is $h, g \in G_1$. As defined by $g_i = g^{\alpha^i} (i = 1, \ldots, 2B)$ and $g_{\alpha,\beta} = (g_1, \ldots, g_B, g_{B+2}, \ldots, g_{2B})$, when the next two pairs are $(h, g, g_{\alpha,\beta}, T = e(h, g)^{\alpha\beta+1})$, $(h, g, g_{\alpha,\beta}, W = e(h, g)^z)$, the algorithm A solving the deterministic BDHE assumption is$|\Pr[A(h, g, g_{\alpha,\beta}, T) = 1] - \Pr[A(h, g, g_{\alpha,\beta}, W) = 1]| \geq \epsilon$ is satisfied, algorithm A is said to have profit of ϵ.

2.3. Data Security Technology in the Cloud

Encryption technologies protecting stored cloud data include symmetric and public key encryption. ID-based encryption (IDE) (including public key encryption, ABE, and proxy re-encryption) is used to safely share data. Existing symmetric encryption that uses the same key for encryption and decryption in large distributed environments, including the cloud environment, has key distribution and management problems. On the other hand, the existing asymmetric encryption methods using a public key for encryption and a private key for decryption lack computational efficiency. It is not recommended to use such a 1:N environment to encrypt data before sending data to a group because the data owner needs to know who the recipients are, requiring IDs or public keys before encrypting data and sending them individually to authorized users. Several cryptographic access control methods have been developed to reduce the computational costs of existing operations and achieve data confidentiality, access control, and representational policies for data stored in dynamic environments. The most frequently used access control type is ABE [3].

Attribute-based encryption is encryption that allows 1:N different recipients, and when the data owner encrypts the data and uploads it to storage, an unspecified number of participants with the corresponding attributes can access the ciphertext and decrypt it. In other words, the subject who accesses the data ciphertext can designate the owner of the data based on the user's attributes. Accordingly, data access control and data sharing system methods using attribute-based encryption in the cloud environment are being studied a lot. This study also intends to propose a secure data sharing system using a CP-ABE type of attribute-based encryption.

2.4. Attribute-Based Encryption (ABE)

ABE performs encryption and decryption depending on user attributes and an access structure. ABE includes CP-ABE and KP-ABE. At the left of Figure 2 is the CP-ABE method [1,2]. The owner of the data creates an access structure with the attributes of the users who can access the data and creates a ciphertext using the public parameters, master key, and access structure. After, the data is encrypted, and it is transmitted to the cloud. Thereafter, users who want the data then access the cloud to receive

the ciphertext, and compare their own attributes with those specified in the ciphertext's access structure to decrypt the ciphertext. For example, if a user who wants to access the data has the [Internal Medicine] and [Doctor] attributes, as shown in Figure 2, the data owner creates and encrypts an access structure including the [Internal Medicine, Doctor] attributes. Then, only the users with the attributes matching the access structure can decrypt the ciphertext.

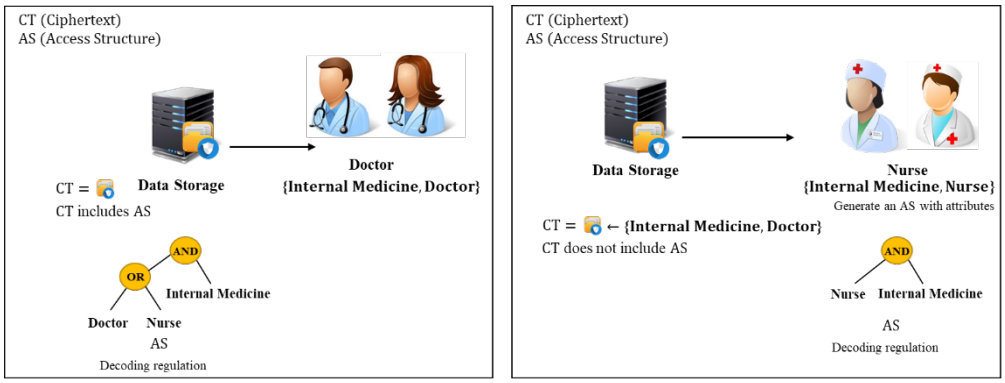

Figure 2. Attribute-Based Encryption (Left ciphertext-policy attribute-based encryption (CP-ABE), Right key-policy attribute-based encryption (KP-ABE)).

The KP-ABE method is as shown at the right of Figure 2. When the data owner creates a ciphertext, the data is encrypted with the attributes of the users who can access the data and is transmitted to the cloud. After that, a user who wants the data can access the cloud to receive the ciphertext, and the user can decrypt the ciphertext by creating an access structure based on their attributes and by generating a key to decrypt the ciphertext. For example, the data owner encrypts the data with the property [Internal Medicine, Nurse] and uploads it to the cloud. After that, a user with the attributes of [Internal Medicine] and [Nurse] can create an access structure to create a key to decrypt the ciphertext.

The difference is between who generates the access structure and key. The owner of the data can specify who can access the data in both CP-ABE and KP-ABE. CP-ABE requires an authority that can issue a key after verifying user attributes, but KP-ABE does not need an authority that can issue a key because the user generates a key directly. This paper projects conducted research on CP-ABE used in a common cloud sharing environment.

Figure 3 is a schematic diagram of a CP-ABE-based data sharing system for safe and efficient data sharing in a medical cloud environment. The data owner creates an access structure with the attributes of the users who can access the data and uploads it to the cloud. Later, any number of unspecified users with the corresponding attributes may access and decrypt the data. The advantage here is that the data can be accessed and decrypted when conditions such as attribute matching are met, so that anonymity for identities of users is provided. In addition, compared to the proxy re-encryption or ID-based encryption methods, it has an advantage that the CP-ABE method does not require a verification process, whereas the user receives data from the data owner for confirmation of data access when the user requests data from the cloud server. In addition, it can be applied to data sharing environments of IoMT and military and corporate cloud systems [5].

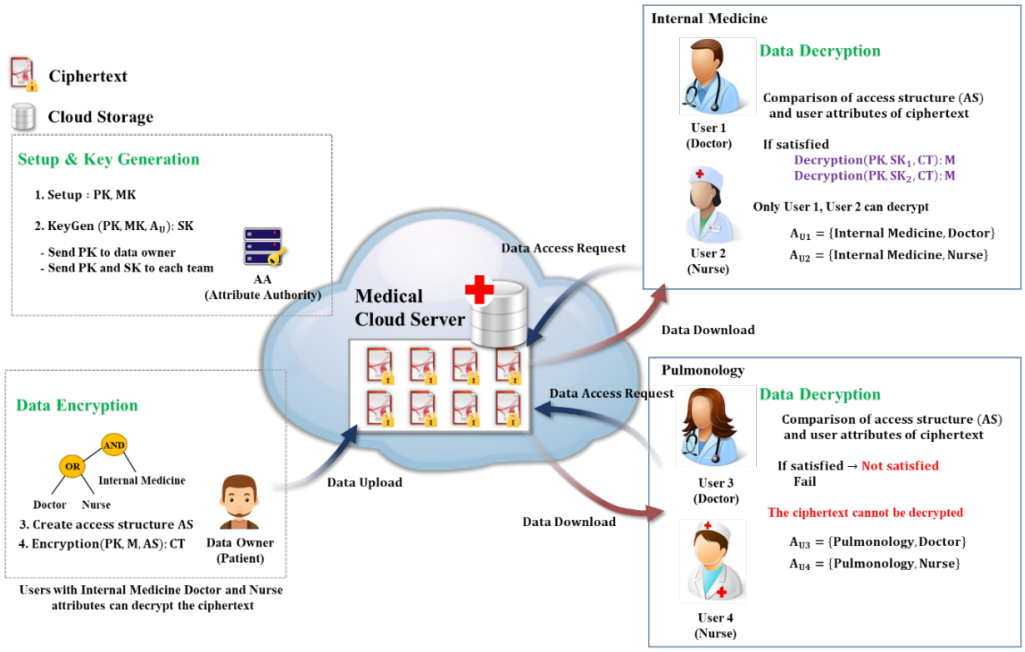

Figure 3. Medical data sharing system based on CP-ABE in a cloud Internet of Medical Things (IoMT) environment.

2.4.1. Key Abuse Issues in CP-ABE

In existing CP-ABE methods, a user requests registration from the AA, and the AA generates and transmits a key employing user attributes to permit decryption. In this case, the secret key $SK = (D = g^{\frac{\alpha+\gamma}{\beta}}, \forall j \in S: D_j = g^{\gamma} \cdot H(j)^{r_j}, D_j' = g^{r_j})$ is a parameter operation that includes the user attribute value, and there is no value to identify the user. Thus, if the user maliciously distributes the key to a third party, that party can access the cloud and decrypt the data. Figure 4 shows a scenario where a package file worth $100 is placed for 10 days on the cloud server. User 1 purchases it and employs it for 10 days. If User 1 sells the key to Users 2 and 3 for $60 each, those users can also access the file and user 1 cannot be identified. The problem here is that there is a process in which the key is distributed, but above all, the problem is that the user who was issued the key cannot be known. This is a fundamental problem with CP-ABE. To solve this, as shown in Table 1, CP-ABE schemes that provide traceability to prevent key abuse have been proposed [5–10]. However, CP-ABE schemes that provide traceability can invade user privacy by exposing the users' identifier values to the AA when the keys are issued, and thereby the anonymity of the users is reduced. Accordingly, this study proceeds in the direction of verifying the identity of the user through a tracking process when the key that is distributed is later misused while protecting the privacy of the users through anonymous IDs.

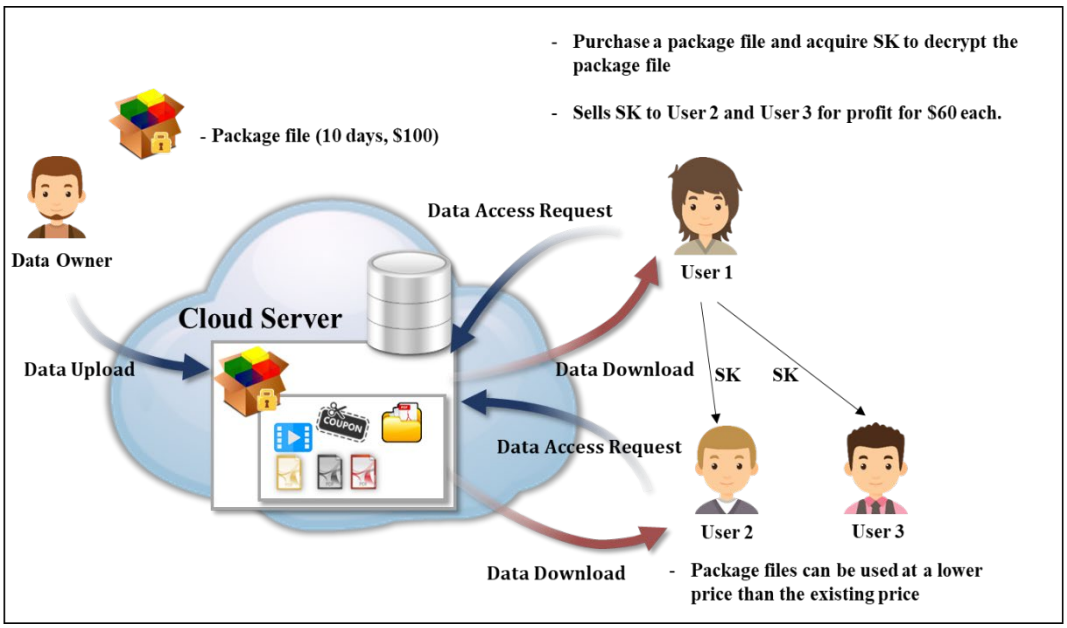

Figure 4. Key abuse and misuse issues that can occur in the CP-ABE data sharing system.

2.4.2. Necessity of Verifiable Outsourcing Techniques

As the cloud features many outsourced servers processing data and owner/user messages, computational demands are high. In addition, data can be shared among mobile devices and the Internet of Things. However, users do not know if data was properly converted by the process of encrypting and decrypting messages on a server that supports outsourcing. In particular, although the data owner encrypts the message M, if the value decrypted by the actual user is M', it is impossible for the user to know that the M' value is different from the original M. Accordingly, it is necessary for the user to verify that the calculated value returned by the outsourcing server is a properly calculated value while performing the final decryption [15–18].

2.4.3. Previously Proposed CP-ABE Scheme

To date, CP-ABE aims to improve the efficiency of computation by preventing key abuse and decrypting ciphertext, as shown in Table 1. Table shows CP-ABE methods that deal with key security as well as other security requirements for CP-ABE methods, such as traceability for tracking key abuse, constant size output of ciphertext, and the application of outsourcing techniques for efficiency of operation. Among the methods to provide traceability, the Liu scheme [6], Yu scheme [8], and Li scheme [5] are $SK = (D = g^{\frac{\alpha+\gamma}{\beta}}, \forall \mathrm{j} \in \mathrm{S}: D_j = g^{\gamma} \cdot H(j)^{\mathrm{r}j}, D_j{}' = g^{\mathrm{r}j})$ and contain the user value (signature, ID) value. AA is the identity table that manages the user's information, and the issued key suggests a scheme. If the issued key is misused, a value that can obtain the user's identifier is extracted from the misused key, and the value is sent to AA to verify the identity of the user who initially issued the key. Assuming that in Li's scheme, AA maintains the ID table of registered users, AA tracks the user after verifying the key with Key Sanity Check as follows. The Key Sanity Check process in the Li scheme is as follows:

$$\begin{gathered} K' \epsilon Z_N, K, L, L', K_x \epsilon \mathrm{G} \\ e(g, L') = e(\mathrm{g}^{\alpha}, L') \neq 1 \\ e\left(\mathrm{g}^{\alpha} \cdot \mathrm{g}^{K'}, \mathrm{K}\right) = e(g, g)^{\alpha} \cdot e\left(\mathrm{L}^{K'} \cdot L', h\right) \neq 1 \\ x \in S, s.t. e(U_x, \mathrm{L}^{K'} \cdot \mathrm{L}') = e(g, K_x)^{\alpha} \neq 1 \end{gathered} \quad (1)$$

After verification, $K' = c$ is extracted from the key and transmitted to the AA, and the AA checks the user ID value corresponding to $K' = c$ to confirm the identity of the user who was first issued the key.

$$SK = \left(K = g^{\frac{\alpha}{a+c}} h^t R,\ K' = c, L = g^t R_0, L' = g^{at} R'_0, \left\{K_x = U_x{}^{(a+c)t} R_x\right\}_{x \in S}\right) \quad (2)$$

However, if the AA, which is an organization that manages attributes included in the Li scheme, manages the identity table of registered users, the anonymity of users will not be preserved, which infringes the privacy of users. In addition, the Li scheme, which proposed traceability as the main core, was not considered in terms of efficiency. In particular, the size of the ciphertext is proportional to the number of attributes, which wastes space in the storage of data and increases the user's decryption computations.

Table 1. Comparison of CP-ABE schemes provided for traceability.

CP-ABE Scheme	User Privacy	Traceability	Ciphertext Size	Support on Outsourcing Server	Outsourcing Results Verification
Qi Li scheme [5]	Not protected	Provided using identity table on management server	Proportional to the number of attributes	Provided	Not provided
Liu scheme [6]				Not provided	
Hahn scheme [7]			Constant size ciphertext	Provided	
Yu scheme [8]		Provided through the signature value included in the key	Proportional to the number of attributes	Provided	
Luo scheme [11]				Not provided	
Jiang scheme [12]		Provided using identity table on management server			

In 2015, the Hahn scheme [13] proposed an attribute-based, secure data-sharing technique supporting outsourcing of cryptographic and decryption operations of a constant size. However, the system is applicable only to private cloud environments. It is not suitable for public cloud environments that share data with others. In 2017, the Helil scheme [15] proposed a hierarchical attribute-based access control method using a constant size ciphertext, but the size of the ciphertext is proportional to the number of attributes, and storage space is wasted. Since the $\forall y \in \mathrm{AS}: C_y = g^{q_y(0)}, C'_y = H\left(att(y)\right)^{q_y(0)}$, part of the ciphertext (CT) assigned to the access structure, the key size, increases with the number of attributes:

$$\mathrm{CT} = \begin{pmatrix} AS, \mathrm{C}' = \{\mathrm{M}\}_{Key}, \mathrm{C}^* = \mathrm{g}^s, \forall y \in \mathrm{AS}: \\ C_y = g^{q_y(0)}, C'_y = H\left(att(y)\right)^{q_y(0)} \end{pmatrix} \quad (3)$$

Although the computations in the encryption process do not change by multiplying the number of different attributes by one constant size when creating the access structure, research on constant-size output is needed because the size of the ciphertext can be minimized. Table 2 shows the CP-ABE methods that dealt with the verifiable outsourcing method and the constant size output of the cipher text among the various security requirements for CP-ABE methods.

Table 2. Comparison of CP-ABE schemes with outsourced verification.

CP-ABE Scheme	Ciphertext Size	Support on Outsourcing Server	Outsourcing Results Verification
Hahn scheme [13]	constant-size ciphertext	Provided	Failed to provide
Wei Teng scheme [14]		Failed to provide	

Helil scheme [15]	Proportional to the number of attributes	Provided	Failed to provide
Lai scheme [16]			Verify by inserting the MAC authentication code in the cipher text.
Premkamal scheme [17]			Message verification with VK, which can verify ciphertext.
Qin scheme [18]	constant-size ciphertext		The hash value is used to verify the accuracy of outsourcing decrypt.
Jiguo Li scheme [20]			Verify by inserting the MAC authentication code in the cipher text.
Zhidan Li scheme [21]	Proportional to the number of attributes		After partial decoding on 2 servers, verify that the results match.

The schemes (Lai scheme [16], Jiguo Li scheme [20], and Zhidan Li scheme [21]) that proposed verifiable outsourcing encrypt two different messages when generating a ciphertext and generate a value that can verify the two messages. Later, the user verifies the integrity of the message by creating a verification value with the two messages obtained by decrypting the ciphertext. For illustration, the Jiguo Li scheme [20] generates the ciphertext as follows.

$$\mathrm{CT} = \left(AS, \widehat{C}, C_1, C_2, C_3, C_1', C_2', C_3'\right) \quad (3)$$

where C_1, C_2, C_3 is the encryption value for M, C_1', C_2', C_3' is the encryption value for M', and $\hat{C} = u^{H(M)} v^{H(M')} d$ is the step of verifying the message integrity with the message M, M' obtained by the user later. In the above method, the integrity of the message obtained by the user is verified. However, it is not known whether the data was actually uploaded by the owner of the data, and the encryption and decryption operations are high compared to the previous CP-ABE methods. Therefore, it is necessary to introduce a signature-based verification step that can verify whether the data was uploaded by the data owner while also verifying the integrity of the message acquired by the user, and it is necessary to introduce an outsourcing method capable of efficient calculation.

2.4.4. Security Model

Our attack model is similar to that of the CP-ABE introduced by Zhou and Huang (2010), based on a semantic security game [25]. The game engages in message transmission and reception between probabilistic polynomial time adversaries (PPT attacker) and a challenger; the probability that the attacker can finally compromise the safety of the CP-ABE technique is the game. The probability of winning is derived. The game details follow:

- Init: The attacker selects the challenger access policy W and gives it to the challenger.
- Setup: The challenger performs the Setup step to generate the public key PK_S, and the challenger sends PK_S to the attacker.
- Phase: The attacker requests the secret key for the following access policy L from the challenger. At this time, if L does not satisfy W the challenger sends the secret key SK corresponding to L to the attacker. The attacker can repeat Phase 1 as needed.
- Challenge: The challenger runs the encrypt algorithm to derive $(< \mathrm{C}_{0,}\mathrm{C}_1 >, \mathrm{Key})$. The challenger sets $\mathrm{Key}_0 = \mathrm{Key}$ for the original Key, and then generates a random key Key_1. At this time, the sizes of the two keys $\mathrm{Key}_0, \mathrm{Key}_1$ are the same. The challenger selects a random value $\mathrm{b} \in \{0,1\}$ and sends $(< \mathrm{C}_{0,}\mathrm{C}_1 >, \mathrm{Key}_\mathrm{b})$ to the attacker.
- Guess: The attacker guesses $\mathrm{b}' \in \{0,1\}$ for the ciphertext. If L does not satisfy W and $\mathrm{b}' = \mathrm{b}$, then the attacker is defined as winning. The final definition of the probability that an attacker can win the game in this game is $\Pr[\mathrm{b}' - \mathrm{b}] - 1/2$.

Definition 1. The algorithm is termed a semantic security algorithm if the attacker receives a negligible benefit in the above game within a polynomial time t [25–27].

3. Security Requirements

This section looks at security of the cloud-based IoMT environment, and describes the problems that may occur when using CP-ABE as shown in Figure 5.

- Collusion attacks: Users can infer other users' attributes through collusion with each other and generate another user's secret key with the inferred attributes. Therefore, when a AA generates a secret key, it is necessary to generate a secret key by applying various variables in addition to the user's attributes. In addition, users can leak data through collusion attack with service providers. Therefore, even if data is leaked through a collusion attack, data security technology so only a legitimate user can decrypt and view it is required.
- **Unauthorized user access control:** Since the cloud is a public environment with a large scope, anyone can access the data stored therein, thereby creating various security threats. Accordingly, access control technology and security technology for accessing stored data are required. If attribute-based encryption is used among security technologies, only users with attributes previously specified by the data owner can access the stored data. Therefore, it is necessary to apply attribute-based encryption because it can provide the confidentiality and integrity of data.
- Tracking users through a distributed key: The problem of the basic CP-ABE method is that there is no value that can identify the user issued a key, so it is impossible to identify the user who was first issued a distributed key. Therefore, if the distributed key is misused, study is needed to verify the identity of the user first issued the key through a tracking process [28].
- **User privacy protection:** Attribute-based encryption ensures anonymity because data owners and users encrypt and decrypt data with their own attributes. However, in order to provide traceability, the user's privacy can be infringed by exposing the user's identifier value to the AA when the key is issued by the attribute verification agency. Therefore, research is needed to protect users' privacy in the cloud.
- **Verify data integrity as uploaded by data owner:** In existing CP-ABE schemes, it is assumed that if the user accesses the ciphertext uploaded by the data owner and decrypts the ciphertext to obtain the message, it is a legitimate message. In addition, it is assumed that the result of the partially decrypted ciphertext is a legitimate message because the outsourcing server is trusted in the schemes. However these are in correct assumptions. The message uploaded to the cloud can be falsified, and it is not known whether the value calculated by the outsourcing server is the correct value [29,30]. Accordingly, it is necessary to verify whether the user's final decrypted value is the original message of the data owner.
- **Efficiency:** In some of the existing CP-ABE methods, the size of the ciphertext is proportional to the number of attributes specified in the access structure when generating the ciphertext. Accordingly, the size of the ciphertext increases linearly with the number of attributes, which occupies costly cloud storage space. In addition, the size of the ciphertext increases the burden of computation for the decrypting user. In order to solve this, it is necessary to introduce a server that supports outsourcing that can partially process the computation amount, and it is necessary to study how to reduce the computation amount and reduce the burden.

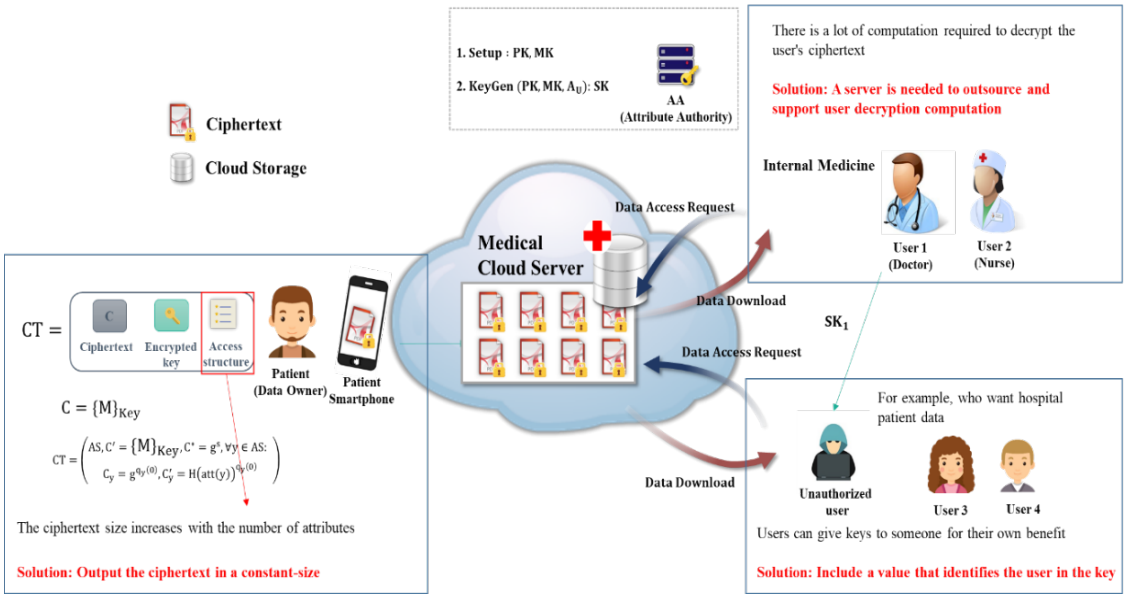

Figure 5. Problems that can occur in CP-ABE.

4. Proposed Scheme

This section proposes a CP-ABE-based access control method that provides user privacy protection and key abuse prevention for secure data sharing in a medical cloud environment.

As shown in Figure 6, the participants in this proposed scheme consist of five objects: data owner, cloud server (storage, access control (AC)), tracing authority (TA), attribute authority (AA), and user. A detailed description of the object role and protocol flow proposed in this paper are below.

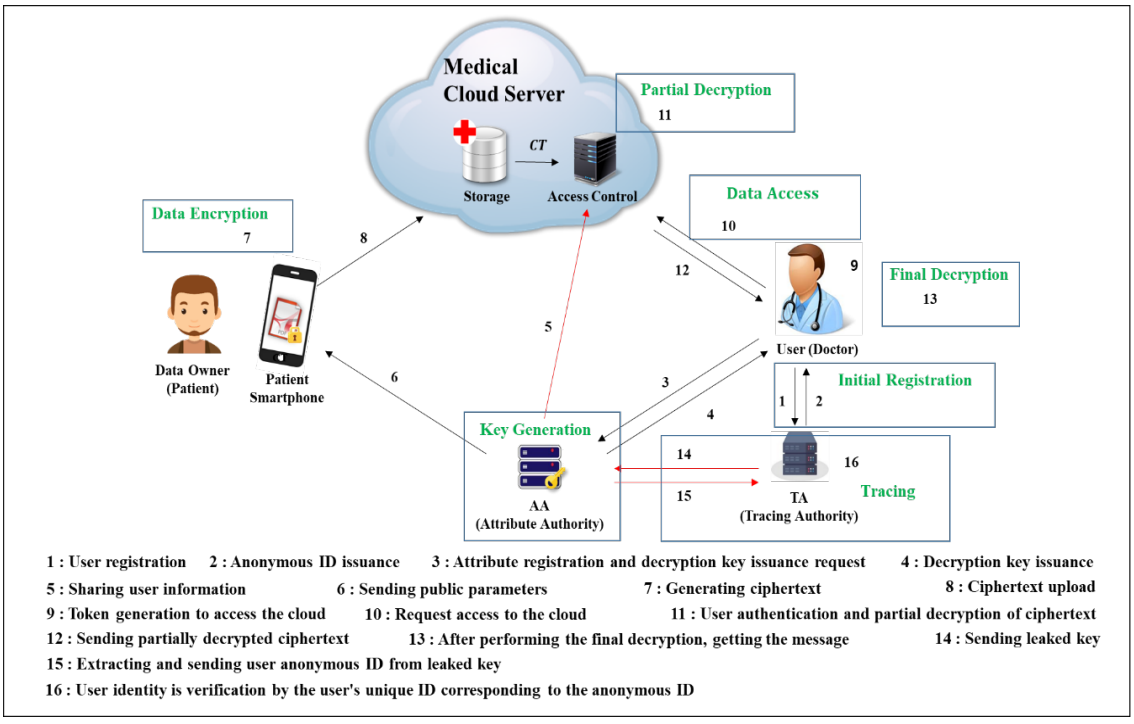

Figure 6. Overall scenario of the proposed scheme.

4.1. System Model

4.1.1. System Objects

The roles of each object in this proposed scheme are as follows.

- **Data Owner (Patient):** Data Owner is a user who uses the cloud to store encrypted data. They create an access structure based on the attributes of the users who can access their data, encrypt the data, and upload it to the cloud.
- **Medical Cloud Server:** Cloud servers consist largely of storage and access control (AC). Storage is where encrypted data is stored, and an AC server is a trusted server that supports outsourcing operations. The AC server role controls user access and processes part of the decryption operation after comparing the attributes of the access structure specified in the ciphertext with those of the user who requested the ciphertext. As a result, the decryption computations of the user is reduced, thereby increasing the efficiency of the user's decryption operation.
- **Trace Authority (TA):** A trusted server that manages user information. The user registers with the TA before the key is issued by the AA. TA generates and issues the user's anonymous ID value. In the event of a problem with a leaked key, the AA can then be used to track down and identify the first user to whom the key was issued.
- **Attribute Authority (AA):** As a semi-trusted server, it manages the user's attributes and creates a key that can decrypt in CP-ABE. A key is generated and sent to the user that allows the user to decode the ciphertext based on the user's attributes at the time of the key request. At this time, the AA generates a key with the user's attributes, so no value exists to identify the user and therefore the user cannot be identified. It can later work with the TA to trace the user who was issued the key.
- **User (Doctor or Nurse, etc.):** Users access encryption data stored on cloud storage through their attributes. The data is obtained by performing partial decryption using their attributes and final decryption using keys issued from the AA.

4.1.2. System Overall Scenario

This proposed scheme is a CP-ABE-based data sharing system in the medical cloud environment that provides user privacy protection and key abuse prevention, and ensures the integrity of the data uploaded by the data owner through a verification phase. Assuming the data owner as a patient, as shown in Figure 6, the patient creates an access structure with the attributes of the user who has access to his data, and uploads the data by encrypting it. Then, among data users (doctors, nurses, etc.) who can access the medical cloud server, only users corresponding to the attributes specified by the patient can check patient data. In the scenario of this paper, the sharing or delegation of credential (key) authority to access is not considered, and only users who have been granted access authority from AA can check the ciphertext uploaded by the data owner, assuming that the doctor has shared or delegated permission to access patient data (credential, key) with other users. In the future, problems such as leakage of patient's personal information stored in the server from the user who received the delegated key or alteration of server data may occur. At this time, the characteristics of the thesis are that the identity can be verified by tracking the user who first shared or delegated the key, and the integrity of the data can be verified in the outsourced data processing stage. Data sharing is mainly performed in a six-phase scenario, and a tracking phase is performed to trace and identify the user who was first issued the key in the case of a leaked key. Each phase proceeds as follows.

- **User registration and key issuance phase:** The user registers with the TA before receiving the key from the AA. The TA registers its own unique identifier and ID values, creates an anonymous ID_i, and sends it to the user (shown in steps 1-2 in Figure 6).
- **Setup(k):** The public key PK and master key AMK are generated by the AA by inputting the security parameter k (shown in steps 3 in Figure 6).

- **KeyGen**(**MK**, **S**, **PK**): The user sends ID_i, and attributes to the AA. The AA uses the values received from the user to generate a secret key SK capable of decrypting the ciphertext and transmits SK to the user (shown in steps 3-4 in Figure 6).
- **Encrypt**(**PK**, **M**, **AS**): The user encrypts the message with a normal symmetric key. After that, the access structure (AS) is created, and the symmetric key that encrypts the message with the PK and AS is encrypted to create a ciphertext CS. The ciphertext CT' includes the CS that encrypted the message, the CT that encrypted the key, the access structure AS, and the message verification key value VK (shown in steps 7-8 in Figure 6).
- **User data access and decryption phase**: The user creates a token to access the cloud, the AC server authenticates the user and partially decrypts the encrypted text, and the user performs final decryption. The partial decrypt phase and the final decrypt phase are as follows (shown in steps 9-13 in Figure 6).
- **Partial decrypt**(**CT**, **S**)**:** The AC server performs partial decryption when satisfied that the use attribute set matches the attribute set contained in the ciphertext. After the partial decoding process, the result C and the ciphertext CT' are sent to the user.
- **Final decrypt**(**C**, **PK**, **SK**, CT')**:** The user performs the final decryption of C and CT' received from AC using secret key SK. If the decryption is done correctly, the user can get the key that encrypted the message. The user obtains the message by decrypting the ciphertext with the key and performs the verification step to verify the integrity of the message.
- **Tracking the user who first issued the key:** It is possible to verify the identity of the user who was first issued the key by tracing the distributed key. This can solve the key abuse problem (shown in steps 14 to 16 in Figure 6).

4.2. Proposed schemes

4.2.1. System parameters

The system parameters of this proposed scheme are as follows.

- Storage: Servers that manage data
- Access Control (AC): User access control management
- Trace Authority (TA): Trusted authority that manages user information and traces leaked keys
- Attribute Authority (AA): Authority that verifies user attributes and issues keys
- MSK: master key
- PK, AMK: Public parameter and master key required for attribute-based encryption
- SK: User security key (decryption key)
- UTR: Value for tracking users
- $R\mathrm{ID}_i$, $\mathrm{ID}_{i,1}$: User real ID, user created ID
- ID_i: User anonymous ID
- T_i: Valid anonymous ID period
- ($\mathrm{T}_{pub_{TA}}$,β): TA's public and private key pairs
- ($\mathrm{T}_{pub_{AA}}$,α): AA's public and private key pairs
- A_U, A: User attribute data, A set of attribute data
- AS: Access policy
- TK: Tokens for access to the cloud
- C: Partially decrypted ciphertext
- CT, CS : Data with encrypted key, data with encrypted message
- CT' : Data with key and message encrypted ($CT' =< CT, CS >$)
- T: Timestamp

4.2.2. Assumptions

The proposed scheme assumes the following.

1. Initially, TA and AA define two large primes p, q and the elliptic curves as follows for the security parameters.

$$y^2 = x^3 + ax + b(mod\ p), a, b \in F_P \quad (5)$$

The public key and private key of TA can be represented by $T_{pub\_TA} = \beta \cdot P$, MSK = $\{\beta\}$, and the public key and private key of AA are $T_{pub\_AA} = \alpha \cdot P$, MSK = $\{\alpha\}$.

2. The AA issues a key after verifying the user attributes. At this time, part of the user identity value $(ID_i, \mathrm{Q}_{ID_i})$ is shared with the AC server.

4.2.3. User Registration and Key Issuance Phase

Step 1. Initially, the user requests registration by sending his unique identifier and ID value $(RID_i, ID_{i,1})$ to the TA as a registration message. The TA calculates $ID_{i,2}$ after checking the user's unique identifier value RID_i. After that, the user anonymous identification value ID_i is generated and transmitted to the user. Hash functions: $h_1 : \{0, 1\} * \rightarrow Z_q^*$.

$$ID_{i,2} = RID_i \oplus h_1\left(\beta \cdot ID_{i,1}, T_i, T_{pub_{TA}}\right) \quad (6)$$

$$ID_i = \{ID_{i,1}, ID_{i,2}, T_i\} \quad (7)$$

Step 2. User sends anonymous ID_i and his attribute set A_{U*} to AA. AA creates a public key master key for data owners and users through the setup process. Assume the following during the setup process: Attribute universal set is $U = [att_1, att_2, att_3, \ldots att_n]$. Each attribute has a multi-value of $V_i = \left[v_{i,1}, v_{i,2}, v_{i,ni,}\right]$ and can be represented as att_i as a set. $W = [W_1, W_2, W_3, W_n]$ is an access policy, where $W_i \subset V_i$. When the prime order of bilinear group G is p, AA generates a random constructor $\alpha, \beta, t_i \in Z_p$. After calculating $\mathrm{Y} = \mathrm{e}(g, g)^{\alpha}, \mathrm{h} = g^{\beta} \in G_0, T_i = g^{t_i}$, a public key and a master key are generated as shown.

$<Setup = PK, AMK>$

$$PK = \left\{e, g, \{T_i = g^{t_i}\}_{i\in[1,n]}, h = g^{\beta}, e(g, g)^{\alpha}\right\} \quad (8)$$

$$AMK = \{\alpha, \{t_i\}_{i\in[1,n]}\} \quad (9)$$

$$T_{pub\_AA} = \alpha \cdot P, MSK = \{\alpha\} \quad (10)$$

After that, the key SK, which uses anonymous ID_i and attribute A_{U*} transmitted from the user is generated through the Keygen process. The steps for generating the secret key SK are as follows.

$<KeyGen(PK, ID_i, AMK, A_{U*}) = SK>$

- $random\ number\ d_i \in Z_q{}^*,\ Q_{ID_i} = d_i P$
- $j \in$ S (The symbol indicates the number of each attribute as j, and the set of attributes is expressed as S)
- $\mathrm{r}_1 \ldots \mathrm{r}_j \in \mathrm{Z}_p$ (The random value given for each property is represented by the symbol r_j.)
- hash functions: $h_2 : \{0, 1\}^* \rightarrow \{0, 1\}^{l_m}$, where l_m denotes the bit length of the messages.

$$PSK_{ID_i} = d_i + h_2(ID_i, Q_{ID_i}) \cdot \alpha \quad (11)$$

$$UTR = ID_i \oplus h_2(ID_i, Q_{ID_i}) \quad (12)$$

$$\mathrm{r} = \sum_{j=1}^{n} \mathrm{r}_n \quad (13)$$

Then, calculate $\mathrm{D}' = g^{\alpha - r}$. The secret key generation is performed as follows.

$$SK = \{S, PSK_{ID_i}, UTR, D' = g^{\alpha-r}, \left\{D_{i,1} = g^{r_i}\right\}_{i\in[1,n]}\} \quad (14)$$

After issuing the secret key SK to the user, the user information (ID_i, Q_{ID_i}) is shared with the AC server (shown in Figure 7).

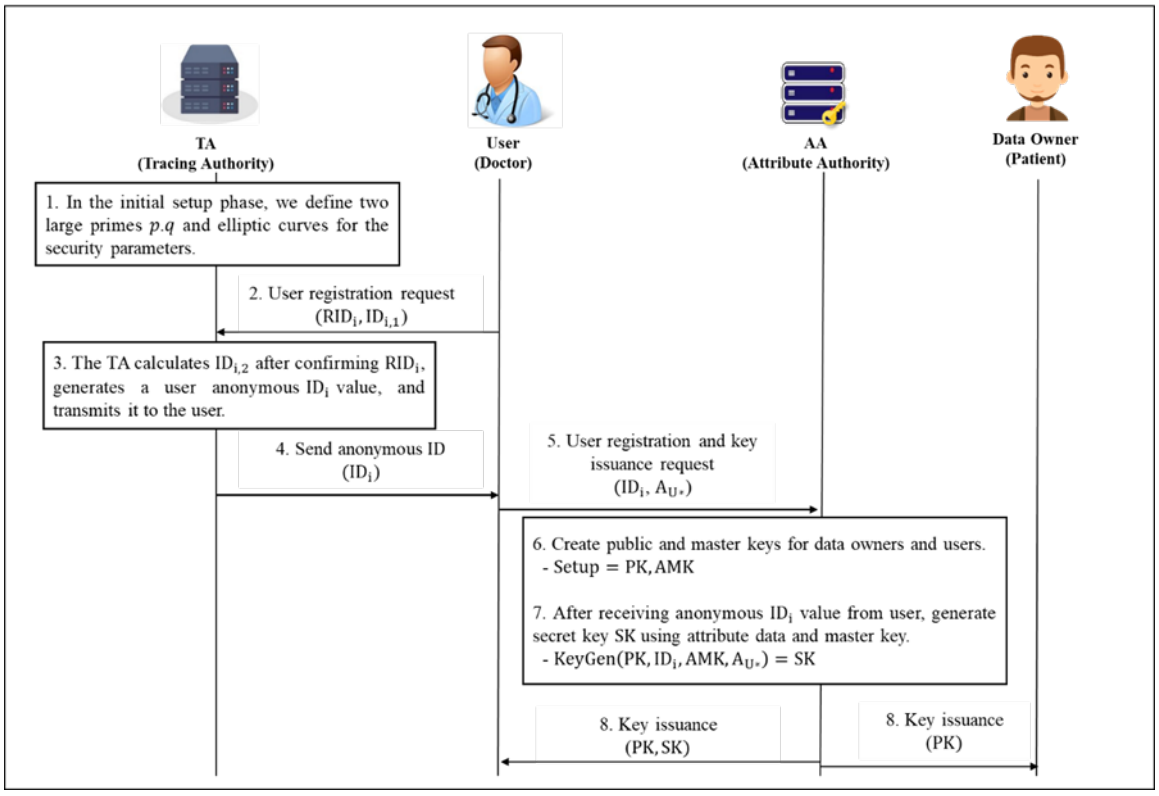

Figure 7. User Registration and Key Issuance Phase.

4.2.4. Data Encryption Phase

In this phase, the data owner encrypts the data and transmits it to the cloud storage for upload(shown in Figure 8).

Step 1. Data owner creates access structures from the user's attributes. Then, multi-values of the attributes contained in the access structure are then calculated according to the conditions, generating a ciphertext CT. Ciphertext CT' is composed of ciphertext CT that encrypts the KEY that can decrypt the message, a CS that encrypts the message, and the verification value VK for the KEY. By calculating the computed values for each attribute in the ciphertext CT, the size of the ciphertext can be reduced, and thus the space of the previously wasted cloud storage can be efficiently used.

$< Encrypt(\mathrm{PK}, \mathrm{M}, \mathrm{AS}) = CT' >$

- $\mathrm{M} \in \mathrm{G}_T$ and access policy $W = [W_1, W_2, W_3, W_n]$.
- Random value generation $s \in \mathrm{Z}_p$ such that

$$\mathrm{s} = \sum_{j=1}^{n} \mathrm{s}_j \tag{15}$$

Then, it is calculated as follows.

$$\tilde{C}: Y^s = \mathrm{KEY} \cdot \mathrm{e}(g, g)^{\alpha s}, \check{C}: g^s, \bar{C}: h^s, \tag{16}$$

If $v_{i,1} \in W_i$, computes

$$\mathrm{C}_i = g^{t_n s} \tag{17}$$

If $v_{i,1} \notin W_i$, computes

$$\mathrm{C}_i = g^{t_{n+1} s} \tag{18}$$

$$\mathrm{CT} =< \tilde{\mathrm{C}}, \check{\mathrm{C}}, \bar{\mathrm{C}}, \mathrm{C}' > \tag{19}$$

$$C' = (h \cdot \prod_{i \in AS} C_i)^s = (h \cdot \prod_{i \in AS} g^{t_i})^s, g^t = \prod_{i=1}^{n} g^{t_i} \quad (10)$$

$$CT = < AS, \tilde{C}, \check{C}, \bar{C}, C' > \quad (11)$$

$$CS = Enc_{KEY}(M), VK = (g^{h(KEY)}, g^{h(M)}) \quad (12)$$

$$CT' = < CT, CS, VK > \quad (13)$$

The cloud storage safely stores the ciphertext CT received from the user.

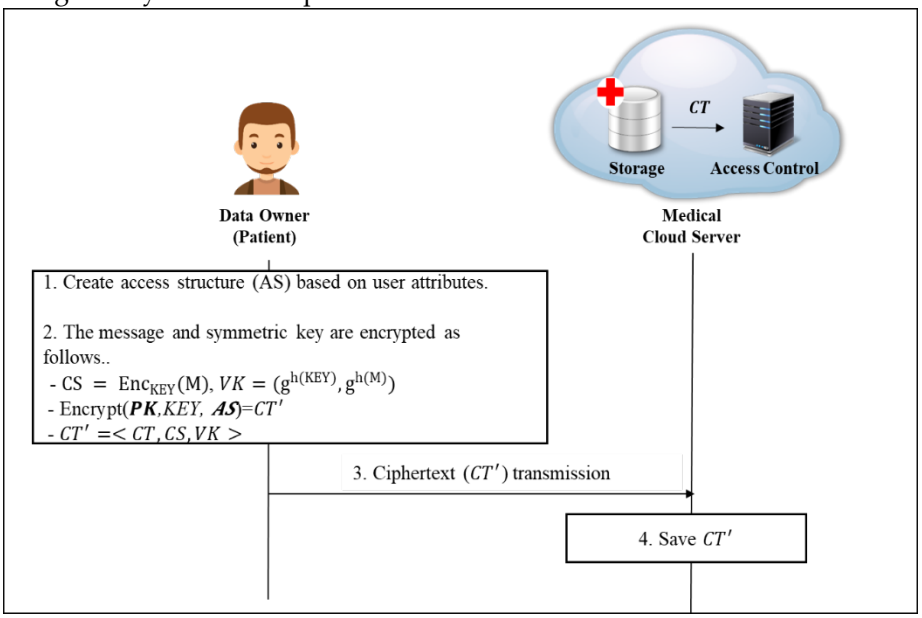

Figure 8. Data Encryption Phase.

4.2.5. User Data Access and Data Decryption Phase

In this phase, the user accesses the cloud, receives the ciphertext, and then decrypts it. The decryption step consists of two steps: partial decoding on the AC server and final decoding by the user (shown in Figure 9).

Step 1. The user creates a TK that can access the cloud through the PSK_{ID_i} from the secret key SK received from AA. Then, when requesting access to the cloud via $ID_i||TK||T$, the AC server verifies (ID_i, Q_{ID_i}) of the TK value through the PSK_{ID_i} user information values shared by AA and determines that it is a legitimate user.

$$\text{Token (TK)} = PSK_{ID_i}||A_{U*} \quad (14)$$

Since the AC server knows the value of (ID_i, Q_{ID_i}) for the user, it verifies the user by performing the following verification process

$$PSK_{ID_i} \cdot P = Q_{ID_i} + h_2(ID_i, Q_{ID_i}) \cdot T_{pub\_AA} \quad (15)$$

When it is verified that the user is a legitimate user, a partial decryption is performed after requesting the ciphertext desired by the user from the storage and comparing the access structure and user attribute values specified in the ciphertext.

$<Partial\ decrypt(CT, A_{U*}, x) = C>$

The process of performing partial decryption with ciphertext and user attributes is as follows.

$$C = \frac{e(g^r, C')}{e(\check{C}, (\prod_{j \in S} g^{t_i})^s)} = e(g, g)^{rs} \quad (16)$$

Step 2. The AC server transmits the result C and the ciphertext CT' obtained by performing partial decryption to the user.

Step 3. The user performs the final decryption of the CT' received from the AC server with SK, C, and PK, and extracts a key that can decrypt the message. Thereafter, after verifying whether the existing message can be decrypted, the ciphertext CS is finally decrypted.

<$Final\ decrypt$(PK, SK, C, CT') = M>

The user performs the final decryption to obtain a key capable of acquiring the message M.

$$KEY' = \frac{\tilde{C}}{e(\check{C}, \mathrm{D}') \cdot \mathrm{C}} = \frac{\mathrm{KEY} \cdot \mathrm{e}(g,g)^{\alpha s}}{e(g^s, g^{\alpha - r}) \cdot \mathrm{e}(g,g)^{rs}} \quad (27)$$

$$M' = Dec_{KEY'}(CS) \quad (28)$$

Create VK′ with M' and KEY'. After that, the verification process is performed by comparison to the existing VK and the integrity of the message is confirmed.

$$\mathrm{VK} = \mathrm{VK}' = (g^{\mathrm{h(KEY)}}, g^{\mathrm{h(M)}}) = g^{\mathrm{h}(KEY')}, g^{\mathrm{h}(M')} \quad (2917)$$

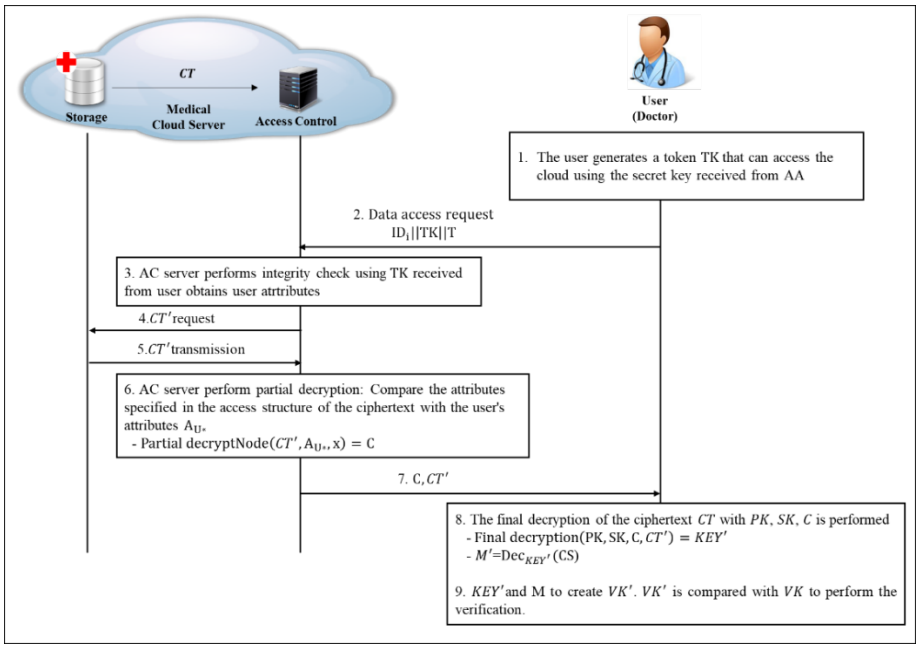

Figure 9. User Data Access and Data Decryption Phase.

4.2.6. Tracing Phase

This step is intended to prevent unauthorized third parties from accessing the cloud by receiving a key from someone. At this time, the investigating agency cannot know the process of distributing the secret key SK, but it can know the identity of the user who received and distributed the key for the first time through the tracking process. Here, we assume that $SK = \{S, PSK_{ID_i}, UTR, D' = g^{\alpha - r}, \{D_{i,1} = g^{r_i}\}_{i \in [1,n]}\}$ has been distributed and we proceed to the first step (shown in Figure 10).

Step 1. AA verifies PSK_{ID_i} in the parameters of the distributed key and extracts the user's anonymous ID value ID_i. Thereafter, the user's anonymous ID value ID_i is transmitted to the tracing authority agency TA.

$$PSK_{ID_i} \cdot P = Q_{ID_i} + h_2(ID_i, Q_{ID_i}) \cdot T_{pub\_AA} \text{ (Verification complete)} \quad (3018)$$

$$ID_i = UTR \oplus PSK_{ID_i} \quad (3119)$$

Step 2. The TA extracts the corresponding unique ID value RID_i after receiving the user's anonymous ID value ID_i. Afterwards, the identity of the user who issued the key for RID_i is confirmed.

$$ID_i = \{ID_{i,1}, ID_{i,2}, T_i\} \quad (32)$$

$$RID_i = ID_{i,2} \oplus h_1(\beta \cdot ID_{i,1}, T_i, T_{pub\_TA}) \quad (330)$$

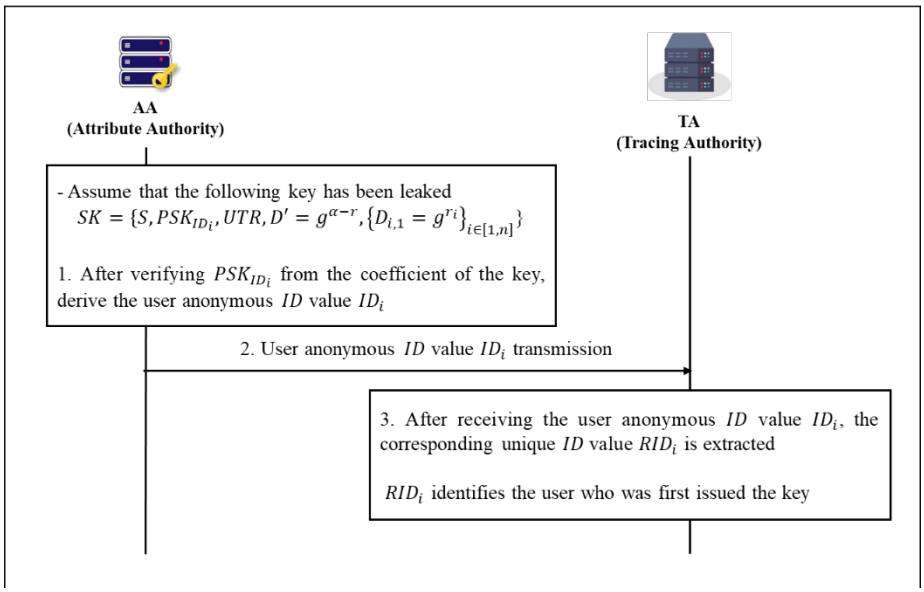

Figure 10. Tracing the User Who First Issued the Key Phase.

5. Analysis of Proposed Scheme

This proposed scheme satisfies the security requirements give in Section 3. In 5.1, the security requirements of this proposed method were analyzed, and in 5.2, the efficiency of operation was analyzed (shown in Table 3 and Table 4).

5.1. Security Analysis

Table 3. Comparison of security requirements of this proposed scheme and the existing CP-ABE.

	Hahn Scheme [12]	Jiang Scheme [12]	Yu Scheme [22]	Qi Li Scheme [5]	Premkamal Scheme [17]	Jiguo Li Scheme [18]	Proposed Scheme
Collusion/ masquerade attack	Safe						
Data Storage space	Efficient	Inefficient				Efficient	
Ciphertext length	Constant size	Proportional to the number of attributes				Constant size	
User privacy	Infringement impossible	Infringement Possible (Trusted servers manage user information through identity tables)				Infringement impossible	

Tracing the first user to distribute the key	Untraceable	Traceable via identity table	Traceable with user signatures stored in identity tables	Traceable via identity table	Untraceable	TA and AA work together to track who was issued the first key
Verifying the integrity of the data owner's message	Not considered			Message verification with VK, which can verify ciphertext	Verify by inserting MAC authentication code in ciphertext	Verify by inserting the message hash value in the ciphertext
Outsourcing operation support	Supported	Not supported		Supported		

- **Collusion attack:** In the proposed scheme, AA uses nonce value in addition to the attribute values when generating the secret key, so even if the user determined the attributes through a collusion attack, the secret key can be generated. In addition, in the part where the partial decryption is performed by comparing the attributes of the user and the attributes of the access policy specified in the ciphertext on the AC server, the message M cannot be viewed because the user's secret key is not known. Therefore, it is safe from collusion attack between users or between users and service providers.
- **Authenticated user access control:** In the proposed scheme, only the user who generated the token by receiving the secret key from AA can access the data. In addition, only users who satisfy the attributes of the access structure specified by the data owner can access the data stored in the cloud. The user's access is primarily blocked by the AC server, and when the above conditions are satisfied partial decryption is performed and the ciphertext and the result of partial decryption are transmitted to the user. Therefore, the proposed scheme provides confidentiality and integrity to stored data because only authenticated users can access it.
- **Tracing the user first issued an abused key:** The existing scheme for providing traceability shown in Table 1 verify the leaked key via a Key Sanity Check, and after verification, the user's identification information contained in the key is verified and the AA identifies the user who was issued the key for the first time. Our proposed scheme also includes a value PSK_{ID_i}, UTR that can identify a user from SK. The difference with the schemes presented in Table 1 is the aspect of user privacy protection. Existing schemes manage the user's information in AA because the key must include information that can identify the user when issuing the key. This can violate the user's privacy, excluding the user's anonymity, which is provided by default in attribute-based encryption. Therefore, in this proposed scheme, an entity that issues an anonymous ID value called a TA is provided, and the user is provided with an anonymous ID value when registering so that the CP-ABE data sharing system can be utilized anonymously. In the event a key is misused, the AA and TA can cooperate with each other to track and identify the user who was first issued the key, and user privacy can be protected.
- **Verify data integrity:** In our proposed scheme, the attribute-based encryption/decryption is not the message but the message encryption key, unlike existing schemes. Therefore, in the partial decoding process in the outsourcing server, the message is not converted. In addition, in the proposed ciphertext CT' = <CT, CS, VK>, there is a verification key VK that can verify the integrity of the message of the data owner. Accordingly, the user verifies through the KEY' and M' obtained after the final decoding $\mathrm{VK} = \mathrm{VK}' = (g^{\mathrm{h(KEY)}}, g^{\mathrm{h(M)}}) = g^{\mathrm{h}(KEY')}, g^{\mathrm{h}(M')}$ to verify the integrity of the message. Compared to the existing Yu and Jiguo Li schemes that support verifiable outsourcing, the verification of message integrity is relatively simple.

Table 4. Comparing the computation amount of this proposed scheme with existing CP-ABE schemes.

	Hahn Scheme [13]	Jiang Scheme [12]	Yu Scheme [22]	Qi Li Scheme [5]
Encryption	$c_e + (n + 4)M$	$c_e + M + (n + 1)E$	$2c_e + (n + 3)M + (4n + 6)E$	$c_e + (4n + 5)E + (2n + 2)M + 2H + 1Enc$
Partial decryption (server)	$2c_e + 3E + 2nM$	-	-	$4nc_e + (2n + 3)M + nE$
Final decryption (user)	$(2n + 3)c_e + (2n + 3)M$	$2nc_e + (n + 1)M$	$(2n + 3)c_e + (2n + 3)M + (2n + 1)E$	E+M+H+1Dec
Message verification	-	-	-	2H
	Premkamal Scheme [17]		Jiguo Li Scheme [18]	Proposed Scheme
Encryption	$2(n + 1)c_e + (2n + 1)E + H + M + 1Enc$		$2c_e + 2(n + 1)M + 2(n + 4)E + 2H$	$c_e + (n + 1)M + (n + 5)E + 2H + 1Enc$
Partial decryption (server)	$(n + 3)c_e + nE$		$4c_e$+2M	$2c_e + nM + 2E$
Final decryption (user)	1Dec		$2H + 4E + 4M$	$c_e + 2M + 2H + 1Dec$
Message verification	E+3M+H		$2H + 2E + 2M$	2H

c_e : *Pairing operation*; M : *Multiplication operation*; n: *Number of attributes*; E: *Exponentiation operation*; H: *Hash function* *Enc*: *Symmetric key encryption*; *Dec*: *Symmetric key decryption*

5.2. Efficiency

The throughput experiments in Figure 11, 12 were performed on a Windows system with a 3.50GHz Intel Core i5-4690 processor and 8GB of RAM. For the pairing operation, see Pairing Based Crypto Library (Lynn, B., "The pairing-based cryptography (PBC) library," Available: http://crypto.stanford.edu/pbc, 2012.). The symmetric key cipher used for comparison measurement is AES-128, and the parameter of bit operation is 128 bits. When performing encryption/decryption once with a symmetric key, the message length was based on 1000. In addition, because the operation speed is very small for multiplication operations and exponentiation operations, it is expressed as 0.001. In Figure 11, 12, the values are expressed on a millisecond basis.

- **Cloud storage space efficiency:** In existing CP-ABE methods, when the ciphertext is generated, the size of the ciphertext increases in proportion to the number of attributes specified in the access policy, thus wasting storage space. In particular, in the Qi Li scheme, the size of the ciphertext $\mathrm{CT} < \mathrm{T}, \mathrm{C}, C_1, C_2, \{C_{i,1}, C_{i,2}, C_{i,3}\}_{i\in[\mathrm{m}]} >$ increases with the number of attributes. In the proposed scheme, the number of attributes specified in the access policy is represented as a single number by performing a separate operation $\mathrm{C}' = (\mathrm{h} \cdot \prod_{i\in AS} C_i)^s = (\mathrm{h} \cdot \prod_{i\in AS} g^{t_i})^s$, which results in a constant-size ciphertext. As a result, as opposed to existing CP-ABE methods, the proposed scheme generates a ciphertext of a certain size, not proportional to the number of attributes specified when generating the ciphertext. The wasted cloud storage space can be used efficiently. However, only the size of the ciphertext is constant. The amount of computation required for encryption varies depending on the number of attributes.
- **Efficiency of computation:** In conventional CP-ABE methods, when decrypting the user receives the ciphertext and decrypts it. Accordingly, the amount of computation for the use is proportional to the size of the ciphertext. This proposed scheme performs partial decoding by including an AC server that supports outsourcing. As the result of partial decryption by the AC server, the user receives the C and the ciphertext CT' and proceeds to the final decryption, so the message M can be obtained. By using the AC server for outsourcing, the computational efficiency can be increased by reducing the amount of computation required for the user. As shown in Table 4, this proposed scheme has more user decryption computations than the Qi Li and Premkamal, Jiguo Li schemes among the CP-ABE methods that support an outsourcing

server. This is because most of the decryption operations are performed on the server, so the more users there are, the more computations the server processes. However, in the Premkamal scheme and Jiguo Li scheme, when the user distributes the key to someone, the distributed key can be used for access by unauthorized users. At this time, if the distributed key is misused, the user who originally issued the key cannot be tracked. In other words, the aforementioned key abuse problem arises. Traceability is provided in the qi Li system, but personal privacy issues may arise because the server that issued the key knows the user's information. Our proposed scheme solves the problem of key abuse by tracking the user who was issued the key for the first time through the leaked key and provides user anonymity.

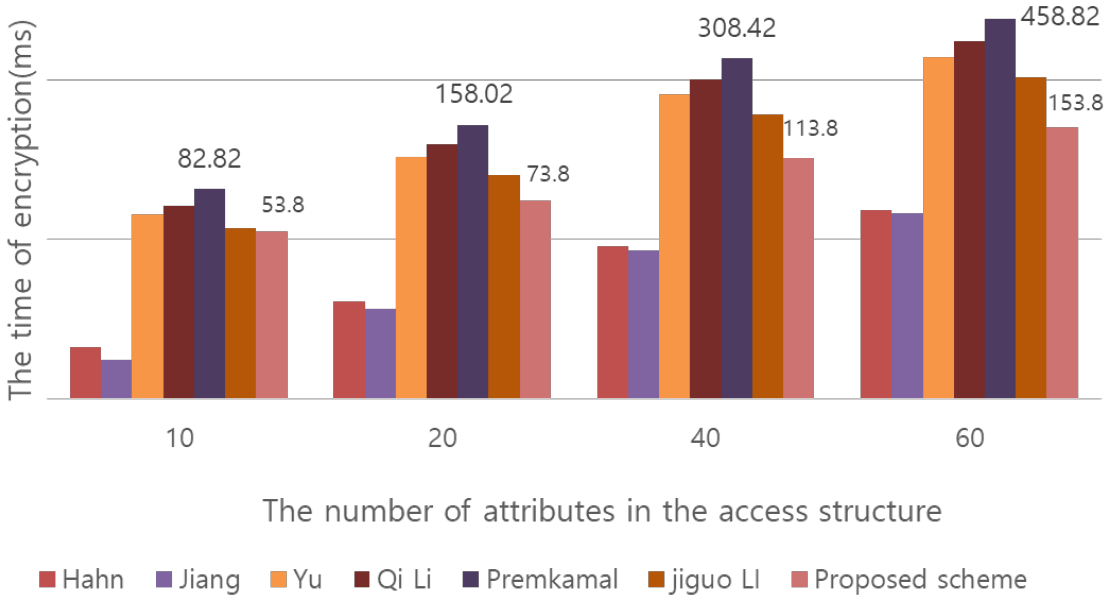

Figure 11. Comparison of data encryption time between existing CP-ABE schemes and the proposed scheme.

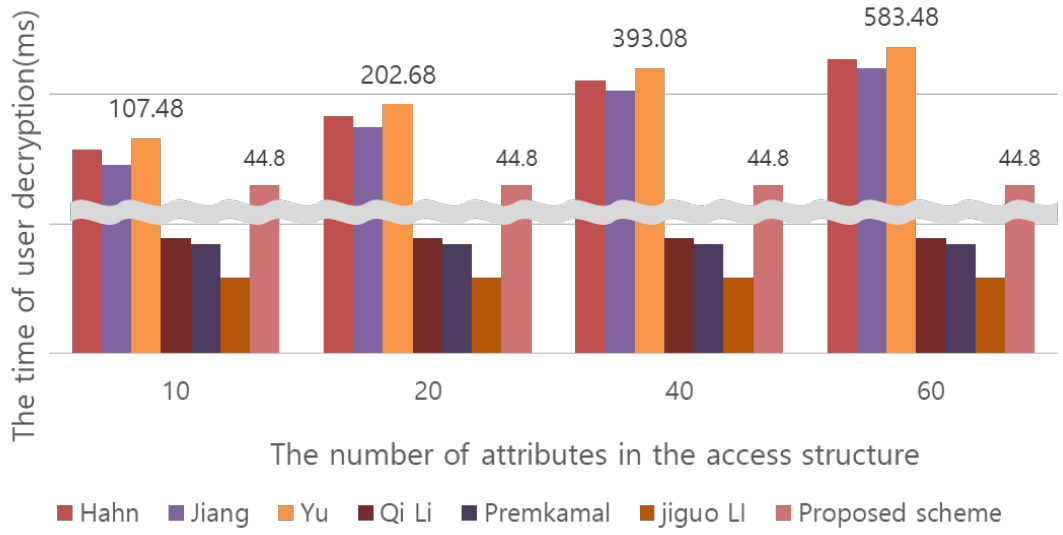

Figure 12. Comparison of data decryption time between existing CP-ABE schemes and the proposed scheme.

6. Conclusions

This proposed scheme is a CP-ABE-based medical data sharing system that supports major exploit prevention and outsourcing operations and allows medical data to be safely and efficiently shared in the cloud in IoMT environments. It provides traceability by tracking the user who was issued a key through the distributed key, but preserves the user's anonymity because the key is issued by the AA using the user's anonymous ID value. In addition, it is safe against various security threats such as masquerade attacks and collusion attacks, and unauthorized third party access is not possible. In terms of efficiency, the ciphertext size of the existing CP-ABE scheme was proportional to the number of attributes. However, this proposed approach effectively removes the wasted space of existing cloud storage because the size of the ciphertext is not proportional to the number of attributes, like $C' = (h \cdot \prod_{i \in AS} C_i)^s = (h \cdot \prod_{i \in AS} g^{t_i})^s$. In addition, the calculation efficiency is higher than the Hahn scheme, Jiang scheme, and Yu scheme CP-ABE methods, which do not provide outsourcing of existing ciphertext and outputs of a certain size in the process of user decrypting ciphertext. The message verification operation is also 2H, which is more efficient than other methods. Lastly, when decrypting a ciphertext on the user's side, since some of the decryption operations are supported by the AC server, the computational efficiency for users who are burdened with the decryption process may be increased due to lack of computing resources.

In the future research, this proposed scheme has a lot of computation to verify the user's identity through the key compared to the existing schemes. Therefore, a lightweight CP-ABE scheme is needed in order to increase the efficiency of computation. Additionally a key escrow problem may occur in the AA, so research is needed to solve it.

Author Contributions: Conceptualization, Y.W.H., I.Y.L.; data investigation, Y.W.H.; analysis and validation, Y.W.H., I.Y.L.; writing—original draft, Y.W.H.; writing—review and editing, Y.W.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2019R1A2C1085718) and was supported by the Soonchunhyang University Research Fund.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. John, B.; Sahai, A.; Waters, B. Ciphertext-policy attribute-based encryption. *IEEE Symp. Secur. Priv.* **2007**, doi:10.1109/SP.2007.11.
2. Ling, C.; Calvin, N. Provably secure ciphertext policy ABE. In Proceedings of the 14th ACM Conference on Computer and Communications Security, ACM, Alexandria, VA, USA, 29 October–22 November 2007; pp. 456–465.
3. Sekhar, B.R.; Kumar, B.S.; Reddy, L.S.; Poorna Chandar, V. CP-ABE based encryption for secured cloud storage access. *Int. J. Sci. Eng. Res.* **2012**, *3*, 1–5.
4. Zhu, S.; Yang, X. Protecting data in cloud environment with attribute-based encryption. *Int. J. Grid Util. Comput.* **2015**, *6*, 91–97.
5. Qi, L.; Zhu, H.; Ying, Z.; Zhang, T. Traceable ciphertext-policy attribute-based encryption with verifiable outsourced decryption in ehealth cloud. *Wirel. Commun. Mob. Comput.* **2018**, *2018*, 1701675.
6. Zhen, L.; Cao, Z.; Wong, D.S. White-box traceable ciphertext-policy attribute-based encryption supporting any monotone access structures. *IEEE Trans. Inf. Forensics Secur.* **2012**, *8*, 76–88.
7. Changhee, H.; Kwon, H.; Hur, J. Efficient attribute-based secure data sharing with hidden policies and traceability in mobile health networks. *Mobile Inf. Syst.* **2016**, *2016*, 6545873.
1. Yu, G.; Cao, Z.; Zeng, G.; Han, W. Accountable ciphertext-policy attribute-based encryption scheme supporting public verifiability and nonrepudiation. In *International Conference on Provable Security*; Springer: Berlin/Heidelberg, Germany, 2016.
8. Zhang, R.; Hui, L.; Yiu, S.; Yu, X.; Liu, Z.; Jiang, Z.L. A traceable outsourcing cp-abe scheme with attribute revocation. In Proceedings of the 2017 IEEE Trustcom/BigDataSE/ICESS, IEEE, Sydney, NSW, Australia, 1–4 August 2017.
9. Yang, Y.; Liu, X.; Deng, R.H.; Li, Y. Lightweight sharable and traceable secure mobile health system. *IEEE Trans. Dependable Secur. Comput.* **2017**, *17*, 78–91.

10. Luo, E.; Meng, D.; Wang, W.; Luo, E.; Wang, G. Attribute-Based Traceable Anonymous Proxy Signature Strategy for Mobile Healthcare. In *International Conference on Information Security Practice and Experience;* Springer: Berlin/Heidelberg, Germany, 2018.
11. Jiang, Y.; Susilo, W.; Mu, Y.; Guo, F. Ciphertext-policy attribute-based encryption with key-delegation abuse resistance. *Australasian Conference on Information Security and Privacy*; Springer: Berlin/Heidelberg, Germany, 2016, pp. 477–494.
12. Hahn, C.; Hur, J. Constant-size Ciphertext-policy Attribute-Based Data Access and Outsourceable Decryption Scheme. *J. KIISE* **2016**, *43*, 933–945, doi:10.5626/jok.2016.43.8.933.
13. Teng, W.; Yang, G.; Xiang, Y.; Zhang, T.; Wang, D. Attribute-based access control with constant-size ciphertext in cloud computing. *IEEE Trans. Cloud Comput.* **2017**, *5*, 617–627, doi:10.1109/TCC.2015.2440247.
14. Nurmamat, H.; Rahman, K. CP-ABE access control scheme for sensitive data set constraint with hidden access policy and constraint policy. *Secur. Commun. Netw.* **2017**, *2017*, doi:10.1155/2017/2713595.
15. Lai, J.; Deng, R.H.; Guan, C.; Weng, J. Attribute-based encryption with verifiable outsourced decryption. *IEEE Trans. Inf. Forensics Secur.* **2013**, *8*, 1343–1354, doi:10.1109/TIFS.2013.2271848.
16. Kumar, P.P.; Pasupuleti, S.K.; Alphonse, P.J.A. A new verifiable outsourced ciphertext-policy attribute based encryption for big data privacy and access control in cloud. *J. Ambient. Intell. Humaniz. Comput.* **2019**, *10*, 2693–2707.
17. Qin, B.; Deng, R.H.; Liu, S.; Ma, S. Attribute-based encryption with efficient verifiable outsourced decryption. *IEEE Trans. Inf. Forensics Secur.* **2015**, *10*, 1384–1393, doi:10.1109/TIFS.2015.2410137.
18. Hongwei, L.; Wang, X.; Zhang, P. Verifying Outsourced Decryption of CP-ABE with Signature. Proceedings of the 4th International Conference on Mechatronics, Materials, Chemistry and Computer Engineering 2015. Atlantis Press: Paris, France, 2015.
19. Li, J.; Sha, F.; Zhang, Y.; Huang, X.; Shen, J. Verifiable outsourced decryption of attribute-based encryption with constant ciphertext length. *Secur. Commun. Netw.* **2017**, doi:10.1155/2017/3596205.
20. Li, Z.; Li, W.; Jin, Z.; Zhang, H.; Wen, Q. An Efficient ABE Scheme With Verifiable Outsourced Encryption and Decryption. *IEEE Access* **2019**, *7*, 29023–29037, doi:10.1109/ACCESS.2018.2890565.
21. Yang, Y.; Liu, X.; Zheng, X.; Rong, C.; Guo, W. Efficient traceable authorization search system for secure cloud storage. *IEEE Trans. Cloud Comput.* **2018**, doi:10.1109/TCC.2018.2820714.
22. Shen, J.; Zhou, T.; Chen, X.; Li, J.; Susilo, W. Anonymous and traceable group data sharing in cloud computing. *IEEE Trans. Inf. Forensics Secur.* **2017**, *13*, 912–925.
23. Yu, G.; Wang, Y.; Cao, Z.; Lin, J.; Wang, X. Traceable and undeniable ciphertext-policy attribute-based encryption for cloud storage service. *Int. J. Distrib. Sens. Netw.* **2019**, *15*, 1550147719841276.
24. Liu, J.; Hu, Q.; Li, C.; Sun, R.; Du, X.; Guizani, M. A Traceable Concurrent Data Anonymous Transmission Scheme for Heterogeneous VANETs. 2018 IEEE Global Communications Conference (GLOBECOM), IEEE, Abu Dhabi, United Arab Emirates, 9–13 December 2018.
25. Zhou, Z.; Huang, D. On efficient ciphertext-policy attribute based encryption and broadcast encryption. In Proceedings of the 17th ACM Conference on Computer and Communications Security, Chicago IL, USA, 4–8 October 2010; pp. 753–755.
26. Canetti, R.; Halevi, S.; Katz, J. Chosen Ciphertext Security from Identity Based Encryption. In *Advances in Cryptology—Eurocrypt, Volume 3027 of LNCS*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 207–222.
27. Boneh, D.; Boyen, X. Efficient Selective-ID Secure Identity Based Encryption without Random Oracles. In *Advances in Cryptology—Eurocrypt, Volume 3027 of LNCS*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 223–238.
28. Rohit, A.; Mohanty, S.K.; Sakurai, K. A Traceable Signcryption Scheme for Secure Sharing of Data in Cloud Storage. In Proceedings of the 2016 IEEE International Conference on Computer and Information Technology (CIT), IEEE, Nadi, Fiji, 7–10 December 2016.
29. Fan, K.; Wang, J.; Wang, X.; Li, H.; Yang, Y. A secure and verifiable outsourced access control scheme in fog-cloud computing. *Sensors* **2017**, *17*, 1695.
30. Liu, Z.; Jiang, Z.L.; Wang, X.; Yiu, S.M. Practical attribute-based encryption: Outsourcing decryption, attribute revocation and policy updating. *J. Netw. Comput. Appl.* **2018**, *108*, 112–123.