The ISC Int'l Journal of Information Security

January 2024, Volume 16, Number 1 (pp. 93–114)

http://www.isecure-journal.org

HUAP: Practical Attribute-Based Access Control Supporting Hidden Updatable Access Policies for Resource-Constrained Devices

Mostafa Chegenizadeh [1,*], Mohammad Ali [2], Javad Mohajeri [3], and Mohammad Reza Aref [4]

[1] *Department of Informatics, University of Zurich, Switzerland*
[2] *Department of Mathematics and Computer Science, Amirkabir University of Technology, Tehran, Iran*
[3] *Electronic Research Institute, Sharif University of Technology, Tehran, Iran*
[4] *Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran*

ARTICLE INFO.

Article history:
Received: April 29, 2023
Revised: July 31, 2023
Accepted: November 21, 2023
Published Online: November 30, 2023

Keywords:
Access Policy Update, Anonymous Attribute-Based Encryption, Blind Access Policy, Cloud Computing, Fast Decryption, Online/Offline Encryption

Type: Research Article

doi: 10.22042/isecure.2023.395028.954

dor: 20.1001.1.20082045.2024.16.1.6.0

ABSTRACT

Attribute-based encryption (ABE) is a promising cryptographic mechanism for providing confidentiality and fine-grained access control in the cloud-based area. However, due to high computational overhead, common ABE schemes are not suitable for resource-constrained devices. Additionally, access policies should be able to be updated efficiently by data owners, and in some circumstances, hidden access policies are necessary to preserve the privacy of clients and data. In this paper, we propose a ciphertext-policy attribute-based access control scheme that, for the first time, simultaneously provides online/offline encryption, hidden access policy, and access policy update. In our scheme, resource-constrained devices are equipped with online/offline encryption reducing the encryption overhead significantly. Furthermore, attributes of access policies are hidden such that the attribute sets satisfying an access policy cannot be guessed by other parties. Moreover, data owners can update their defined access policies while outsourcing a major part of the updating process to the cloud service provider. In particular, we introduce blind access policies that enable the cloud service provider to update the data owners' access policies without receiving a new re-encryption key. Besides, our scheme supports fast decryption such that the decryption algorithm consists of a constant number of bilinear pairing operations. The proposed scheme is proven to be secure in the random oracle model and under the hardness of Decisional Bilinear Diffie–Hellman (DBDH) and Decision Linear (D-Linear) assumptions. Also, performance analysis results demonstrate that the proposed scheme is efficient and practical.

1 Introduction

With the rapidly increasing number of cloud-based services, the need for methods to provide data secrecy and user privacy grows significantly [1]. Cloud computing technology enables data owners to out-

* Corresponding author.
Email addresses: mostafa.chegenizadeh@alum.sharif.edu, mali71@aut.ac.ir, mohajer@sharif.ir, aref@sharif.ir
ISSN: 2008-2045

source their private data to a cloud service provider and define an access policy preventing unauthorized parties from accessing their data.

Attribute-based encryption (ABE) [2] offers access control for protecting information within the cloud computing environment. ABEs are divided into two primary categories key-policy ABE (KP-ABE) [3] and ciphertext-policy ABE (CP-ABE) [4]. In a KP-ABE scheme, access rights of users are determined by a trusted third party, and ciphertexts are labeled by some attributes. A user can decrypt a ciphertext if and only if the attributes of the ciphertext satisfy the user's access right. However, in a CP-ABE scheme, access rights of users are specified according to their attributes, and each ciphertext is associated with an access policy such that only users whose attributes satisfy the access policy can recover the associated message [5]. As in CP-ABE data owners can determine the privileges of authorized users, it is more suitable for real cloud-based applications like smart health (s-health) [6].

Although CP-ABE brings great benefits, there are also some main challenges. Firstly, in traditional CP-ABE, access policies are stored in a clear-text form. As access policies consist of authorized users' attributes, revealing the access policies may leak some sensitive information about the associated data or the associated recipients. Anonymous ABE (A-ABE) schemes [6–13] alleviate this problem by affording hidden access policies. Indeed, in these schemes, no party can obtain any information about the authorized users' attributes.

Secondly, in many situations, data owners need to update their defined access policies, revoke the access right of some data users (policy deletion), or grant some new access privileges to some other users (policy addition). The revoked users must be unable to extract the underlying values that are encrypted under the new access policies. An obvious solution to this problem is to decrypt and then re-encrypt the data. However, it is clearly impractical for large amounts of data. To efficiently address the problem, data owners should be able to outsource the updating process to a proxy server. However, in traditional ciphertext-policy attribute-based proxy re-encryption (CP-ABPRE) schemes [14–18], data owners have to generate some re-encryption keys whenever they need a policy update. As a result, the growing number of ciphertexts as well as the rising number of access policy updates makes the updating process inefficient [19]. Moreover, as we know, the existing CP-ABPRE schemes require that the data owner be online to generate the re-encryption key, while the data owner may not be available when the access policy update is needed, for example, due to limited network bandwidth or limited computational power [20]. Therefore, the process of access policy update should be feasible even when the data owner is offline.

Thirdly, in many existing applications like s-health, data owners usually use resource-limited devices for encrypting and sending data to the cloud service provider. Therefore, the data owners have trouble in completing the whole computations of the encryption algorithm [21]. Online/offline encryption mechanism [22–25] is a promising solution to this problem. In this setting, the encryption process is divided into two phases: the offline phase and the online phase. In the offline phase, the device can access enough power resources and has enough time to generate some offline ciphertexts while messages are not known. In the online phase, while the device can access limited power and computational resources, once a message is known, the device uses a pre-computed offline ciphertext to obtain an online ciphertext in a short period of time [26].

Fourthly, in traditional CP-ABE, the same entity collects data and also defines access policies. However, in reality, there may be several devices that collect data while another party defines the access policy. Directly adopting traditional CP-ABE in such a situation requires that all of the data collector devices be aware of the current defined access policy and encrypt data according to it. Therefore, whenever the corresponding data owner wants to define a new access policy over the data, the encryption algorithm running by these devices needs to be updated, while re-programming these devices is difficult in some applications like s-health, and hence changing the encryption algorithm is not feasible [27]. Moreover, by adopting traditional CP-ABE, all of the data collector devices should share a similar set of secret parameters. Therefore, revealing the secret parameters of each of these devices threatens the security of all the others.

To make sense, consider the following s-health scenario in which simultaneously resolving all of the above issues is necessary. Main entities in a Body Sensor Network (BSN) are shown in Figure 1. In a BSN, there are several resource-constrained sensors that collect health data from a patient's body, where each sensor collects a specific kind of data such as blood pressure, blood oxygen level, heart rate, respiratory rate, body temperature, etc. Each sensor encrypts its collected data independently and then outsources the encrypted data to the cloud through a gateway. At the other side, the data owner connects to the cloud and defines an access policy for the whole of the outsourced health data. In this case, the following security and performance requirements should be

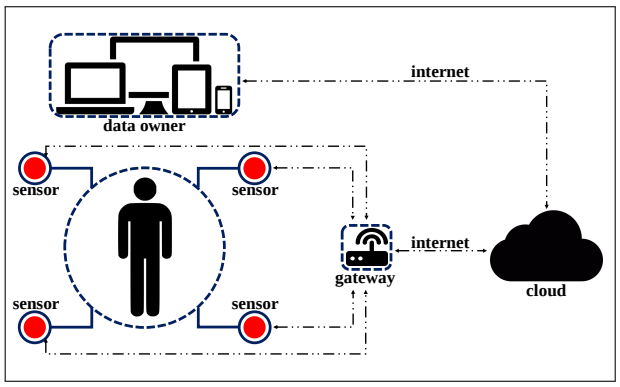


Figure 1. Body Sensor Network.

fulfilled:

a) To preserve the attribute privacy of authorized data users, the attribute sets satisfying the access policy should be hidden.
b) The data owner should be able to update the defined access policy efficiently.
c) The computational overhead on the resource-constrained sensors should be as low as possible.
d) The sensors should encrypt the collected data independent of the defined access policy. Therefore, updating the access policy should not change the performance of these sensors. Moreover, The sensors should work independently such that revealing secret parameters of a sensor does not threaten the security of the data collected by the other sensors.

In this paper, to address the aforementioned challenges, we present the HUAP scheme, which provides a secure fine-grained access control system for resource-constrained devices in cloud-based applications. The contributions of this work can be summarized as follows.

- In HUAP, access policies are hidden, and hence the attribute privacy of the authorized data users is preserved such that the attribute sets satisfying the defined access policy cannot be guessed by unauthorized data users or the cloud service provider.
- HUAP realizes online/offline encryption in order to reduce the encryption overhead. Moreover, the proposed scheme achieves fast decryption, where the decryption algorithm consists of a constant number of bilinear pairing operations. Therefore, the computational overhead of encryption and decryption is considerably decreased.
- HUAP achieves a large attribute universe, where any string can be used as an attribute while the number of public parameters of the system remains constant.
- HUAP introduces a new concept called blind access policy. The attribute sets that satisfy a blind access policy are determined by the associated data owner defining the blind access policy. On the other side, the cloud service provider can encrypt several messages under a pre-generated blind access policy without knowing anything about the associated attributes that satisfy the policy.
- HUAP enables data owners to efficiently update their defined access policies. To reduce computational overhead on the user side, most of the operations related to the access policy update process are outsourced to the cloud service provider without leakage of any information about the previous and new access policies. In particular, a data owner can remain offline during the process of policy deletion (offline policy deletion), as the cloud can update the access policy without receiving any new re-encryption key.
- In HUAP, data collector devices perform independently of the access policy defined by the corresponding data owner. As a result, updating the access policy does not affect the performance of these devices. Moreover, many devices can perform simultaneously to collect the data corresponding to a data owner such that revealing information about the secret parameters of a device does not threaten the security of the other devices, the privacy of their collected data, or the hiddenness of the defined access policy.
- We prove that HUAP is selective ciphertext-policy and chosen-plaintext secure (CPA-secure) under the Decisional Bilinear Diffie–Hellman (DBDH) assumption and the Decisional Linear (DL) assumption in random oracle model.

2 Related Work

In this section, we summarize the related work on attribute-based encryption, anonymous attribute-based encryption, updating access policy in attribute-based encryption, and online/offline cryptography.

Attribute-based encryption. After introducing the notion of attribute-based encryption (ABE) by Sahai and Waters [2], key-policy attribute-based encryption (KP-ABE) proposed by Goyal *et al.* [3], and Ciphertext-policy attribute-based encryption (CP-ABE) proposed by Bethencourt *et al.* [4], divided this class of cryptographic schemes into two primary groups. However, CP-ABE seems to be more suitable than KP-ABE for providing fine-grained access control in public cloud-based data sharing applications. Because in CP-ABE, data owners can enforce their desired access policies over their outsourced data, while in KP-ABE, this is the attribute authority that encapsulates the access policies in secret keys issued for data users, and the data owners can only define a set of attributes related to their outsourced data. We refer the reader to [28–32] to study more about the topic.

Anonymous attribute-based encryption. Anonymous attribute-based encryption (A-ABE) has

been proposed to protect the users' attribute privacy. In anonymous ABE schemes, to protect sensitive information included in access policies of ciphertexts, the policies are hidden such that an unauthorized data user whose attributes do not satisfy an access policy cannot guess which attributes are required to decrypt the associated ciphertext. With regard to hidden access policies, there are two main categories in the literature: fully hidden and partially hidden. In fact, access policies consist of a set of attributes expressed as a couple: attribute name and attribute value [33]. To be specific, a fully hidden access policy obscures the names of the attributes in the policy as well as the values associated with each attribute name. On the other hand, in a partially hidden access policy, only the attribute values are hidden, and the secrecy of the attribute names is not provided. Kapadia *et al.* [8] proposed the first anonymous ABE scheme which supports AND-gate access policies on positive and negative attributes, but their scheme was vulnerable to collusion attacks. Nishide *et al.* [9] designed an efficient anonymous ABE scheme resisting collusion attacks. Li *et al.* [10] proposed an anonymous ABE scheme to realize user accountability. Afterward, Lai *et al.* [11] proposed an anonymous ABE scheme to protect user privacy and achieve full security. However, in their scheme, data users have to repeat the decryption algorithm until successful decryption is achieved, and if all the possible decryption tests are unsuccessful, then the user concludes that his attributes do not satisfy the underlying access policy. It is obvious that this approach is time-consuming, and the scheme is inefficient. To address the problem, Zhang *et al.* [7] designed a technique called match-then-decrypt that enables the data users to efficiently check whether their attributes satisfy a hidden access policy or not. Subsequently, they proposed another anonymous ABE scheme that also supports large universe and linear secret sharing scheme (LSSS) policies [6]. However, their proposed scheme is not adequately efficient as it is based on composite order groups dealing with large elements. Hao *et al.* [12] realized a fuzzy attribute positioning mechanism that fully hides access policies by applying a garbled bloom filter. Xiong *et al.* [34] proposed an anonymous attribute-based broadcast encryption scheme in edge computing that realizes direct revocation by embedding the list of identities of authorized data users in the ciphertext. However, by raising the number of users in the system, the number of system public parameters grows, and hence the scheme is not suitable for large networks. Zhang *et al.* [13] proposed an anonymous ABE scheme for personal health record systems. Their proposed scheme supports fast decryption. Also, by using hash functions, it enables data users to verify the validity of the received ciphertext. However, none of the aforementioned schemes support access policy update.

Online/offline cryptography. The notion of online/offline was first formalized by Even *et al.* [35] in digital signatures. In an online/offline signature scheme, the offline phase is performed before the message is known. Once the message is determined, the data owner uses a trapdoor to generate a dual signature. The technique of online/offline ABE was introduced by Hohenberger *et al.* [22]. Datta *et al.* [23] proposed the first adaptive payload-hiding online/offline KP-ABE scheme which supports a large attribute universe. Liu *et al.* [24] proposed an online/offline CP-ABE scheme for resource-constrained devices in the mobile cloud computing area. Li *et al.* [25] proposed an online/offline KP-ABE scheme that moves a vast majority of the encryption computational overhead on the data owner's side to the offline phase. The scheme realizes the public ciphertext test before performing the decryption algorithm and also eliminates a major part of the computational operations by adding some public parameters to the system. However, the aforementioned schemes support neither access policy update nor hidden access policies.

Access policy update. Updating the access policy is one of the most critical and essential tasks for access control administration. According to the existing schemes, the approach for access policy updating can be divided into the following: 1) deploying proxy re-encryption, 2) embedding required update parameters in the ciphertext. The notion of proxy re-encryption (PRE) was first formalized by Blaze *et al.* [36]. The first ciphertext-policy attribute-based proxy re-encryption (CP-ABPRE) scheme was proposed by Liang *et al.* [14]. In their cloud-based access control system, deploying CP-ABE, data owners can generate a re-encryption key to outsource updating their defined access policies. Using this re-encryption key, the proxy server updates the access policy of a ciphertext. Subsequently, Luo *et al.* [15] proposed another CP-ABPRE scheme supporting multi-value positive attributes. Afterward, an efficient CP-ABPRE scheme with a constant number of pairing operations was proposed by Seo *et al.* [16]. Liu *et al.* [37, 38] proposed the notion of time-based proxy re-encryption in which the access policies and the attribute secret keys are updated with respect to the global time of the system. Li *et al.* [39] proposed a fine-grained access control scheme with policy updating for the smart grid area. Also, Jiang *et al.* [20] designed a CP-ABE scheme supporting access policy updates based on AND-gate access policies. Huang *et al.* [40] proposed a hierarchical ABE for resource-constrained IoT devices that supports updating access policies. In order to relieve the local computational burden, their scheme

partially outsources the process of computationally expensive encryption operations to a gateway and decryption operations to the cloud. Li *et al.* [17] proposed a CP-ABE scheme that enables the data owner to outsource updating the access policy and also the shared files to reduce the storage and communication costs of the client. Sethi *et al.* [41] constructed a multi-authority ABE scheme that supports white-box traceability and access policy updates. Belguith *et al.* [19] presented a KP-ABE scheme that verifiably outsources the data decryption process to edge nodes. In their scheme, the data owner sends some secret parameters along with the ciphertext to the cloud. The cloud can utilize these parameters to update the access policy of the ciphertext. Hence, the scheme is capable of offline policy deletion, where the cloud can update the access policy without receiving any new re-encryption key from the data owner. Recently, Wang *et al.* [42] have proposed an ABE scheme with dynamic access policies for electronic health records shared in a cloud. They enable the cloud to update access policies without the need for authorized users to receive new decryption keys. Their design also supports outsourcing decryption. However, none of the aforementioned schemes hides the access policy.

To simultaneously support anonymity and access policy update, Zhang *et al.* [18] proposed an anonymous CP-ABPRE scheme in which the proxy server can update hidden access policies. However, to update an access policy, an authorized data user should generate a new re-encryption key for the proxy server, while the data owner cannot generate a valid re-encryption key. In addition, generating re-encryption keys requires running the whole of the encryption algorithm, which increases the computational and communication overhead on the user side. Moreover, it is necessary for the authorized data user to be online while providing the required re-encryption key. Afterward, Yan *et al.* [33] proposed a multi-authority attribute-based encryption scheme with dynamic policy updating for personal health record systems. Their scheme uses partially hidden access policies to protect the user's identity and attribute privacy. However, it does not fully hide the attributes in access policies, and hence the attribute names are disclosed.

On the other hand, some other schemes have been proposed to simultaneously support anonymity and online/offline encryption. Yan *et al.* [43] proposed an attribute-based encryption scheme with partially hidden policies for the Internet of Things. In this scheme, data users can outsource the decryption process to the cloud and then verify returned results. However, their construction is based on inefficient composite-order groups. Tian *et al.* [44] proposed a multi-authority attribute-based access control scheme with partially hidden policies for intelligent transportation systems. This scheme supports online/offline encryption and outsourced decryption to achieve lightweight computation for IoT devices. Sun *et al.* [45] proposed a lightweight policy-hiding attribute-based access control scheme with online/offline encryption for IoT-oriented s-health applications. The authors in this scheme propose an optimized vector transformation approach to decrease the overhead of key generation, encryption, and decryption algorithms. However, access policies are AND-gates on positive and negative attributes with wildcards, and hence the scheme is less expressive than other relevant schemes. Zhao *et al.* [46] proposed a multi-authority CP-ABE scheme with hidden policies that supports online/offline encryption. Zhang *et al.* [47] proposed a blockchain-enabled attribute-based access control scheme that supports both hidden policy and online/offline encryption and is suitable for edge computing in smart healthcare systems. Very recently, [48] have proposed an efficient attribute-based encryption scheme that achieves hidden policy as well as online/offline encryption and outsources decryption.

Table 1 summarizes the result of functional comparison between our proposed scheme and other similar ABE schemes in the literature that support at least one of the following features:

a) hidden access policy
b) online/offline encryption
c) access policy update

This paper is an extended version of a conference paper published in [49]. We extend our previous work by expanding system architecture, improving the related cryptographic structures, evaluating performance based on the actual execution time, and providing security proof in detail.

We refer the reader to [50, 51] to study more about the recent works in this area.

3 Preliminaries

In this section, we briefly present some cryptographic notions related to our work.

3.1 Cryptographic Background

In the following, we provide some cryptographic background.

Bilinear pairing. Assume that $\mathbb{G}$ and $\mathbb{G}_T$ are two cyclic multiplicative groups of a large prime order p, $1_{\mathbb{G}}$ is the identity of $\mathbb{G}$, $1_{\mathbb{G}_T}$ is the identity of $\mathbb{G}_T$, and g is a generator of $\mathbb{G}$. The map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a bilinear pairing if it satisfies the following properties:

a) Bilinear: For any $a, b \in Z_p$, we have $e\left(g^a, g^b\right) =$

Table 1. Security and performance comparison

Scheme	Access Policy Type	Hidden Access Policy	Large Universe	Fast Decryption	Online/offline Encryption	Access Policy Update	Offline Policy Deletion
[6]	LSSS	✓ Partially	✓	✗	✗	✗	-
[7]	AND-gates	✓ Fully	✓	✓	✗	✗	-
[12]	LSSS	✓ Fully	✗	✗	✗	✗	-
[13]	LSSS	✓ Partially	✓	✓	✗	✗	-
[17]	LSSS	✗	✗	✗	✗	✓	✗
[18]	AND-gates	✓ Fully	✗	✗	✗	✓	✗
[19]	LSSS	✗	✗	✗	✗	✓	✓
[20]	AND-gates	✗	✗	✓	✗	✓	✓
[24]	LSSS	✗	✗	✗	✓	✗	-
[25]	LSSS	✗	✗	✗	✓	✗	-
[33]	LSSS	✓ Partially	✗	✗	✗	✓	✗
[34]	LSSS	✓ Partially	✓	✗	✗	✗	-
[40]	LSSS	✗	✗	✗	✗	✓	✗
[41]	LSSS	✗	✓	✗	✗	✓	✗
[42]	LSSS	✗	✓	✗	✗	✓	✓
[43]	LSSS	✓ Partially	✓	✗	✓	✗	-
[44]	LSSS	✓ Partially	✓	✗	✓	✗	-
[45]	AND-gates	✓ Fully	✗	✓	✓	✗	-
[46]	LSSS	✓ Partially	✓	✗	✓	✗	-
[47]	LSSS	✓ Partially	✓	✗	✓	✗	-
[48]	LSSS	✓ Fully	✓	✗	✓	✗	-
HUAP	AND-gates	✓ Fully	✓	✓	✓	✓	✓

$e(g,g)^{a.b}$.

b) Non-degenerate: There exists at least two $g_1, g_2 \in \mathbb{G}$ such that $e(g_1, g_2) \neq 1_{\mathbb{G}_T}$.

c) Computable: For all $g_1, g_2 \in \mathbb{G}$, $e(g_1, g_2)$ can be computed by a polynomial-time algorithm.

Proxy re-encryption. Usually a proxy re-encryption scheme consists of three polynomial time algorithms: key generation, encryption and re-encryption, and three main entities: *Alice*, *Bob*, and a *proxy*. At first, there is a message M encrypted by *Alice*'s public key PK_{Alice} noted as C_{Alice}. Then a re-encryption key $RK_{Alice \to Bob}$ is sent to the *proxy* by *Alice*. The re-encryption key enables the *proxy* to re-encrypt C_{Alice} and create a new ciphertext C_{Bob} that is encrypted by PK_{Bob}. The main challenge in the proxy re-encryption is preventing the *proxy* from obtaining any information about the message M, and secret keys of *Alice* and *Bob*.

3.2 Complexity Assumptions

Below are some complexity assumptions.

Decisional Bilinear Diffie–Hellman (DBDH) assumption. Let $\mathbb{G}$ be a cyclic multiplicative group of a large prime order p, g be a generator of $\mathbb{G}$, $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be a bilinear pairing, $Z \in_R \mathbb{G}_T$, and $a, b, c \in_R Z_p$. We say that the DBDH assumption [52] holds if no probabilistic polynomial-time algorithm can distinguish the tuple $[g, g^a, g^b, g^c, e(g,g)^{abc}]$ from the tuple $[g, g^a, g^b, g^c, Z]$ with non-negligible advantage.

Decision Linear (D-Linear) assumption. Let $\mathbb{G}$ be a cyclic multiplicative group of a large prime order p, g be a generator of $\mathbb{G}$, and $z_1, z_2, z_3, z_4, z \in_R Z_p$. We say that the D-Linear assumption [53] holds if no probabilistic polynomial-time algorithm can distinguish the tuple $[g,\ g^{z_1}, g^{z_2}, g^{z_1 z_3}, g^{z_2 z_4}, g^{z_3+z_4}]$ from the tuple $[g, g^{z_1}, g^{z_2}, g^{z_1 z_3}, g^{z_2 z_4}, g^{z}]$ with non-negligible advantage.

3.3 Aceess Policies

Access policy is a rule W over some attributes. For a given attribute list L, the access policy returns true if L satisfies W and the notation $L \models W$ represents this situation. Otherwise, if L does not satisfy W, the notation $L \not\models W$ is used, and the access policy returns false.

In our scheme, the access policies consist of multiple AND-gates supporting multi-value attributes and wildcards where wildcard $*$ is known as "don't care" value. The notion generalizes the common concept of access policies in [9] consisting of a single AND-gate supporting multi-value attributes and wildcards. Assume that n is the total number of attributes in the system and $\mathbb{U} = \{\omega_1, \omega_2, \ldots, \omega_n\}$ is the universal attribute set. Each attribute can take multiple values, and the set of possible values for ω_i is $S_i = \{v_{i,1}, v_{i,2}, \ldots, v_{i,n_i}\}$ where n_i is the number of possible values for ω_i, $i = 1, 2, \ldots, n$.

Given an attribute list $L = [L_1, L_2, \ldots, L_n]$ and an access policy $A = \bigvee_{j=1}^{m} W_j$, where $W_j = [W_{j,1}, W_{j,2}, \ldots, W_{j,n}]$ and for all $1 \leq i \leq n, W_{j,i} \subseteq S_i$ and $L_i \in S_i$. In particular, $W_{j,i} = *$ means that

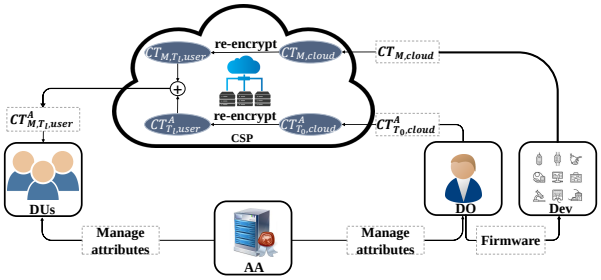


Figure 2. HUAP architecture

$W_{j,i} = S_i$. We say that L satisfies W_j and we write $L \models W_j$, if $L_i \in W_{j,i}$ for all $1 \leq i \leq n$. Otherwise, we say it does not satisfy W_j, $L \not\models W_j$. Also, a given attribute list $L = [L_1, L_2, \ldots, L_n]$ satisfies an access policy $A = \bigvee_{j=1}^{m} W_j$ if $L \models W_j$ for some $1 \leq j \leq m$. Otherwise, L does not satisfy A.

For example, assume that there are five attributes in the universe. We consider an access policy $A = W_1 \bigvee W_2$, where $W_1 = [W_{1,1} = \{v_{1,1}, v_{1,3}\}, W_{1,2} = \{v_{2,1}, v_{2,2}\}, W_{1,3} = *, W_{1,4} = *, W_{1,5} = *]$, and $W_2 = [W_{2,1} = *, W_{2,2} = \{v_{2,4}\}, W_{2,3} = \{v_{3,2}\}, W_{2,4} = *, W_{2,5} = *]$. According to the above access policy, if a recipient wants to decrypt a message corresponding to A, he must have the value $v_{1,1}$ or $v_{1,3}$ for ω_1, and $v_{2,1}$ or $v_{2,2}$ for ω_2, while the values for ω_3, ω_4, and ω_5 are not cared for, or he has to have the value $v_{2,4}$ for ω_2, and $v_{3,2}$ for ω_3, while the values for ω_1, ω_4, and ω_5 are not cared for.

4 System Model and Design Goals

In this section, firstly, we present the architecture of the system. Then, we give an overview of the proposed scheme. The latter consists of two parts: the algorithms of the scheme and the flow of sharing data in the system. Afterward, we review the trust model and security assumptions, explain blind access policies, and finally, describe the design goals of the proposed scheme.

4.1 System Architecture

As shown in Figure 2, the system architecture of the proposed scheme consists of the following entities:

- **Attribute Authority (AA):** It is a fully trusted entity that generates the system public key and system master key. It also generates attribute secret keys of users.
- **Cloud Service Provider (CSP):** It is an honest but curious entity with abundant storage capacity and computational power. Data encrypted by DOs are stored and managed by the CSP. It also provides a fine-grained access control service.
- **Device (Dev):** It is a device that generates private messages, encrypts, and sends them to the CSP.
- **Data Owner (DO):** It is a user that wishes to define a hidden access policy over encrypted messages generated by Dev and outsourced to the CSP.
- **Data User (DU):** It is a user with an attribute secret key associated with an attribute list L. It aims to access some encrypted data outsourced into the CSP. DU can decrypt the encrypted data if and only if her/his attributes satisfy the access policy of the ciphertext.

In this architecture, we have assumed that Dev is connected directly to the CSP. However, in practice, this connection can be through a semi-trusted gateway that honestly relays ciphertexts to the CSP [54].

4.2 Overview of Scheme

The proposed HUAP scheme consists of the following algorithms. The most relevant notations used in our scheme are summarized in Table 2:

a) $SystemSetup(1^\lambda) \rightarrow (PK, MK)$: The system setup algorithm is run by AA. A security parameter λ is chosen as the input of the algorithm. The outputs of the algorithm are the system public key PK which is published, and the system master key MK that is kept private.
b) $AttrKeyGen(PK, MK, L) \rightarrow SK_L$: The attribute key generation algorithm is run by AA. The system public key PK, the system master key MK, and an attribute list L are inputs of this algorithm. It returns the attribute secret key SK_L associated with the attribute list L as output.
c) $DOParamSetup\,(PK) \rightarrow (PP, SP)$: The data owner parameters setup algorithm is run by DO. The system public key PK is taken as input and the outputs of the algorithm are the data public parameter PP and the data secret parameter SP.
d) $RKeyGen(PP) \rightarrow RK$: The re-encryption key generation algorithm is run by DO to obtain a proxy re-encryption key. The data public parameter PP is taken as input and the re-encryption key RK is returned as the output.
e) $OfflineEncrypt(PK, PP) \rightarrow CT_{off}$: The offline encryption algorithm is run by Dev while it is offline. This algorithm takes the system public key PK and the data public parameter PP as input. It outputs an offline ciphertext CT_{off}.
f) $OnlineEncrypt(M, CT_{off}) \rightarrow CT_{M,cloud}$: The online encryption algorithm is run by Dev while it is online. This algorithm takes some message M, and an offline ciphertext CT_{off} as input. It outputs a message ciphertext $CT_{M,cloud}$.
g) $AnonEncrypt(PK, PP, SP, RK, A) \rightarrow CT^A_{T_0,cloud}$: The anonymous encryption algorithm is run by DO. This algorithm takes the system public key PK, the data public parameter PP, the data secret param-

Table 2. Summary of notations

Notation	Description
PK	system public key
MK	system master key
SK_L	attribute secret key (associated with attribute list L)
PP	data public parameter
SP	data secret parameter
RK	re-encryption key
CT_{off}	offline ciphertext
$CT_{M,cloud}$	message ciphertext
$CT^A_{T_0,cloud}$	policy ciphertext (associated with hidden access policy A)
CT^A_{cloud}	cloud ciphertext (associated with hidden access policy A)
$CT_{M,T_l,user}$	re-encrypted message ciphertext (with respect to timestamp T_l)
$CT^A_{T_l,user}$	re-encrypted policy ciphertext (associated with hidden access policy A with respect to timestamp T_l)
$CT^A_{M,T_l,user}$	user ciphertext (associated with hidden access policy A with respect to timestamp T_l)
dk_{T_l}	data decryption key (with respect to timestamp T_l)
CT_{W_j,T_l}	data decryption key dk_{T_l} encrypted under sub-policy W_j (with respect to timestamp T_l)
$\widetilde{CT}_{W_j,T_0}$	blind access policy associated with sub-policy W_j

eter SP, the re-encryption key RK, and an access policy $A = \bigvee_{j=1}^{m} W_j$ as inputs. It outputs a policy ciphertext $CT^A_{T_0,cloud} = \{CT_{W_j,T_0}, \widetilde{CT}_{W_j,T_0}\}_{1\le j\le m}$, where for $1 \le j \le m$, the two components CT_{W_j,T_0} and $\widetilde{CT}_{W_j,T_0}$ are associated with W_j.

h) Reencrypt$(PP, RK, T_l, CT^A_{cloud}) \rightarrow CT^A_{M,T_l,user}$: The re-encryption algorithm is run by CSP. The data public parameter PP, the re-encryption key RK, a timestamp T_l, and some cloud ciphertext $CT^A_{cloud} = (CT_{M,cloud}, CT^A_{T_0,cloud} = \{CT_{W_j,T_0}, \widetilde{CT}_{W_j,T_0}\}_{1\le j\le m})$ are the inputs of the algorithm. The output is a user ciphertext under the hidden access policy A with respect to the timestamp T_l, denoted as

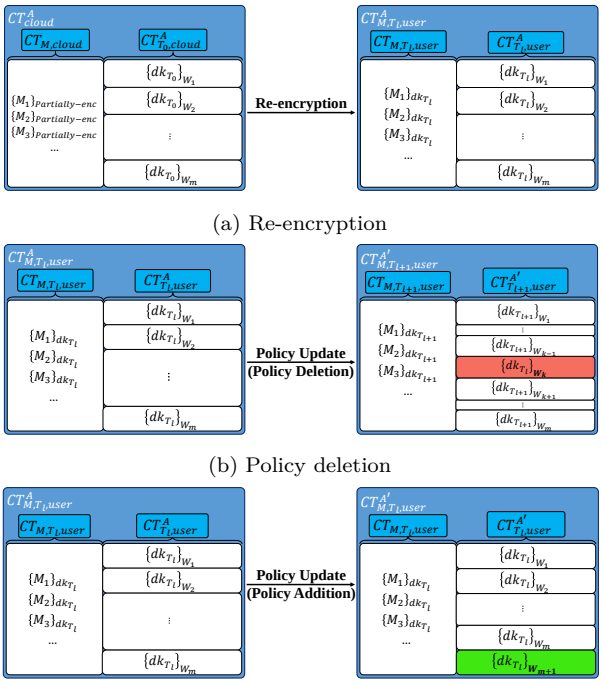
$$\begin{aligned} CT^A_{M,T_l,user} &= (CT_{M,T_l,user}, CT^A_{T_l,user} \\ &= \{CT_{W_j,T_l}, \widetilde{CT}_{W_j,T_l}\}_{1\le j\le m}). \end{aligned}$$

Here, $CT_{M,T_l,user}$ is the re-encrypted version of $CT_{M,cloud}$, and for $1 \le j \le m$, CT_{W_j,T_l} and $\widetilde{CT}_{W_j,T_l}$ are the re-encrypted versions of CT_{W_j,T_0} and $\widetilde{CT}_{W_j,T_0}$ respectively.

i) AnonDecrypt$(PK, PP, CT^A_{M,T_l,user}, SK_L) \rightarrow M$ or $\perp$: The anonymous decryption algorithm is run by DU. The system public key PK, the data public parameter PP, some user ciphertext $CT^A_{M,T_l,user} = (CT_{M,T_l,user}, CT^A_{T_l,user} = \{CT_{W_j,T_l}, \widetilde{CT}_{W_j,T_l}\}_{1\le j\le m})$, and the attribute secret key SK_L are inputs of the algorithm. The output is the original message M or $\perp$. This algorithm consists of two phases: *matching phase* and *decryption phase*.

(a) *Matching phase:* If $L \not\models W_j$ for all $1 \le j \le m$, this phase returns $\perp$ and anonymous decryption algorithm is terminated. Otherwise, the subsequent decryption phase is run.

(b) *Decryption phase:* This phase returns the message M.



(a) Re-encryption

(b) Policy deletion

(c) Policy addition

Figure 3. Ciphertexts

Now, an overview of the HUAP scheme is given in the following:

a) System initialization: The AA runs the *SystemSetup* algorithm to generate the system public key PK and the system master key MK. PK is published by AA and MK is kept private to itself. Then, it runs *AttrKeyGen* algorithm when it receives a request from an authorized DU. The generated attribute secret key is returned to the DU.

b) Data initialization: The DO runs *DOParamSetup* algorithm to generate the data public parameter PP and the data secret parameter SP. He publishes PP and keeps SP confidential. DO also runs *RKeyGen* algorithm and sends the generated re-encryption key RK to the CSP through a secure channel.

c) Data outsource: Before the message is determined, Dev uses the system public key PK and the data public parameter PP to prepare some offline ciphertexts CT_{off} by running $OfflineEncrypt$ algorithm. Once a message M is known, Dev runs $OnlineEncrypt$ algorithm to encrypt the message and calculate the message ciphertext $CT_{M,cloud}$. Then, Dev sends $CT_{M,cloud}$ to the CSP. On the other side, the DO uses the data secret parameter SP to calculate a data decryption key dk_{T_0}. Then, she/he defines a hidden access policy $A = \bigvee_{j=1}^{m} W_j$ and encrypts the data decryption key dk_{T_0} under A by running $AnonEncrypt$ algorithm. The output is a policy ciphertext $CT^A_{T_0,cloud} = \{CT_{W_j,T_0}, \widetilde{CT}_{W_j,T_0}\}_{1\le j\le m}$, in which for $1 \le j \le m$, CT_{W_j,T_0} represents the data decryption key dk_{T_0} encrypted under sub-policy W_j, and $\widetilde{CT}_{W_j,T_0}$ is a blind access policy. Afterward, the DO sends $CT^A_{T_0,cloud}$ to the CSP. When the CSP receives $CT^A_{cloud} = (CT_{M,cloud}, CT^A_{T_0,cloud} = \{CT_{W_j,T_0}, \widetilde{CT}_{W_j,T_0}\}_{1\le j\le m})$ from Dev and DO, it runs $Reencrypt$ algorithm using the current timestamp T_l to generate a user ciphertext $CT^A_{M,T_l,user} = (CT_{M,T_l,user}, CT^A_{T_l,user} = \{CT_{W_j,T_l}, \widetilde{CT}_{W_j,T_l}\}_{1\le j\le m})$. Here, for $1 \le j \le m$, the component CT_{W_j,T_l} represents the data decryption key dk_{T_l} encrypted under sub-policy W_j. The data decryption key dk_{T_l} can be used to retrieve the underlying message M from $CT_{M,T_l,user}$. Finally, $CT^A_{M,T_l,user}$ is published by the CSP for DUs. Figure 3a shows the cloud ciphertext CT^A_{cloud} and the user ciphertext $CT^A_{M,T_l,user}$ during re-encryption.

d) Data access: When a DU with attribute secret key SK_L wants to decrypt a user ciphertext $CT^A_{M,T_l,user}$ which is encrypted under a hidden access policy $A = \bigvee_{j=1}^{m} W_j$, for $1 \le j \le m$, she/he runs the *matching* phase of $AnonDecrypt$ algorithm to check whether his attribute secret key SK_L satisfies W_j or not. If $L \models W_j$ for some j, DU runs the *decryption* phase and obtains the associated message M. Otherwise, $AnonDecrypt$ algorithm returns $\perp$.

e) Access policy update: A DO can update his defined access policy $A = \bigvee_{j=1}^{m} W_j$ at any time. Let $CT^A_{cloud} = (CT_{M,cloud}, CT^A_{T_0,cloud} = \{CT_{W_j,T_0}, \widetilde{CT}_{W_j,T_0}\}_{1\le j\le m})$ be a cloud ciphertext associated with a hidden access policy A. Assume that the CSP has re-encrypted the cloud ciphertext CT^A_{cloud} using the timestamp T_l, and has published the resulted user ciphertext $CT^A_{M,T_l,user} = (CT_{M,T_l,user}, CT^A_{T_l,user} = \{CT_{W_j,T_l}, \widetilde{CT}_{W_j,T_l}\}_{1\le j\le m})$. Moreover, assume that the DO wants to update the access policy A by defining a new access policy, A'. In the following, the policy deletion and the policy addition operations are described. It can be shown that all possible updates of an access policy can be made by performing combinations of policy deletion and policy addition operations.

(a) *Policy deletion:* If a DO wants to revoke some DUs whose attributes satisfy W_k, $1 \le k \le m$, he should define a new access policy as $A' = \bigvee_{j=1, j\neq k}^{m} W_j$. So, if DO is online, he only should send a request to the CSP. Otherwise, he should determine an expiration date for W_k. Upon receiving the request or reaching the expiration date, the CSP deletes the components CT_{W_k,T_0} and $\widetilde{CT}_{W_k,T_0}$ from the cloud ciphertext CT^A_{cloud}. Then, DO re-encrypts the new cloud ciphertext $CT^{A'}_{cloud}$ by running $Reencrypt$ algorithm. The output of the algorithm is a user ciphertext $CT^{A'}_{M,T_{l+1},user}$ which is under the new access policy A' and the new timestamp T_{l+1}. Figure 3b shows a user ciphertext published by the CSP for DUs during the policy deletion operation.

(b) *Policy addition:* If a DO wants to expand the access policy A, he should define a new access policy as $A' = \left(\bigvee_{j=1}^{m} W_j\right) \bigvee W_{m+1}$. Therefore, DO just should provide two new components CT_{W_{m+1},T_0} and $\widetilde{CT}_{W_{m+1},T_0}$ associated with W_{m+1}. Then, by re-encrypting these two components, the CSP computes CT_{W_{m+1},T_l} and $\widetilde{CT}_{W_{m+1},T_l}$. Then, the CSP appends the re-encrypted components to the user ciphertext $CT^A_{T_l,user}$. Hence, there is no need for the CSP to change the previously published components $CT_{M,T_l,user}$ or $CT^A_{T_l,user}$. Figure 3c shows a user ciphertext generated by the CSP for DUs during policy addition operation.

4.3 Security Model

AA is assumed to be trusted. The CSP is assumed to be honest but curious. It executes the given protocol correctly, but it may try to obtain additional information about the stored data. All DUs are assumed to be malicious; they try to learn some unauthorized information about data stored in the CSP. Also, it is assumed that the CSP does not collude with DUs, while unauthorized DUs may collude with each other to access the data outsourced to the CSP.

The security of our proposed scheme is proven in the indistinguishability against selective ciphertext-policy and chosen-plaintext attacks (IND-sCP-CPA) security model [4, 7, 55]. The model is an interactive game between an adversary and a challenger. The adversary attempts to (1) obtain some information about a plaintext from the corresponding ciphertext, and (2) distinguish the access policies embedded in ciphertexts.

4.3.1 IND-sCP-CPA Game

Init: $\mathcal{A}$ submits two challenge access policies A_0^* and A_1^* to the challenger. $\mathcal{A}$ also submits a timestamp T_l.

Setup: The challenger $\mathcal{S}$ specifies a security parameter λ, and runs the $SystemSetup$ algorithm to get a system master key MK and the corresponding system public key PK. Also, $\mathcal{S}$ runs the $DOParamSetup$ algorithm to get a data secret parameter SP and the corresponding data public parameter PP. Moreover, $\mathcal{S}$ runs the $RKeyGen$ algorithm to get a re-encryption key RK. It keeps MK, SP, and RK secretly and gives PK and PP to $\mathcal{A}$.

Phase 1: The adversary $\mathcal{A}$ makes some queries to the following oracles:

- **AttrKeyGen oracle $\mathcal{O}_{AttrKeyGen}$:** $\mathcal{A}$ submits an attribute list L. The challenger runs the $AttrKeyGen$ algorithm and returns the corresponding attribute secret key SK_L to $\mathcal{A}$ only if $L \nvDash A_0^* \wedge L \nvDash A_1^*$. Otherwise, it outputs $\perp$.
- **Reencrypt oracle $\mathcal{O}_{Reencrypt}$:** The adversary $\mathcal{A}$ submits a timestamp T_i, and a cloud ciphertext CT^A_{cloud}. The challenger runs the $Reencrypt$ algorithm and returns the corresponding user ciphertext $CT^A_{M,T_i,user}$ only if $i \leq l$. Otherwise, it outputs $\perp$.

Challenge: When Phase 1 is over, $\mathcal{A}$ sends two different equal-length messages M_0 and M_1 to the challenger. Afterward, the challenger first runs $OfflineEncrypt$ algorithm to get an offline ciphertext CT_{off}. Then, the challenger randomly selects a bit $\nu \in \{0,1\}$, and computes the message ciphertext $CT_{M_\nu,cloud} = OnlineEncrypt(M_\nu, CT_{off})$. The challenger also computes the policy ciphertext $CT^{A^*_\nu}_{T_0,cloud} = AnonEncrypt(PK, PP, SP, RK, A^*_\nu)$. Finally, the challenger returns the cloud ciphertext $CT^{A^*_\nu}_{cloud} = \{CT_{M_\nu,cloud}, CT^{A^*_\nu}_{T_0,cloud}\}$ to $\mathcal{A}$.

Phase 2: It is similar to Phase 1.

Guess: $\mathcal{A}$ outputs a bit $\nu' \in \{0,\ 1\}$ as a guess of ν, and it wins the game if $\nu' = \nu$.
In the IND-sCP-CPA game, we define the advantage of $\mathcal{A}$ as follows,
$\mathrm{Adv}^{IND-sCP-CPA}_{HUAP}(\mathcal{A}) = \left|Pr\left[\nu' = \ \nu\right] - \frac{1}{2}\right|$.

Definition 1. HUAP is said to be IND-sCP-CPA secure if the advantage of a probabilistic polynomial-time adversary to win the IND-sCP-CPA game is a negligible function in the security parameter.

4.4 Blind Access Policies

HUAP introduces the innovative concept of a "blind access policy," empowering data owners to define hidden attribute sets fulfilling the policy. Consequently, the cloud service provider (CSP) can encrypt multiple messages using a pre-generated blind access policy, without requiring knowledge of the attributes complying with the policy. A blind access policy is similar to the regular ciphertext of an ABE scheme with hidden policies, except it contains no messages. This inspiration comes from online/offline encryption schemes, where an offline ciphertext is created without any messages. In essence, a blind access policy serves as a hidden access policy capable of encrypting any provided message.

In our proposed scheme, the data owner generates a policy ciphertext $CT^A_{T_0,cloud} = \{CT_{W_j,T_0}, \widetilde{CT}_{W_j,T_0}\}_{1\leq j\leq m}$, comprising two parts for each $1 \leq j \leq m$: the data decryption key dk_{T_0} encrypted under sub-policy W_j (i.e. CT_{W_j,T_0}) and $\widetilde{CT}_{W_j,T_0}$, representing the blind access policy, with the attribute set identical to the hidden policy W_j. When the CSP updates the decryption key dk_{T_0} through the $Reencrypt$ algorithm, it simultaneously encrypts the corresponding updating parameter using the blind access policy. As a result, only data users whose attributes satisfy the blind access policy can compute the updating parameter and obtain the updated decryption key dk_{T_l}. Importantly, the CSP remains unaware of any information regarding the attributes defined in the blind access policy, as it closely resembles a hidden policy.

4.5 Design Goals

The following security and performance goals are considered in our proposed scheme.

a) Access policy update: The DO should be able to update defined access policies. In particular, the DO should be able to revoke the access right of a group of DUs at any time, whether she/he is online or even offline.
b) Fine-grained access control: The DO should be able to define a desired access policy for each part of his data.
c) Data confidentiality: The CSP and unauthorized DUs must not be able to access the outsourced data.
d) Collusion resistance: Multiple malicious data users may collude with each other to access some stored data by combining their attribute secret keys. Our scheme must resist such collusion attacks.
e) Attribute privacy protection: In many applications, such as s-health, the access policy itself is considered sensitive information and must be hidden. Therefore,

the CSP and unauthorized DUs must not be able to obtain any information about the access policies defined by the DO.
f) Cost efficiency: The computational cost on DUs, DOs, and Devs should be minimized.

5 HUAP: Attribute-Based Access Control Supporting Hidden Updatable Access Policies

In this section, we present our anonymous CP-ABE scheme that supports hidden updatable access policies. The proposed scheme utilizes online/offline encryption to reduce the computational cost for resource-constrained devices. Our scheme also enables data owners to outsource a major part of the access policy update process to the CSP without the need to generate new re-encryption keys. Our proposed outsourcing approach is based on blind access policies. In fact, the data owner defines a blind access policy and sends it to the cloud along with other ciphertext components. This blind access policy enables the cloud to share random parameters with authorized data users determined by the data owner. Whenever an access policy update is required, the cloud itself generates a new random parameter and re-encrypts all the past and future encrypted messages based on this random parameter such that the revoked data users cannot decrypt these re-encrypted messages. Then the cloud utilizes a blind access policy to share this random parameter with other authorized data users. In particular, the cloud cannot obtain any information about the identities or the attributes of these authorized data users.

In this scheme, we split the ciphertext into two major parts. The first part is associated with encrypted messages and is generated accumulatively by resource-constrained devices. The second part is corresponding to the hidden access policy which is defined by the data owner. In fact, the messages are encrypted in the first part, and the corresponding decryption key is encrypted under a hidden access policy in the second part. Therefore, a data owner can define hidden access policies, while the devices that generate and encrypt messages do not need to know anything about the defined access policies.

5.1 Our Proposed Construction

a) SystemSetup(1^λ)*:* AA chooses two cyclic multiplicative groups $\mathbb{G}$ and $\mathbb{G}_T$ of a large prime order p, g as a generator of $\mathbb{G}$ and $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ as a bilinear pairing. Let $H : \{0,1\}^* \to \mathbb{G}$, $\widehat{H} : \{0,1\}^* \to Z_p^*$ be two hash functions, and $F : \mathbb{G}_T \to \mathbb{G}$ be a function mapping elements of $\mathbb{G}_T$ to elements of $\mathbb{G}$. AA also chooses uniformly at random $y \in_R Z_p$ and $g_1, g_2, g_3, g_4 \in_R \mathbb{G}$, and computes $Y = e(g_1, g_2)^y$. The system public key is published as $PK = (g, g_1, g_2, g_3, g_4, Y)$ and the system master key $MK = (y)$ is kept private by AA.
b) DOParamSetup(PK)*:* DO chooses mk_0, mk_1, $sk \in_R Z_p^*$, uniformly at random. Then the data public parameter $PP = (Q_0 = g_3^{sk}, PP_0 = e(g_3, g_4)^{mk_0}, PP_1 = g_3^{mk_1})$ is published and the data secret parameter $SP = (mk_0, mk_1, SK_1 = g_4^{mk_0}, sk)$ is kept private by DO.
c) AttrKeyGen(PK, MK, L)*:* Assume that AA wants to generate an attribute secret key corresponding to an attribute list $L = [L_1, L_2, \ldots, L_n]$. Also, assume that the universal attribute set is $\mathbb{U} = \{\omega_1, \omega_2, \ldots, \omega_n\}$ and each attribute supports multiple values, where the multi-value set for ω_i is $S_i = \{v_{i,1}, v_{i,2}, \ldots, v_{i,n_i}\}$. For $1 \le i \le n$, AA chooses $r_i \in_R Z_p$ such that $\sum_{i=1}^{n} r_i = y$. Also, AA chooses $\hat{r}_i \in_R Z_p$ for $1 \le i \le n$ and computes $\hat{r} = \sum_{i=1}^{n} \hat{r}_i$. Then AA chooses $r, \lambda, \hat{\lambda} \in_R Z_p$ and computes $D_0 = g_2^{\lambda}$, $\widehat{D}_0 = g_1^{\hat{\lambda}}$, $D_{\Delta,0} = g_1^{r}$ and $\widehat{D}_{\Delta,0} = g_2^{y-\hat{r}}$. For $1 \le i \le n$, suppose that $L_i = v_{i,k_i}$, AA computes $\left[D_{\Delta,i}, D_{i,1}, \widehat{D}_{i,1}\right]$ as follows,

$$\left[\begin{array}{l} D_{\Delta,i} = g_2^{\hat{r}_i}.H(i||v_{i,k_i})^{r}, D_{i,1} = g_1^{r_i}.H\left(0\,||i||\,v_{i,k_i}\right)^{\lambda}, \\ \widehat{D}_{i,1} = g_2^{r_i}.H\left(1\,||i||\,v_{i,k_i}\right)^{\hat{\lambda}} \end{array}\right].$$

Finally, the attribute secret key is $SK_L = \langle D_0, \widehat{D}_0, D_{\Delta,0}, \widehat{D}_{\Delta,0}, \left\{D_{\Delta,i}, D_{i,1}, \widehat{D}_{i,1}\right\}_{1\le i\le n}\rangle$.
d) RKeyGen(PP)*:* DO selects $s_{cloud} \in_R Z_p$ and sets re-encryption key as $RK = s_{cloud}$. Then DO sends the re-encryption key RK to the CSP through a secure channel. Therefore, the DO and the CSP will be able to compute $S_{T_l} = \widehat{H}(s_{cloud}||l)$ for $l \in \mathbb{Z}^+$.
e) OfflineEncrypt(PK, PP)*:* Dev chooses $r_d \in_R Z_p^*$ and computes offline ciphertext CT_{off} as follows,

$$CT_{off} = \left(U_0 = g_3^{r_d}, U_1 = g_3^{r_d.sk}, V_0 = e\left(g_3, g_4\right)^{r_d.mk_0}\right).$$

f) OnlineEncrypt(M, CT_{off})*:* Once the message M is determined, Dev calculates the message ciphertext $CT_{M,cloud}$, signs it and finally sends it to the CSP:

$$CT_{M,cloud} = (U_0, U_1, V = M.V_0)\,.$$

g) AnonEncrypt(PK, PP, SP, RK, A)*:* Suppose that a DO wants to define an access policy $A = \bigvee_{j=1}^{m} W_j$ where $W_j = [W_{j,1}, W_{j,2}, \ldots, W_{j,n}]$. First, DO computes the data decryption key with respect to the timestamp T_0 as $dk_{T_0} = g_4^{mk_0}.g_3^{sk.mk_1}.g_3^{mk_1.S_{T_0}}$. Then, for $1 \le j \le m$: DO chooses $s_{1,j}, s'_{1,j}, s''_{1,j}, s_{2,j}, s''_{2,j} \in_R Z_p^*$. Also, for $1 \le i \le n$, the DO chooses $\{\sigma_{i,j,\Delta}, \sigma_{i,j,0}, \sigma_{i,j,1}, \sigma'_{i,j,0}, \sigma'_{i,j,1} \in_R \mathbb{G} | 1 \le i \le n\}$ such that $\prod_{i=1}^{n} \sigma_{i,j,\Delta} = \prod_{i=1}^{n} \sigma_{i,j,0} = \prod_{i=1}^{n} \sigma_{i,j,1} = \prod_{i=1}^{n} \sigma'_{i,j,0} = \prod_{i=1}^{n} \sigma'_{i,j,1} = 1_{\mathbb{G}}$. Then, the DO computes $[C_{i,t,\Delta,w_j}, C_{i,t,0,w_j}, \widehat{C}_{i,t,0,w_j}, C'_{i,t,0,w_j}, \widehat{C}'_{i,t,0,w_j}]$ for $1 \le i \le n$ as follows,

(a) If $v_{i,t} \in W_{j,i}$, then

$$\left[\begin{array}{l} C_{i,t,\Delta,w_j} = \sigma_{i,j,\Delta}.H(i||v_{i,t})^{s'_{1,j}}, \\ C_{i,t,0,w_j} = \sigma_{i,j,0}.H\left(0\,||i||\,v_{i,t}\right)^{s''_{1,j}}, \\ \widehat{C}_{i,t,0,w_j} = \sigma_{i,j,1}.H\left(1\,||i||\,v_{i,t}\right)^{s_{1,j}-s''_{1,j}}, \\ C'_{i,t,0,w_j} = \sigma'_{i,j,0}.H\left(0\,||i||\,v_{i,t}\right)^{s''_{2,j}}, \\ \widehat{C}'_{i,t,0,w_j} = \sigma'_{i,j,1}.H(1||i||v_{i,t})^{s_{2,j}-s''_{2,j}} \end{array}\right].$$

(b) If $v_{i,t} \notin W_{j,i}$, then $[C_{i,t,\Delta,w_j}, C_{i,t,0,w_j}, \widehat{C}_{i,t,0,w_j}, C'_{i,t,0,w_j}, \widehat{C}'_{i,t,0,w_j}]$ are random elements in $\mathbb{G}$.

Then, DO computes CT_{W_j,T_0} and $\widetilde{CT}_{W_j,T_0}$ as follows,

$$\widetilde{CT}_{W_j,T_0} = \left(\begin{array}{l} \widetilde{C}'_{w_j} = Y^{s_{2,j}}, C'_{1,w_j} = g_2^{s''_{2,j}}, \widehat{C}'_{1,w_j} = g_1^{s_{2,j}-s''_{2,j}}, \\ \{\{\ C'_{i,t,0,w_j}, \widehat{C}'_{i,t,0,w_j}\}_{1\le t\le n_i}\}_{1\le i\le n} \end{array}\right),$$

$$CT_{W_j,T_0} = \left(\begin{array}{l} \widetilde{C}_{w_j} = dk_{T_0}.F\left(Y^{s_{1,j}}\right), C_{\Delta,w_j} = Y^{s'_{1,j}}, \\ \widehat{C}_{0,w_j} = g_1^{s'_{1,j}}, C_{1,w_j} = g_2^{s''_{1,j}}, \widehat{C}_{1,w_j} = g_1^{s_{1,j}-s''_{1,j}}, \\ \{\{\ C_{i,t,\Delta,w_j}, C_{i,t,0,w_j}, \widehat{C}_{i,t,0,w_j}\}_{1\le t\le n_i}\}_{1\le i\le n} \end{array}\right).$$

Finally, the policy ciphertext which is prepared for sending to the cloud is as follows, where $\widetilde{CT}_{W_j,T_0}$ is a blind access policy for $1 \le j \le m$:

$$CT^A_{T_0,cloud} = \{CT_{W_j,T_0}, \widetilde{CT}_{W_j,T_0}\}_{1\le j\le m}.$$

h) Reencrypt$(PP, RK, T_l, CT^A_{cloud})$*:* Suppose that the CSP wants to re-encrypt the ciphertext $CT^A_{cloud} = (CT_{M,cloud}, CT^A_{T_0,cloud} = \{CT_{W_j,T_0}, \widetilde{CT}_{W_j,T_0}\}_{1\le j\le m})$ for the timestamp T_l, where the underlying access policy is $A = \bigvee_{j=1}^{m} W_j$. First, the CSP chooses $r' \in_R Z_p^*$, calculates $S_{T_l} = \widehat{H}(RK||l)$, and computes $CT_{M,T_l,user}$ as follows,

$$CT_{M,T_l,user} = \left[\begin{array}{l} U_0^{T_l} = U_0.g_3^{r'}, \\ U_1^{T_l} = U_1.g_3^{r'.sk}.\left(U_0^{T_l}\right)^{S_{T_l}}, \\ V^{T_l} = V.e\left(g_3, g_4\right)^{mk_0.r'} \end{array}\right].$$

Then, for $1 \le j \le m$, suppose that CT_{W_j,T_0} and $\widetilde{CT}_{W_j,T_0}$ are as the same as in *AnonEncrypt* algotithm. CSP chooses $r''_{w_j}, r_{w_j} \in_R Z_p^*$ and computes $\widetilde{CT}_{W_j,T_l}$ and CT_{W_j,T_l} as follows,

$$\widetilde{CT}_{W_j,T_l} = \left(\begin{array}{l} \widetilde{C}'_{w_j,T_l} = Y^{r_{w_j}}.\left(\widetilde{C}'_{w_j}\right)^{r''_{w_j}}, \\ C'_{1,w_j,T_l} = \left(C'_{1,w_j}\right)^{r''_{w_j}}, \widehat{C}'_{1,w_j,T_l} = \left(\widehat{C}'_{1,w_j}\right)^{r''_{w_j}}, \\ \left\{\left\{\begin{array}{l} C'_{i,t,0,w_j,T_l} = \left(C'_{i,t,0,w_j}\right)^{r''_{w_j}}, \\ \widehat{C}'_{i,t,0,w_j,T_l} = \left(\widehat{C}'_{i,t,0,w_j}\right)^{r''_{w_j}} \end{array}\right\}_{1\le t\le n_i}\right\}_{1\le i\le n} \end{array}\right),$$

$$CT_{W_j,T_l} = \left(\begin{array}{l} \widetilde{C}_{w_j,T_l} = F\left(Y^{r_{w_j}}\right).g_3^{mk_1.\left(S_{T_l}-S_{T_0}\right)}.\widetilde{C}_{w_j}, \\ C_{\Delta,w_j,T_l} = C_{\Delta,w_j}, \widehat{C}_{0,w_j,T_l} = \widehat{C}_{0,w_j}, \\ C_{1,w_j,T_l} = C_{1,w_j}, \widehat{C}_{1,w_j,T_l} = \widehat{C}_{1,w_j}, \\ \left\{\left\{\begin{array}{l} C_{i,t,\Delta,w_j,T_l} = C_{i,t,\Delta,w_j}, \\ C_{i,t,0,w_j,T_l} = C_{i,t,0,w_j}, \\ \widehat{C}_{i,t,0,w_j,T_l} = \widehat{C}_{i,t,0,w_j} \end{array}\right\}_{1\le t\le n_i}\right\}_{1\le i\le n} \end{array}\right).$$

where $\widetilde{C}_{w_j,T_l} = F\left(Y^{r_{w_j}}\right).dk_{T_l}.F\left(Y^{s_{1,j}}\right)$. Here, for $1 \le j \le m$, $\widetilde{CT}_{W_j,T_0}$ is a blind access policy that is utilized to share the randomly generated parameter $Y^{r_{w_j}}$ with authorized data users. Finally, the re-encrypted ciphertext which is prepared for data users with respect to the timestamp T_l is:

$$CT^A_{M,T_l,user} = \left(\begin{array}{l} CT_{M,T_l,user}, \\ CT^A_{T_l,user} = \left\{CT_{W_j,T_l}, \widetilde{CT}_{W_j,T_l}\right\}_{1\le j\le m} \end{array}\right).$$

i) AnonDecrypt$(PK, PP, CT_{M,T_l,user}, CT_{W_j,T_l}, \widetilde{CT}_{W_j,T_l}, SK_L)$*:* DU tests and decrypts ciphertext $CT_{M,T_l,user}$ with attribute secret key SK_L in two following phases:

(a) *matching phase:* For $1 \le i \le n$, suppose that $L_i = v_{i,t}$. $L \models W_j$ if and only if the following equation holds:

$$C_{\Delta,w_j,T_l} = \frac{e\left(\widehat{C}_{0,w_j,T_l}, \widehat{D}_{\Delta,0}.\prod_{i=1}^{n} D_{\Delta,i}\right)}{e\left(\prod_{i=1}^{n} C_{i,t,\Delta,w_j,T_l}, D_{\Delta,0}\right)}. \quad (1)$$

If $L \models W_j$, the subsequent decryption phase is started. Otherwise, the algorithm *AnonDecrypt* returns $\perp$.

(b) *decryption phase:* Suppose that $L \models W_j$ and $L_i = v_{i,t}$ for $1 \le i \le n$. At first, the DU computes $Y^{r_{w_j}}$ and $Y^{s_{1,j}}$ as follows,

$$Y^{r_{w_j}} = \widetilde{C}'_{w_j,T_l}.\frac{e\left(\prod_{i=1}^{n} C'_{i,t,0,w_j,T_l}, D_0\right)}{e\left(C'_{1,w_j,T_l}, \prod_{i=1}^{n} D_{i,1}\right)}.\frac{e\left(\prod_{i=1}^{n} \widehat{C}'_{i,t,0,w_j,T_l}, \widehat{D}_0\right)}{e\left(\widehat{C}'_{1,w_j,T_l}, \prod_{i=1}^{n} \widehat{D}_{i,1}\right)}, \quad (2)$$

$$Y^{s_{1,j}} = \frac{e\left(C_{1,w_j,T_l}, \prod_{i=1}^{n} D_{i,1}\right)}{e\left(\prod_{i=1}^{n} C_{i,t,0,w_j,T_l}, D_0\right)}.\frac{e\left(\widehat{C}_{1,w_j,T_l}, \prod_{i=1}^{n} \widehat{D}_{i,1}\right)}{e\left(\prod_{i=1}^{n} \widehat{C}_{i,t,0,w_j,T_l}, \widehat{D}_0\right)}. \quad (3)$$

Then, the DU computes dk_{T_l} as follows,

$$dk_{T_l} = \frac{\widetilde{C}_{w_j,T_l}}{F\left(Y^{r_{w_j}}\right).F\left(Y^{s_{1,j}}\right)} \quad (4)$$

Finally, the DU retrieves the message M as follows,

$$M = V^{T_l} / \frac{e\left(U_0^{T_l}, dk_{T_l}\right)}{e\left(PP_1, U_1^{T_l}\right)}. \quad (5)$$

5.2 Consistency of the Proposed Construction

In the following, we show the correctness of equations (1) to (5). Firstly, the attributes satisfy the access policy if and only if (1) holds, as it is shown in the following:

$$\begin{aligned}
&\frac{e\left(\widehat{C}_{0,w_j,T_l}, \widehat{D}_{\Delta,0} \cdot \prod_{i=1}^{n} D_{\Delta,i}\right)}{e\left(\prod_{i=1}^{n} C_{i,t,\Delta,w_j,T_l}, D_{\Delta,0}\right)} \\
&= \frac{e\left(g_1^{s'_{1,j}}, g_2^{y-\hat{r}} \cdot \prod_{i=1}^{n} g_2^{\hat{r}_i} . H(i||v_{i,k_i})^{r}\right)}{e\left(\prod_{i=1}^{n} \sigma_{i,j,\Delta} . H(i||v_{i,t})^{s'_{1,j}}, g_1^{r}\right)} \\
&= \frac{e\left(g_1^{s'_{1,j}}, g_2^{y} \cdot \prod_{i=1}^{n} H(i||v_{i,k_i})^{r}\right)}{e\left(\prod_{i=1}^{n} H(i||v_{i,t})^{s'_{1,j}}, g_1^{r}\right)} \\
&= e\left(g_1^{s'_{1,j}}, g_2^{y}\right) = \left(e\left(g_1, g_2\right)^{y}\right)^{s'_{1,j}} = C_{\Delta,w_j,T_l}.
\end{aligned}$$

The correctness of (2) is shown in the following:

$$\begin{aligned}
&\widetilde{C}'_{w_j,T_l} \cdot \frac{e\left(\prod_{i=1}^{n} C'_{i,t,0,w_j,T_l}, D_0\right) . e\left(\prod_{i=1}^{n} \widehat{C}'_{i,t,0,w_j,T_l}, \widehat{D}_0\right)}{e\left(C'_{1,w_j,T_l}, \prod_{i=1}^{n} D_{i,1}\right) . e\left(\widehat{C}'_{1,w_j,T_l}, \prod_{i=1}^{n} \widehat{D}_{i,1}\right)} \\
&= \widetilde{C}'_{w_j,T_l} \cdot \frac{e\left(\prod_{i=1}^{n} \left(C'_{i,t,0,w_j}\right)^{r''_{w_j}}, g_2^{\lambda}\right)}{e\left(\left(C'_{1,w_j}\right)^{r''_{w_j}}, \prod_{i=1}^{n} g_1^{r_i} . H\left(0\,||i||\,v_{i,k_i}\right)^{\lambda}\right)} \\
&\quad \cdot \frac{e\left(\prod_{i=1}^{n} \left(\widehat{C}'_{i,t,0,w_j}\right)^{r''_{w_j}}, g_1^{\hat{\lambda}}\right)}{e\left(\left(\widehat{C}'_{1,w_j}\right)^{r''_{w_j}}, \prod_{i=1}^{n} g_2^{r_i} . H\left(1\,||i||\,v_{i,k_i}\right)^{\hat{\lambda}}\right)} \\
&= \widetilde{C}'_{w_j,T_l} \cdot \frac{e\left(\prod_{i=1}^{n} \left(\sigma'_{i,j,0} . H\left(0\,||i||\,v_{i,t}\right)^{s''_{2,j}}\right)^{r''_{w_j}}, g_2^{\lambda}\right)}{e\left(\left(g_2^{s''_{2,j}}\right)^{r''_{w_j}}, \prod_{i=1}^{n} g_1^{r_i} . H\left(0\,||i||\,v_{i,k_i}\right)^{\lambda}\right)} \\
&\quad \cdot \frac{e\left(\prod_{i=1}^{n} \left(\sigma'_{i,j,1} . H(1||i||v_{i,t})^{s_{2,j}-s''_{2,j}}\right)^{r''_{w_j}}, g_1^{\hat{\lambda}}\right)}{e\left(\left(g_1^{s_{2,j}-s''_{2,j}}\right)^{r''_{w_j}}, \prod_{i=1}^{n} g_2^{r_i} . H\left(1\,||i||\,v_{i,k_i}\right)^{\hat{\lambda}}\right)} \\
&= \widetilde{C}'_{w_j,T_l} \cdot \frac{e\left(\prod_{i=1}^{n} H\left(0\,||i||\,v_{i,t}\right), g_2^{\lambda . r''_{w_j} . s''_{2,j}}\right)}{e\left(g_2^{r''_{w_j} . s''_{2,j}}, g_1^{y} \cdot \prod_{i=1}^{n} H\left(0\,||i||\,v_{i,k_i}\right)^{\lambda}\right)} \\
&\quad \cdot \frac{e\left(\prod_{i=1}^{n} H(1||i||v_{i,t}), g_1^{\hat{\lambda} . r''_{w_j} . \left(s_{2,j}-s''_{2,j}\right)}\right)}{e\left(g_1^{r''_{w_j} . \left(s_{2,j}-s''_{2,j}\right)}, g_2^{y} . \prod_{i=1}^{n} H\left(1\,||i||\,v_{i,k_i}\right)^{\hat{\lambda}}\right)} \\
&= \widetilde{C}'_{w_j,T_l} \cdot \frac{1}{e\left(g_1^{r''_{w_j} . s_{2,j}}, g_2^{y}\right)} = \frac{Y^{r_{w_j}} . \left(\widetilde{C}'_{w_j}\right)^{r''_{w_j}}}{e\left(g_1^{r''_{w_j} . s_{2,j}}, g_2^{y}\right)} \\
&= \frac{Y^{r_{w_j}} . Y^{r''_{w_j} . s_{2,j}}}{e\left(g_1^{r''_{w_j} . s_{2,j}}, g_2^{y}\right)} = Y^{r_{w_j}}.
\end{aligned}$$

For showing the correctness of (3) we have:

$$\begin{aligned}
&\frac{e\left(C_{1,w_j,T_l}, \prod_{i=1}^{n} D_{i,1}\right) . e\left(\widehat{C}_{1,w_j,T_l}, \prod_{i=1}^{n} \widehat{D}_{i,1}\right)}{e\left(\prod_{i=1}^{n} C_{i,t,0,w_j,T_l}, D_0\right) . e\left(\prod_{i=1}^{n} \widehat{C}_{i,t,0,w_j,T_l}, \widehat{D}_0\right)} \\
&= \frac{e\left(g_2^{s''_{1,j}}, \prod_{i=1}^{n} g_1^{r_i} . H\left(0\,||i||\,v_{i,k_i}\right)^{\lambda}\right)}{e\left(\prod_{i=1}^{n} \sigma_{i,j,0} . H\left(0\,||i||\,v_{i,t}\right)^{s''_{1,j}}, g_2^{\lambda}\right)} \\
&\quad \cdot \frac{e\left(g_1^{s_{1,j}-s''_{1,j}}, \prod_{i=1}^{n} g_2^{r_i} . H\left(1\,||i||\,v_{i,k_i}\right)^{\hat{\lambda}}\right)}{e\left(\prod_{i=1}^{n} \sigma_{i,j,1} . H\left(1\,||i||\,v_{i,t}\right)^{s_{1,j}-s''_{1,j}}, g_1^{\hat{\lambda}}\right)} \\
&= \frac{e\left(g_2^{s''_{1,j}}, g_1^{y} \cdot \prod_{i=1}^{n} H\left(0\,||i||\,v_{i,k_i}\right)^{\lambda}\right)}{e\left(\prod_{i=1}^{n} H\left(0\,||i||\,v_{i,t}\right)^{s''_{1,j}}, g_2^{\lambda}\right)} \\
&\quad \cdot \frac{e\left(g_1^{s_{1,j}-s''_{1,j}}, g_2^{y} \cdot \prod_{i=1}^{n} H\left(1\,||i||\,v_{i,k_i}\right)^{\hat{\lambda}}\right)}{e\left(\prod_{i=1}^{n} H\left(1\,||i||\,v_{i,t}\right)^{s_{1,j}-s''_{1,j}}, g_1^{\hat{\lambda}}\right)} \\
&= e\left(g_1^{s_{1,j}}, g_2^{y}\right) = Y^{s_{1,j}}.
\end{aligned}$$

Then, the data decryption key dk_{T_l} is retrieved using (4) as follows,

$$\frac{\widetilde{C}_{w_j,T_l}}{F\left(Y^{r_{w_j}}\right) . F\left(Y^{s_{1,j}}\right)} = \frac{F\left(Y^{r_{w_j}}\right) . dk_{T_l} . F\left(Y^{s_{1,j}}\right)}{F\left(Y^{r_{w_j}}\right) . F\left(Y^{s_{1,j}}\right)} = dk_{T_l}$$

Finally, the message M can be recovered using (5) as follows,

$$\begin{aligned}
&\frac{e\left(U_0^{T_l}, dk_{T_l}\right)}{e\left(PP_1, U_1^{T_l}\right)} = \frac{e\left(g_3^{r_d+r'}, g_4^{mk_0} . g_3^{mk_1.sk} . g_3^{mk_1.S_{T_l}}\right)}{e\left(g_3^{mk_1}, g_3^{(r_d+r').sk} . \left(U_0^{T_l}\right)^{S_{T_l}}\right)} \\
&= \frac{e\left(g_3^{r_d+r'}, g_4^{mk_0}\right) . e\left(g_3^{r_d+r'}, g_3^{mk_1.sk}\right) . e\left(g_3^{r_d+r'}, g_3^{mk_1.S_{T_l}}\right)}{e\left(g_3^{mk_1}, g_3^{(r_d+r').sk}\right) . e\left(g_3^{mk_1}, \left(g_3^{r_d+r'}\right)^{S_{T_l}}\right)} \\
&= e\left(g_3, g_4\right)^{mk_0 \cdot (r_d+r')} = A,
\end{aligned}$$

and then,

$$M = \frac{V^{T_l}}{\frac{e\left(U_0^{T_l}, dk_{T_l}\right)}{e\left(PP_1, U_1^{T_l}\right)}} = \frac{M.e\left(g_3^{mk_0}, g_4^{r_d+r'}\right)}{A}.$$

6 Security Proof

In this section, we prove the security of the HUAP scheme in the random oracle model. We prove that our proposed scheme is secure according to Definition 1. This security proof extends the security proof presented in [7].

Theorem 1. *The HUAP scheme is IND-sCP-CPA secure in the random oracle model such that* $\epsilon_{CPA} \leq 2\epsilon_{DBDH} + n.\epsilon_{DL}$*, where* ϵ_{CPA} *denotes the advantage of a polynomial-time adversary* $\mathcal{A}$ *to win the IND-sCP-CPA game,* ϵ_{DBDH} *and* ϵ_{DL} *respectively denote the advantage of a distinguisher of a DBDH challenge and a D-Linear challenge, and* n *represents the total number of attributes in the universe.*

Proof. The proof of this theorem is provided through three lemmas. In this proof, we marginally alter the original game G into a sequence of hybrid games denoted by $\{G'_0, G_0, G_1, \ldots, G_{h_{max}}\}$. Firstly, in Lemma 1, to establish the first hybrid game G'_0, we embed a DBDH challenge into the ciphertext by substituting the challenge ciphertext component V with a random element in $\mathbb{G}_T$, while the other components are generated in a routine manner. In this fashion, we construct a distinguisher of the DBDH challenge with the help of the distinguisher of G and G'_0. Subsequently, in Lemma 2, to form the next hybrid game G_0, we embed another DBDH challenge into the ciphertext by substituting the challenge ciphertext component $\widetilde{C}_{w^*_\nu}$ with a random element in $\mathbb{G}$. Afterward, in Lemma 3, we establish a series of hybrid games denoted by G_h, where $1 \leq h \leq h_{max}$, by embedding D-Linear challenges into the corresponding ciphertexts as follows. Suppose that h_{max} is the number of attribute values like $v_{i,t}$ satisfying $(v_{i,t} \in W^*_{0,i} \wedge v_{i,t} \notin W^*_{1,i})$ or $(v_{i,t} \notin W^*_{0,i} \wedge v_{i,t} \in W^*_{1,i})$. For each of these attribute values, we modify the game G_{h-1} into a game G_h by substituting the ciphertext components $\{\{C_{i,t,\Delta,w^*_\nu}, C_{i,t,0,w^*_\nu}, \widehat{C}_{i,t,0,w^*_\nu}, C'_{i,t,0,w^*_\nu}, \widehat{C}'_{i,t,0,w^*_\nu}\}_{1\leq t\leq n_i}\}_{1\leq i\leq n}$ with random elements, while other components are generated normally. The process is repeated until there is no attribute value $v_{i,t}$ satisfying $(v_{i,t} \in W^*_{0,i} \wedge v_{i,t} \notin W^*_{1,i})$ or $(v_{i,t} \notin W^*_{0,i} \wedge v_{i,t} \in W^*_{1,i})$. In the last game denoted by $G_{h_{max}}$, the challenge ciphertext components are chosen independently from the random bit ν, and hence the adversary $\mathcal{A}$ has not any advantage in winning the game.

Through Lemma 1 to Lemma 3, we prove that $|Pr[\mathcal{E}] - Pr[\mathcal{E}'_0]| \leq \epsilon_{DBDH}$, $|Pr[\mathcal{E}'_0] - Pr[\mathcal{E}_0]| \leq \epsilon_{DBDH}$, and $|Pr[\mathcal{E}_{h-1}] - Pr[\mathcal{E}_h]| \leq \epsilon_{DL}$ for $1 \leq h \leq h_{max}$, where $\mathcal{E}$, $\mathcal{E}'_0$, and $\mathcal{E}_h$ respectively denote the event that $\mathcal{A}$ wins the game G, G'_0, and G_h. Thus, $\epsilon_{CPA} = \left|Pr[\mathcal{E}] - \frac{1}{2}\right| = |Pr[\mathcal{E}] - Pr[\mathcal{E}_{h_{max}}]|$, and from the triangle inequality, we have:

$$\epsilon_{CPA} \leq |Pr[\mathcal{E}] - Pr[\mathcal{E}'_0]| + |Pr[\mathcal{E}'_0] - Pr[\mathcal{E}_0]| + \sum_{h=1}^{h_{max}} |Pr[\mathcal{E}_{h-1}] - Pr[\mathcal{E}_h]|.$$

Therefore, it is obviously concluded that $\epsilon_{CPA} \leq 2\epsilon_{DBDH} + n.\epsilon_{DL}$. □

Lemma 1. *Under the DBDH assumption, the difference between advantages of* $\mathcal{A}$ *in games* G *and* G'_0 *is negligible such that* $|Pr[\mathcal{E}] - Pr[\mathcal{E}'_0]| \leq \epsilon_{DBDH}$.

Proof. We show that if $\epsilon'_0 = |Pr[\mathcal{E}] - Pr[\mathcal{E}'_0]|$ is not negligible, then there exists a simulator $\mathcal{S}'_0$ that can break the DBDH assumption. To construct $\mathcal{S}'_0$, suppose that it is given a DBDH instance $[g, A, B, C, Z] = [g, g^a, g^b, g^c, Z]$ by the challenger where $g, A, B, C \in \mathbb{G}$ and $Z \in \mathbb{G}_T$, and at the other side, it plays the role of a challenger for the adversary $\mathcal{A}$. In this manner, $\mathcal{S}'_0$ will be able to win the DBDH game with the non-negligible advantage ϵ'_0 by exploiting $\mathcal{A}$. Accordingly, the simulator $\mathcal{S}'_0$ acts as follows.

Init: The adversary $\mathcal{A}$ gives $\mathcal{S}'_0$ two challenge access policies $A^*_0 = W^*_0 = \left[W^*_{0,1}, \ldots, W^*_{0,n}\right]$ and $A^*_1 = W^*_1 = \left[W^*_{1,1}, \ldots, W^*_{1,n}\right]$. Afterward, $\mathcal{S}'_0$ selects a random bit $\nu \in \{0, 1\}$.

Setup: The simulator $\mathcal{S}'_0$ selects $g_1, g_2 \in_R \mathbb{G}$, and $\omega, y, sk, mk_1 \in_R Z^*_p$, sets $g_3 = g^\omega$, $g_4 = B$, and computes $Y = e(g_1, g_2)^y$, $Q_0 = g_3^{sk}$, $PP_1 = g_3^{mk_1}$, $PP_0 = e(g_3, g_4)^{mk_0} = e\left(g^{mk_0}, B\right)^\omega = e(A, B)^\omega$, which implies that $mk_0 = a$. Afterwards, $\mathcal{S}'_0$ sends the system public key $PK = \langle g, g_1, g_2, g_3, g_4, Y\rangle$ and the data public parameter $PP = (Q_0, PP_0, PP_1)$ to $\mathcal{A}$.

Phase 1: The simulator $\mathcal{S}'_0$ answers $\mathcal{A}$'s queries by simulating $\mathcal{O}_{Hash}$ and $\mathcal{O}_{AttrKeyGen}$ as follows.

- **Hash query** $\mathcal{O}_{Hash}(\mathcal{M})$**:** Firstly, the simulator $\mathcal{S}'_0$ selects an empty list $\mathcal{L}_H$. When the adversary $\mathcal{A}$ queries the random oracle $\mathcal{O}_{Hash}(.)$ for an input $\mathcal{M}$, $\mathcal{S}'_0$ checks in the list $\mathcal{L}_H$ if the value of $H(\mathcal{M})$ has been defined. If there exists a record in $\mathcal{L}_H$ associated to the queried point $\mathcal{M}$, $\mathcal{S}'_0$ returns the previously defined value. Otherwise, $\mathcal{S}'_0$ selects $\{\{\tau_{i,t}, a_{i,t}, b_{i,t} \in_R Z_p\}_{1\leq t\leq n_i}\}_{1\leq i\leq n}$, and calculates the output as follows.

  a) For $\mathcal{M} = (i||v_{i,t})$, $\mathcal{S}'_0$ sets $\alpha = \tau_{i,t}$, and returns $H(\mathcal{M}) = g^\alpha$.

b) For $\mathcal{M} = (0||i||v_{i,t})$, $\mathcal{S}_0'$ sets $\alpha = a_{i,t}$, and returns $H(\mathcal{M}) = g^{\alpha}$.
c) For $\mathcal{M} = (1||i||v_{i,t})$, $\mathcal{S}_0'$ sets $\alpha = b_{i,t}$, and returns $H(\mathcal{M}) = g^{\alpha}$.
Finally, $\mathcal{S}_0'$ adds $\langle \mathcal{M}, \alpha, H(\mathcal{M}) \rangle$ to the list $\mathcal{L}_H$.

• **AttrKeyGen query $\mathcal{O}_{AttrKeyGen}(L)$:** When the adversary $\mathcal{A}$ queries the random oracle $\mathcal{O}_{AttrKeyGen}(.)$ for an attribute list $L = [L_1, L_2, \ldots, L_n]$ where $L_i = v_{i,k_i}$ and with the restriction that $(L \nvDash A_0^* \wedge L \nvDash A_1^*)$, the simulator $\mathcal{S}_0'$ firstly selects $r_1, r_2, \ldots, r_n \in_R Z_p$ such that $\sum_{i=1}^{n} r_i = y$. Also, $\mathcal{S}_0'$ selects $r, \lambda, \hat{\lambda} \in_R Z_p$ and $\{\hat{r}_i \in_R Z_p\}_{1 \le i \le n}$, sets $\hat{r} = \sum_{i=1}^{n} \hat{r}_i$, and computes $D_0 = g_2^{\lambda}$, $\widehat{D}_0 = g_1^{\hat{\lambda}}$, $D_{\Delta,0} = g_1^{r}$, $\widehat{D}_{\Delta,0} = g_2^{y-\hat{r}}$. Afterwards, $\mathcal{S}_0'$ computes the rest of the attribute secret key components for $1 \le i \le n$:

$$\left[\begin{array}{l} D_{\Delta,i} = g_2^{\hat{r}_i}.H(i||v_{i,k_i})^{r} = g_2^{\hat{r}_i}.g^{r.\tau_{i,k_i}}, \\ D_{i,1} = g_1^{r_i}.H(0||i||v_{i,k_i})^{\lambda} = g_1^{r_i}.g^{\lambda.a_{i,k_i}}, \\ \widehat{D}_{i,1} = g_2^{r_i}.H(1||i||v_{i,k_i})^{\hat{\lambda}} = g_2^{r_i}.g^{\hat{\lambda}.b_{i,k_i}} \end{array}\right].$$

Finally, the following attribute secret key is returned: $SK_L = \langle D_0, \widehat{D}_0, D_{\Delta,0}, \widehat{D}_{\Delta,0}, \{D_{\Delta,i}, D_{i,1}, \widehat{D}_{i,1}\}_{1 \le i \le n} \rangle$.

Challenge: The adversary $\mathcal{A}$ sends two different equal-length messages M_0 and M_1 to the simulator $\mathcal{S}_0'$. This latter sets $V = M_{\nu}.Z^{\omega}$ and hence when $Z = e(g,g)^{abc}$, we have $V = M_{\nu}.e(g,g)^{abc\omega} = M_{\nu}.e(g_3,g_4)^{ac} = M_{\nu}.e(g_3,g_4)^{mk_0.c} = M_{\nu}.PP_0^{r_d}$, which implies that $r_d = c$. Afterwards, $\mathcal{S}_0'$ calculates $U_0 = g_3^{r_d} = g^{r_d.\omega} = (g^{r_d})^{\omega} = (g^c)^{\omega}$, $U_1 = g_3^{r_d.sk} = g^{r_d.sk.\omega} = (g^{r_d})^{sk.\omega} = (g^c)^{sk.\omega}$, $dk_{T_0} = g_4^{mk_0}.g_3^{sk.mk_1}.g_3^{mk_1.S_{T_0}} = g_3^{sk.mk_1}.g^{\omega.mk_1.S_{T_0}+b.a} = g_3^{sk.mk_1}.g^{\alpha}$ where $\alpha \in_R Z_p^*$, and sets $S_{T_0} = {}^{(\alpha - b.a)}/_{(\omega.mk_1)}$. Moreover, $\mathcal{S}_0'$ selects $s_1, s_1', s_1'', s_2, s_2'' \in_R Z_p^*$ and computes $\widetilde{C}'_{w_{\nu}^*} = 1.Y^{s_2}$, $C'_{1,w_{\nu}^*} = g_2^{s_2''}$, $\widehat{C}'_{1,w_{\nu}^*} = g_1^{s_2 - s_2''}$, $\widetilde{C}_{w_{\nu}^*} = dk_{T_0}.F(Y^{s_1})$, $C_{\Delta,w_{\nu}^*} = Y^{s_1'}$, $\widehat{C}_{0,w_{\nu}^*} = g_1^{s_1'}$, $C_{1,w_{\nu}^*} = g_2^{s_1''}$, $\widehat{C}_{1,w_{\nu}^*} = g_1^{s_1 - s_1''}$. Then, $\mathcal{S}_0'$ picks $\{\sigma_{i,\Delta}, \sigma_{i,0}, \sigma_{i,1}, \sigma'_{i,0}, \sigma'_{i,1} \in_R G | 1 \le i \le n\}$ such that $\prod_{i=1}^{n} \sigma_{i,\Delta} = \prod_{i=1}^{n} \sigma_{i,0} = \prod_{i=1}^{n} \sigma_{i,1} = \prod_{i=1}^{n} \sigma'_{i,0} = \prod_{i=1}^{n} \sigma'_{i,1} = 1_{\mathbb{G}}$, and computes $[C_{i,t,\Delta,w_{\nu}^*}, C_{i,t,0,w_{\nu}^*}, \widehat{C}_{i,t,0,w_{\nu}^*}, C'_{i,t,0,w_{\nu}^*}, \widehat{C}'_{i,t,0,w_{\nu}^*}]$ as follows.

a) If $v_{i,t} \in W_{\nu,i}^*$, then

$$\left[\begin{array}{l} C_{i,t,\Delta,w_{\nu}^*} = \sigma_{i,\Delta}.H(i||v_{i,t})^{s_1'} = \sigma_{i,\Delta}.g^{\tau_{i,t}.s_1'}, \\ C_{i,t,0,w_{\nu}^*} = \sigma_{i,0}.H(0||i||v_{i,t})^{s_1''} = \sigma_{i,0}.g^{a_{i,t}.s_1''}, \\ \widehat{C}_{i,t,0,w_{\nu}^*} = \sigma_{i,1}.H(1||i||v_{i,t})^{(s_1 - s_1'')} \\ \qquad\qquad = \sigma_{i,1}.g^{b_{i,t}.(s_1 - s_1'')}, \\ C'_{i,t,0,w_{\nu}^*} = \sigma'_{i,0}.H(0||i||v_{i,t})^{s_2''} = \sigma'_{i,0}.g^{a_{i,t}.s_2''}, \\ \widehat{C}'_{i,t,0,w_{\nu}^*} = \sigma'_{i,1}.H(1||i||v_{i,t})^{(s_2 - s_2'')} \\ \qquad\qquad = \sigma'_{i,1}.g^{b_{i,t}.(s_2 - s_2'')} \end{array}\right].$$

b) If $v_{i,t} \notin W_{\nu,i}^*$, then $[C_{i,t,\Delta,w_{\nu}^*}, C_{i,t,0,w_{\nu}^*}, \widehat{C}_{i,t,0,w_{\nu}^*}, C'_{i,t,0,w_{\nu}^*}, \widehat{C}'_{i,t,0,w_{\nu}^*}]$ are chosen randomly from $\mathbb{G}$.

Finally, the simulator $\mathcal{S}_0'$ returns the following challenge ciphertext of M_{ν} with respect to A_{ν}^*:

$$CT_{cloud}^{A_{\nu}^*} = \left(\begin{array}{l} U_0, U_1, V, \widetilde{C}'_{w_{\nu}^*}, C'_{1,w_{\nu}^*}, \widehat{C}'_{1,w_{\nu}^*}, \\ \{\{ C'_{i,t,0,w_{\nu}^*}, \widehat{C}'_{i,t,0,w_{\nu}^*}\}_{1 \le t \le n_i}\}_{1 \le i \le n}, \\ \widetilde{C}_{w_{\nu}^*}, C_{\Delta,w_{\nu}^*}, \widehat{C}_{0,w_{\nu}^*}, C_{1,w_{\nu}^*}, \widehat{C}_{1,w_{\nu}^*}, \\ \{\{ C_{i,t,\Delta,w_{\nu}^*}, C_{i,t,0,w_{\nu}^*}, \widehat{C}_{i,t,0,w_{\nu}^*}\}_{1 \le t \le n_i}\}_{1 \le i \le n} \end{array}\right).$$

Phase 2: The adversary $\mathcal{A}$ continues to query the oracles as in Phase 1.

Guess: The adversary $\mathcal{A}$ guesses a bit ν' as the value of ν. $\mathcal{A}$ wins the game if $\nu' = \nu$. When $\mathcal{A}$ wins the game, $\mathcal{S}_0'$ returns 1 and otherwise returns 0 to the DBDH challenger. Indicate that if $Z = e(g,g)^{abc}$, then $\mathcal{A}$ is in game G, and otherwise, Z is a random element in $\mathbb{G}_T$ and $\mathcal{A}$ is in game G_0'. Therefore the advantage of $\mathcal{S}_0'$ in the DBDH game is equal to ϵ_0', where ϵ_0' is the difference between the advantages of $\mathcal{A}$ to win the game G and the game G_0'. Finally, with respect to the DBDH assumption we have $|Pr[\mathcal{E}] - Pr[\mathcal{E}_0']| = \epsilon_0' \le \epsilon_{DBDH}$. □

Lemma 2. *Under the DBDH assumption, the difference between advantages of $\mathcal{A}$ in games G_0' and G_0 is negligible such that $|Pr[\mathcal{E}_0'] - Pr[\mathcal{E}_0]| \le \epsilon_{DBDH}$.*

Proof. We show that if $\epsilon_0 = |Pr[\mathcal{E}_0'] - Pr[\mathcal{E}_0]|$ is not negligible, then there exists a simulator $\mathcal{S}_0$ that can break the DBDH assumption. To construct $\mathcal{S}_0$, suppose that it is given a DBDH instance $[g, A, B, C, Z] = [g, g^a, g^b, g^c, Z]$ by the challenger where $g, A, B, C \in \mathbb{G}$ and $Z \in \mathbb{G}_T$, and at the other side, it plays the role of a challenger for the adversary $\mathcal{A}$. In this manner, $\mathcal{S}_0$ will be able to win the DBDH game with the non-negligible advantage ϵ_0 by exploiting $\mathcal{A}$. Accordingly, the simulator $\mathcal{S}_0$ acts as follows.

Init: The adversary $\mathcal{A}$ gives $\mathcal{S}_0$ two challenge access policies $A_0^* = W_0^* = [W_{0,1}^*, \ldots, W_{0,n}^*]$ and $A_1^* = W_1^* = [W_{1,1}^*, \ldots, W_{1,n}^*]$, and a timestamp T_l. Afterward, $\mathcal{S}_0$ selects a random bit $\nu \in \{0,1\}$.

Setup: The simulator $\mathcal{S}_0$ selects $RK, S_{T_0}, \omega, mk_0, mk_1, sk \in_R Z_p$, and $g_3, g_4 \in_R \mathbb{G}$, sets $g_1 = g^{\omega}$, $g_2 = B$ and computes $Q_0 = g_3^{sk}$, $PP_0 = e(g_3,g_4)^{mk_0}$, $PP_1 = g_3^{mk_1}$, $Y = e(g_1,g_2)^{y} = e(g^y,B)^{\omega} = e(A,B)^{\omega}$, which implies that $y = a$. Afterwards, $\mathcal{S}_0$ sends the system public key $PK = \langle g, g_1, g_2, g_3, g_4, Y \rangle$ and the data public parameter $PP = (Q_0, PP_0, PP_1)$ to $\mathcal{A}$.

Phase 1: The simulator $\mathcal{S}_0$ answers $\mathcal{A}$'s queries by simulating $\mathcal{O}_{Hash}$ and $\mathcal{O}_{AttrKeyGen}$ as follows.

• **Hash query $\mathcal{O}_{Hash}(\mathcal{M})$:** Firstly, the simulator $\mathcal{S}_0$ selects an empty list $\mathcal{L}_H$. When the adversary $\mathcal{A}$ queries the random oracle $\mathcal{O}_{Hash}(.)$ for an input $\mathcal{M}$, $\mathcal{S}_0$ checks in the list $\mathcal{L}_H$ if the value of $H(\mathcal{M})$ has been defined. If there exists a record in $\mathcal{L}_H$ associated with the queried point $\mathcal{M}$, $\mathcal{S}_0$ returns the previously defined value. Otherwise, $\mathcal{S}_0$ selects $\{\{\tau_{i,t}, a_{i,t}, b_{i,t} \in_R Z_p\}_{1\le t\le n_i}\}_{1\le i\le n}$, and calculates the output as follows,

a) For $\mathcal{M} = (i||v_{i,t})$, $\mathcal{S}_0$ sets $\alpha = \tau_{i,t}$, returns $H(\mathcal{M}) = g^{\alpha}$ if $v_{i,t} \in W^*_{\nu,i}$, or $H(\mathcal{M}) = g_2^{\alpha}$ if $v_{i,t} \notin W^*_{\nu,i}$.

b) For $\mathcal{M} = (0||i||v_{i,t})$, $\mathcal{S}_0$ sets $\alpha = a_{i,t}$, and returns $H(\mathcal{M}) = g_2^{\alpha}$.

c) For $\mathcal{M} = (1||i||v_{i,t})$, $\mathcal{S}_0$ sets $\alpha = b_{i,t}$, returns $H(\mathcal{M}) = g^{\alpha}$ if $v_{i,t} \in W^*_{\nu,i}$, or $H(\mathcal{M}) = g_2^{\alpha}$ if $v_{i,t} \notin W^*_{\nu,i}$.

Finally, $\mathcal{S}_0$ adds $\langle \mathcal{M}, \alpha, H(\mathcal{M})\rangle$ to the list $\mathcal{L}_H$.

• **AttrKeyGen query $\mathcal{O}_{AttrKeyGen}(L)$:** Suppose that the adversary $\mathcal{A}$ queries the random oracle $\mathcal{O}_{AttrKeyGen}(.)$ for an attribute list $L = [L_1, L_2, \ldots, L_n]$ where $L_i = v_{i,k_i}$ and with the restriction that $(L \not\models A_0^* \wedge L \not\models A_1^*)$. In such a case, there must be an integer $j \in \{1, 2, \ldots, n\}$ such that $L_j \notin W^*_{\nu,j}$. $\mathcal{S}_0$ firstly picks $\{\hat{r}'_i \in_R Z_p\}_{1\le i\le n}$, sets $\hat{r}_j = a + \hat{r}'_j$, and for $1 \le i \le n$ sets $\hat{r}_i = \hat{r}'_i$ if $i \ne j$. Afterwards, $\mathcal{S}_0$ sets $\hat{r} = \sum_{i=1}^{n} \hat{r}_i = a + \sum_{i=1}^{n} \hat{r}'_i$ and computes $\widehat{D}_{\Delta,0} = \prod_{i=1}^{n} g_2^{-\hat{r}'_i} = g_2^{-\sum_{i=1}^{n} \hat{r}'_i} = g_2^{a-\hat{r}} = g_2^{y-\hat{r}}$. Furthermore, $\mathcal{S}_0$ selects $r_i \in_R Z_p$ for $i \ne j$ and sets $r_j = a - \sum_{i=1, i\ne j}^{n} r_i \ mod\ p$. Then, $\mathcal{S}_0$ computes $\{D_{\Delta,i}, D_{i,1}, \widehat{D}_{i,1}\}_{1\le i\le n}$ as follows.

a) If $i = j$, then $\mathcal{S}_0$ selects $\beta, \lambda, \hat{\lambda}' \in_R Z_p$ and computes the following components:

$$\begin{cases} D_{\Delta,j} &= g_2^{\hat{r}_j}.H\left(j||v_{j,k_j}\right)^r = g_2^{\hat{r}'_j+a}.\left(g_2^{\tau_{j,k_j}}\right)^r \\ &= g_2^{\hat{r}'_j+a+r\tau_{j,k_j}} = g_2^{\hat{r}'_j+\beta}, \\ D_{j,1} &= g_1^{r_j}.H\left(0||j||v_{j,k_j}\right)^{\lambda} = g_1^{a-\sum_{i=1,i\ne j}^{n} r_i}.g_2^{a_{j,k_j}\lambda} \\ &= (g^a)^{\omega}.g_2^{a_{j,k_j}\lambda}.\prod_{i=1,i\ne j}^{n} g_1^{-r_i}, \\ \widehat{D}_{j,1} &= g_2^{r_j}.H\left(1||j||v_{j,k_j}\right)^{\hat{\lambda}} = g_2^{r_j}.g_2^{\hat{\lambda}b_{j,k_j}} \\ &= g_2^{r_j+b_{j,k_j}.\left(\hat{\lambda}'-\frac{a}{b_{j,k_j}}\right)} = g_2^{r_j-a+\hat{\lambda}'b_{j,k_j}} \\ &= g_2^{\hat{\lambda}'.b_{j,k_j}}.\prod_{i=1,i\ne j}^{n} g_2^{-r_i} \end{cases}$$

where $r = {}^{(\beta-a)}/_{\tau_{j,k_j}}$ and $\hat{\lambda} = \hat{\lambda}' - \left({}^{a}/_{b_{j,k_j}}\right)$.

b) If $i \ne j$, then $\mathcal{S}_0$ computes the following components:

$$\begin{cases} D_{\Delta,i} &= g_2^{\hat{r}_i}.H\left(i||v_{i,k_i}\right)^r = g_2^{\hat{r}'_i}.\left(g^{\tau_{i,k_i}}\right)^r \\ &= g_2^{\hat{r}'_i}.\left(g^{\frac{\beta-a}{\tau_{j,k_j}}}\right)^{\tau_{i,k_i}} \\ &= g_2^{\hat{r}'_i}.g^{\frac{\beta\tau_{i,k_i}}{\tau_{j,k_j}}}.(g^a)^{-\frac{\tau_{i,k_i}}{\tau_{j,k_j}}}, \\ D_{i,1} &= g_1^{r_i}.H\left(0\,||i||\,v_{i,k_i}\right)^{\lambda} = g_1^{r_i}.g_2^{a_{i,k_i}\lambda}, \\ \widehat{D}_{i,1} &= g_2^{r_i}.H\left(1||i||v_{i,k_i}\right)^{\hat{\lambda}} = g_2^{r_i}.g^{\hat{\lambda}b_{i,k_i}} \\ &= g_2^{r_i}.g^{b_{i,k_i}\hat{\lambda}'}.(g^a)^{-\frac{b_{i,k_i}}{b_{j,k_j}}} \end{cases}$$

Subsequently, $\mathcal{S}_0$ computes the rest of the attribute secret key components as follows,

$$\left[\begin{array}{l} D_0 = g_2^{\lambda}, \widehat{D}_0 = g_1^{\hat{\lambda}} = g_1^{\hat{\lambda}'-\left({}^{a}/_{b_{j,k_j}}\right)} = g_1^{\hat{\lambda}'}.(g^a)^{-\frac{\omega}{b_{j,k_j}}}, \\ D_{\Delta,0} = g_1^{r} = g_1^{\frac{\beta-a}{\tau_{j,k_j}}} = g_1^{\frac{\beta}{\tau_{j,k_j}}}.(g^a)^{-\frac{\omega}{\tau_{j,k_j}}} \end{array}\right]$$

Finally, the following attribute secret key is returned: $SK_L = \langle D_0, \widehat{D}_0, D_{\Delta,0}, \widehat{D}_{\Delta,0}, \{D_{\Delta,i}, D_{i,1}, \widehat{D}_{i,1}\}_{1\le i\le n}\rangle$.

• **Reencrypt query $\mathcal{O}_{Reencrypt}\left(T_i, CT^A_{cloud}\right)$:** Suppose that $\mathcal{A}$ submits a ciphertext $CT^A_{cloud} = (CT_{M,cloud} = [U_0, U_1, V], CT^A_{T_0,cloud} = [CT_{W,T_0}, \widetilde{CT}_{W,T_0}])$, and a timestamp T_i where $i \le l$.

Firstly, $\mathcal{S}_0$ checks in the list $\mathcal{L}_{\widehat{H}}$ if the tuple $\langle i, S_{T_i}\rangle$ has been queried before. If it was, the corresponding value is retrieved. Otherwise, $\mathcal{S}_0$ chooses $S_{T_i} \in_R Z_p^*$ and puts the tuple $\langle i, S_{T_i}\rangle$ in the list $\mathcal{L}_{\widehat{H}}$. Afterwards, $\mathcal{S}_0$ chooses $r' \in_R Z_p^*$ and computes $CT_{M,T_i,user}$ as follows,

$$CT_{M,T_i,user} = \left[\begin{array}{l} U_0^{T_i} = U_0.g_3^{r'}, \\ U_1^{T_i} = U_1.g_3^{r'.sk}.\left(U_0^{T_i}\right)^{S_{T_i}}, \\ V^{T_i} = V.e\left(g_3^{mk_0}, g_4^{r'}\right) \end{array}\right].$$

Then, $\mathcal{S}_0$ chooses $r'', r \in_R Z_p^*$ and computes $\widetilde{CT}_{W,T_i}$ and CT_{W,T_i} as follows,

$$\widetilde{CT}_{W,T_i} = \left(\begin{array}{l} \widetilde{C}'_{w,T_i} = Y^r.\left(\widetilde{C}'_w\right)^{r''}, \\ C'_{1,w,T_i} = \left(C'_{1,w}\right)^{r''}, \widehat{C}'_{1,w,T_i} = \left(\widehat{C}'_{1,w}\right)^{r''}, \\ \left\{\left\{\begin{array}{l} C'_{i,t,0,w,T_i} = \left(C'_{i,t,0,w}\right)^{r''}, \\ \widehat{C}'_{i,t,0,w,T_i} = \left(\widehat{C}'_{i,t,0,w}\right)^{r''} \end{array}\right\}_{1\le t\le n_i}\right\}_{1\le i\le n} \end{array}\right),$$

$$CT_{W,T_i} = \left(\begin{array}{l} \widetilde{C}_{w,T_i} = F\left(Y^r\right).g_3^{mk_1.\left(S_{T_i}-S_{T_0}\right)}.\widetilde{C}_w, \\ C_{\Delta,w,T_i} = C_{\Delta,w}, \widehat{C}_{0,w,T_i} = \widehat{C}_{0,w} \\ C_{1,w,T_i} = C_{1,w}, \widehat{C}_{1,w,T_i} = \widehat{C}_{1,w}, \\ \left\{\left\{\begin{array}{l} C_{i,t,\Delta,w,T_i} = C_{i,t,\Delta,w}, \\ C_{i,t,0,w,T_i} = C_{i,t,0,w}, \\ \widehat{C}_{i,t,0,w,T_i} = \widehat{C}_{i,t,0,w} \end{array}\right\}_{1\le t\le n_i}\right\}_{1\le i\le n} \end{array}\right).$$

Finally, the following user ciphertext, which is computed according to the timestamp T_i is returned:

$$CT^{A}_{M,T_i,user} = \begin{pmatrix} CT_{M,T_i,user}, \\ CT^{A}_{T_i,user} = \left[CT_{W,T_i}, \widetilde{CT}_{W,T_i}\right] \end{pmatrix}.$$

Challenge: The adversary $\mathcal{A}$ sends two different equal-length messages M_0 and M_1 to the simulator $\mathcal{S}_0$. The latter sets V to be a random element in $\mathbb{G}_T$. Afterwards, $\mathcal{S}_0$ selects $r_d \in_R Z_p^*$ and sets $U_0 = g_3^{r_d}$, $U_1 = g_3^{r_d.sk}$, $dk_{T_0} = g_4^{mk_0}.g_3^{sk.mk_1}.g_3^{mk_1.S_{T_0}}$. Subsequently, $\mathcal{S}_0$ calculates $\widetilde{C}_{w_\nu^*} = dk_{T_0}.F(Z^\omega)$ and hence when $Z = e(g,g)^{abc}$, we have $\widetilde{C}_{w_\nu^*} = dk_{T_0}.F\left(e(g_1,g_2)^{ac}\right) = dk_{T_0}.F\left(e(g_1,g_2)^{yc}\right) = dk_{T_0}.F\left(Y^{s_1}\right)$, which implies that $s_1 = c$. Furthermore, $\mathcal{S}_0$ selects $s_1', s_1'', s_2, s_2'' \in_R Z_p^*$, and computes $\widetilde{C}'_{w_\nu^*} = 1.Y^{s_2}$, $C'_{1,w_\nu^*} = g_2^{s_2''}$, $\widehat{C}'_{1,w_\nu^*} = g_1^{s_2-s_2''}$, $C_{\Delta,w_\nu^*} = Y^{s_1'}$, $\widehat{C}_{0,w_\nu^*} = g_1^{s_1'}$, $C_{1,w_\nu^*} = g_2^{s_1''}$, $\widehat{C}_{1,w_\nu^*} = g_1^{s_1-s_1''} = \left(g^c.g^{-s''}\right)^\omega$. $\mathcal{S}_0$ also picks $\{\sigma_{i,\Delta}, \sigma_{i,0}, \sigma_{i,1}, \sigma'_{i,0}, \sigma'_{i,1} \in_R G | 1 \le i \le n\}$ such that $\prod_{i=1}^n \sigma_{i,\Delta} = \prod_{i=1}^n \sigma_{i,0} = \prod_{i=1}^n \sigma_{i,1} = \prod_{i=1}^n \sigma'_{i,0} = \prod_{i=1}^n \sigma'_{i,1} = 1_{\mathbb{G}}$, and computes $[C_{i,t,\Delta,w_\nu^*}, C_{i,t,0,w_\nu^*}, \widehat{C}_{i,t,0,w_\nu^*}, C'_{i,t,0,w_\nu^*}, \widehat{C}'_{i,t,0,w_\nu^*}]$ as follows.

a) If $v_{i,t} \in W^*_{\nu,i}$, then

$$\begin{bmatrix} C_{i,t,\Delta,w_\nu^*} = \sigma_{i,\Delta}.H(i||v_{i,t})^{s_1'} = \sigma_{i,\Delta}.g^{\tau_{i,t}.s_1'}, \\ C_{i,t,0,w_\nu^*} = \sigma_{i,0}.H(0||i||v_{i,t})^{s_1''} = \sigma_{i,0}.g_2^{a_{i,t}.s_1''}, \\ \widehat{C}_{i,t,0,w_\nu^*} = \sigma_{i,1}.H(1||i||v_{i,t})^{(s_1-s_1'')} \\ \qquad = \sigma_{i,1}.g^{b_{i,t}.(s_1-s_1'')} \\ \qquad = \sigma_{i,1}.\left(g^c\right)^{b_{i,t}}.g^{-b_{i,t}.s_1''}, \\ C'_{i,t,0,w_\nu^*} = \sigma'_{i,0}.H(0||i||v_{i,t})^{s_2''} = \sigma'_{i,0}.g_2^{a_{i,t}.s_2''}, \\ \widehat{C}'_{i,t,0,w_\nu^*} = \sigma'_{i,1}.H(1||i||v_{i,t})^{(s_2-s_2'')} \\ \qquad = \sigma'_{i,1}.g^{b_{i,t}.(s_2-s_2'')} \end{bmatrix}.$$

b) If $v_{i,t} \notin W^*_{\nu,i}$, then $[C_{i,t,\Delta,w_\nu^*}, C_{i,t,0,w_\nu^*}, \widehat{C}_{i,t,0,w_\nu^*}, C'_{i,t,0,w_\nu^*}, \widehat{C}'_{i,t,0,w_\nu^*}]$ are chosen randomly from $\mathbb{G}$.

Finally, the simulator $\mathcal{S}_0$ returns the following challenge ciphertext of M_ν with respect to A^*_ν:

$$CT^{A^*_\nu}_{cloud} = \begin{pmatrix} U_0, U_1, V, \widetilde{C}'_{w_\nu^*}, C'_{1,w_\nu^*}, \widehat{C}'_{1,w_\nu^*}, \\ \{\{\ C'_{i,t,0,w_\nu^*}, \widehat{C}'_{i,t,0,w_\nu^*}\}_{1\le t\le n_i}\}_{1\le i\le n}, \\ \widetilde{C}_{w_\nu^*}, C_{\Delta,w_\nu^*}, \widehat{C}_{0,w_\nu^*}, C_{1,w_\nu^*}, \widehat{C}_{1,w_\nu^*}, \\ \{\{\ C_{i,t,\Delta,w_\nu^*}, C_{i,t,0,w_\nu^*}, \widehat{C}_{i,t,0,w_\nu^*}\}_{1\le t\le n_i}\}_{1\le i\le n} \end{pmatrix}.$$

Phase 2: The adversary $\mathcal{A}$ continues to query the oracles as in Phase 1.

Guess: The adversary $\mathcal{A}$ guesses a bit ν' as the value of ν. $\mathcal{A}$ wins the game if $\nu' = \nu$. When $\mathcal{A}$ wins the game, $\mathcal{S}_0$ returns 1 and otherwise returns 0 to the DBDH challenger. Indicate that if $Z = e(g,g)^{abc}$, then $\mathcal{A}$ is in game G'_0, and otherwise, Z is a random element in $\mathbb{G}_T$ and $\mathcal{A}$ is in game G_0. Therefore the advantage of $\mathcal{S}_0$ in the DBDH game is equal to ϵ_0, where ϵ_0 is the difference between the advantages of $\mathcal{A}$ to win the game G'_0 and the game G_0. Finally, with respect to the DBDH assumption we have $|Pr[\mathcal{E}'_0] - Pr[\mathcal{E}_0]| = \epsilon_0 \le \epsilon_{DBDH}$. □

Lemma 3. *Under the D-Linear assumption, the difference between advantages of $\mathcal{A}$ in games G_{h-1} and G_h is negligible for $1 \le h \le h_{max}$, such that $|Pr[\mathcal{E}_{h-1}] - Pr[\mathcal{E}_h]| \le \epsilon_{DL}$.*

Proof. We show that if $\epsilon_h = |Pr[\mathcal{E}_{h-1}] - Pr[\mathcal{E}_h]|$ is not negligible, then there exists a simulator $\mathcal{S}_h$ that can break the D-Linear assumption. To construct $\mathcal{S}_h$, suppose that it is given a D-Linear instance $[g, g^{z_1}, g^{z_2}, Z, g^{z_2 z_4}, g^{z_3+z_4}]$ by the challenger where $Z \in \mathbb{G}$ (note that this D-Linear assumption is equivalent to that of Section 3.2 [9]). On the other side, $\mathcal{S}_h$ plays the role of a challenger for the adversary $\mathcal{A}$. In this manner, $\mathcal{S}_h$ will be able to win the D-Linear game with the non-negligible advantage ϵ_h by exploiting $\mathcal{A}$. Accordingly, the simulator $\mathcal{S}_h$ acts as follows.

Init: The adversary $\mathcal{A}$ gives $\mathcal{S}_h$ two challenge access policies $A^*_0 = W^*_0 = [W^*_{0,1}, \ldots, W^*_{0,n}]$ and $A^*_1 = W^*_1 = [W^*_{1,1}, \ldots, W^*_{1,n}]$, and a timestamp T_l. Afterward, $\mathcal{S}_h$ selects a random bit $\nu \in \{0,1\}$. Suppose that the ciphertext components $\{C_{i_h,t_h,\Delta,w_\nu^*}, C_{i_h,t_h,0,w_\nu^*}, \widehat{C}_{i_h,t_h,0,w_\nu^*}, C'_{i_h,t_h,0,w_\nu^*}, \widehat{C}'_{i_h,t_h,0,w_\nu^*}\}$ that are generated in a routine manner in game G_{h-1}, are chosen randomly from $\mathbb{G}$ in game G_h. Also, without loss of generality, assume that $(v_{i_h,t_h} \notin W^*_{0,i_h} \wedge v_{i_h,t_h} \in W^*_{1,i_h})$. Thus, we proceed assuming $\nu = 1$.

Setup: The simulator $\mathcal{S}_h$ selects $RK, S_{T_0}, \omega, mk_0, mk_1, sk \in_R Z_p$, and $g_3, g_4 \in_R \mathbb{G}$, sets $g_1 = g^{z_1}$ and $g_2 = g^{z_2}$, and computes $Q_0 = g_3^{sk}$, $PP_0 = e(g_3,g_4)^{mk_0}$, $PP_1 = g_3^{mk_1}$. Afterwards, $\mathcal{S}_h$ selects $y \in_R Z_p$ and computes $Y = e(g_1,g_2)^y$. Finally, $\mathcal{S}_h$ sends the system public key $PK = \langle g, g_1, g_2, g_3, g_4, Y\rangle$ and the data public parameter $PP = (Q_0, PP_0, PP_1)$ to $\mathcal{A}$.

Phase 1: The simulator $\mathcal{S}_h$ answers $\mathcal{A}$'s queries by simulating $\mathcal{O}_{Hash}$ and $\mathcal{O}_{AttrKeyGen}$ as follows.

- **Hash query $\mathcal{O}_{Hash}(\mathcal{M})$:** Firstly, the simulator $\mathcal{S}_h$ selects an empty list $\mathcal{L}_H$. When the adversary $\mathcal{A}$ queries the random oracle $\mathcal{O}_{Hash}(.)$ for an input $\mathcal{M}$, $\mathcal{S}_h$ checks in the list $\mathcal{L}_H$ if the value of $H(\mathcal{M})$ has been defined. If there exists a record in $\mathcal{L}_H$ associated to the queried point $\mathcal{M}$, $\mathcal{S}_h$ returns the previously defined value. Otherwise, $\mathcal{S}_h$ selects

$\{\{\tau_{i,t}, a_{i,t}, b_{i,t} \in_R Z_p\}_{1\le t\le n_i}\}_{1\le i\le n}$, and calculates the output as follows.

a) For $\mathcal{M} = (i||v_{i,t})$, $\mathcal{S}_h$ sets $\alpha = \tau_{i,t}$, returns $H(\mathcal{M}) = g^{\alpha}$ if $v_{i,t} \in W^*_{\nu,i}$, or $H(\mathcal{M}) = (g^{z_1})^{\alpha}$ if $v_{i,t} \notin W^*_{\nu,i}$.

b) For $\mathcal{M} = (0||i||v_{i,t})$, $\mathcal{S}_h$ sets $\alpha = a_{i,t}$, returns $H(\mathcal{M}) = (g^{z_2})^{\alpha}$ if $v_{i,t} \in W^*_{\nu,i}$, or $H(\mathcal{M}) = g^{\alpha}$ if $v_{i,t} \notin W^*_{\nu,i}$.

c) For $\mathcal{M} = (1||i||v_{i,t})$, $\mathcal{S}_h$ sets $\alpha = b_{i,t}$, returns $H(\mathcal{M}) = (g^{z_1})^{\alpha}$ if $v_{i,t} \in W^*_{\nu,i}$, or $H(\mathcal{M}) = g^{\alpha}$ if $v_{i,t} \notin W^*_{\nu,i}$.

Finally, $\mathcal{S}_h$ adds $\langle \mathcal{M}, \alpha, H(\mathcal{M})\rangle$ to the list $\mathcal{L}_H$.

• **AttrKeyGen query $\mathcal{O}_{AttrKeyGen}(L)$:** When the adversary $\mathcal{A}$ queries the random oracle $\mathcal{O}_{AttrKeyGen}(.)$ for an attribute list $L = [L_1, L_2, \ldots, L_n]$ where $L_i = v_{i,k_i}$ and with the restriction that $(L \nvDash A^*_0 \wedge L \nvDash A^*_1)$, the simulator $\mathcal{S}_h$ firstly selects $r_1, r_2, \ldots, r_n \in_R Z_p$ such that $\sum_{i=1}^{n} r_i = y$. Besides, $\mathcal{S}_h$ selects $r, \lambda, \hat{\lambda} \in_R Z_p$ and $\{\hat{r}_i \in_R Z_p\}_{1\le i\le n}$, sets $\hat{r} = \sum_{i=1}^{n} \hat{r}_i$, and computes $D_{\Delta,0} = g_1^{r}, \widehat{D}_{\Delta,0} = g_2^{y-\hat{r}}, D_0 = g_2^{\lambda}, \widehat{D}_0 = g_1^{\hat{\lambda}}$. Afterwards, assuming $L_i = v_{i,k_i}$ for $1 \le i \le n$, $\mathcal{S}_h$ computes the rest of the attribute secret key components as follows.

a) If $v_{i,k_i} \in W^*_{\nu,i}$, then

$$\begin{bmatrix} D_{\Delta,i} = g_2^{\hat{r}_i}.g^{r.\tau_{i,k_i}}, D_{i,1} = g_1^{r_i}.(g^{z_2})^{\lambda a_{i,k_i}} \\ \widehat{D}_{i,1} = g_2^{r_i}.(g^{z_1})^{\hat{\lambda} b_{i,k_i}} \end{bmatrix}.$$

b) If $v_{i,k_i} \notin W^*_{\nu,i}$, then

$$\begin{bmatrix} D_{\Delta,i} = g_2^{\hat{r}_i}.(g^{z_1})^{r.\tau_{i,k_i}}, D_{i,1} = g_1^{r_i}.g^{\lambda a_{i,k_i}} \\ \widehat{D}_{i,1} = g_2^{r_i}.g^{\hat{\lambda} b_{i,k_i}} \end{bmatrix}.$$

Finally, the following attribute secret key is returned: $SK_L = \langle D_0, \widehat{D}_0, D_{\Delta,0}, \widehat{D}_{\Delta,0}, \{D_{\Delta,i}, D_{i,1}, \widehat{D}_{i,1}\}_{1\le i\le n}\rangle$.

• **Reencrypt query $\mathcal{O}_{Reencrypt}\left(T_i, CT^{A}_{cloud}\right)$:** The simulator $\mathcal{S}_h$ proceeds as in Lemma 2.

Challenge: The adversary $\mathcal{A}$ sends two different equal-length messages M_0 and M_1 to the simulator $\mathcal{S}_h$. This latter sets V and $\widetilde{C}_{w^*_\nu}$ to be random elements in $\mathbb{G}_T$ and $\mathbb{G}$, respectively. Afterwards, $\mathcal{S}_h$ selects $r_d \in_R Z^*_p$, and sets $U_0 = g_3^{r_d}$, $U_1 = g_3^{r_d.sk}$, $dk_{T_0} = g_4^{mk_0}.g_3^{sk.mk_1}.g_3^{mk_1.S_{T_0}}$. Besides, $\mathcal{S}_h$ selects $s'_1, s_2, s''_2 \in_R Z_p$, and computes $\widetilde{C}'_{w^*_\nu} = 1.Y^{s_2}$, $C'_{1,w^*_\nu} = g_2^{s''_2}$, $\widehat{C}'_{1,w^*_\nu} = g_1^{s_2-s''_2}$, $C_{\Delta,w^*_\nu} = Y^{s'_1}$, $\widehat{C}_{0,w^*_\nu} = g_1^{s'_1}$. Moreover, $\mathcal{S}_h$ sets $C_{1,w^*_\nu} = g^{z_2 z_4} = g_2^{s''_1}$ and $\widehat{C}_{1,w^*_\nu} = Z = g^{z_1 z_3} = g_1^{s_1-s''_1}$ which respectively imply that $s''_1 = z_4$ and $s_1 = z_3 + z_4$. Then, $\mathcal{S}_h$ selects $\{\sigma_{i,\Delta}, \sigma_{i,0}, \sigma_{i,1}, \sigma'_{i,0}, \sigma'_{i,1} \in_R \mathbb{G} | 1 \le i \le n\}$ such that $\prod_{i=1}^{n}\sigma_{i,\Delta} = \prod_{i=1}^{n}\sigma_{i,0} = \prod_{i=1}^{n}\sigma_{i,1} = \prod_{i=1}^{n}\sigma'_{i,0} = \prod_{i=1}^{n}\sigma'_{i,1} = 1_{\mathbb{G}}$, and generates the ciphertext components $\{\{C_{i,t,\Delta,w^*_\nu}, C_{i,t,0,w^*_\nu}, \widehat{C}_{i,t,0,w^*_\nu}, C'_{i,t,0,w^*_\nu}, \widehat{C}'_{i,t,0,w^*_\nu}\}_{1\le t\le n_i}\}_{1\le i\le n}$ as in game G_{h-1} while the components $\{C_{i_h,t_h,\Delta,w^*_\nu}, C_{i_h,t_h,0,w^*_\nu}, \widehat{C}_{i_h,t_h,0,w^*_\nu}, C'_{i_h,t_h,0,w^*_\nu}, \widehat{C}'_{i_h,t_h,0,w^*_\nu}\}$ are computed as follows,

$$\begin{bmatrix} C_{i_h,t_h,\Delta,w^*_\nu} = \sigma_{i_h,\Delta}.H(i_h||v_{i_h,t_h})^{s'_1} = \sigma_{i_h,\Delta}.g^{\tau_{i_h,t_h}.s'_1}, \\ C_{i_h,t_h,0,w^*_\nu} = \sigma_{i_h,0}.H(0||i_h||v_{i_h,t_h})^{s''_1} = \sigma_{i_h,0}.g_2^{a_{i_h,t_h}.s''_1} \\ \quad = \sigma_{i_h,0}.(g^{z_2 z_4})^{a_{i_h,t_h}}, \\ \widehat{C}_{i_h,t_h,0,w^*_\nu} = \sigma_{i_h,1}.H(1||i_h||v_{i_h,t_h})^{(s_1-s''_1)} \\ \quad = \sigma_{i_h,1}.(g_1)^{b_{i_h,t_h}.(s_1-s''_1)} \\ \quad = \sigma_{i_h,1}.(g^{z_1 z_3})^{b_{i_h,t_h}} = \sigma_{i_h,1}.Z^{b_{i_h,t_h}}, \\ C'_{i_h,t_h,0,w^*_\nu} = \sigma'_{i_h,0}.H(0||i_h||v_{i_h,t_h})^{s''_2} = \sigma'_{i_h,0}.g_2^{a_{i_h,t_h}.s''_2}, \\ \widehat{C}'_{i_h,t_h,0,w^*_\nu} = \sigma'_{i_h,1}.H(1||i_h||v_{i_h,t_h})^{(s_2-s''_2)} \\ \quad = \sigma'_{i_h,1}.g_1^{b_{i_h,t_h}.(s_2-s''_2)} \end{bmatrix}.$$

Phase 2: The adversary $\mathcal{A}$ continues to query the oracles as in Phase 1.

Guess: The adversary $\mathcal{A}$ guesses a bit ν' as the value of ν. $\mathcal{A}$ wins the game if $\nu' = \nu$. When $\mathcal{A}$ wins the game, $\mathcal{S}_h$ returns 1 and otherwise returns 0 to the D-Linear challenger. Indicate that if $Z = g^{z_1 z_3}$, then $\mathcal{A}$ is in game G_{h-1}, and otherwise, Z is a random element in $\mathbb{G}$ and $\mathcal{A}$ is in game G_h. Therefore the advantage of $\mathcal{S}_h$ in the D-Linear game is equal to ϵ_h, where ϵ_h is the difference between the advantages of $\mathcal{A}$ to win the game G_{h-1} and the game G_h. Finally, with respect to the D-Linear assumption we have $|Pr[\mathcal{E}_{h-1}] - Pr[\mathcal{E}_h]| = \epsilon_h \le \epsilon_{DL}$ for $1 \le h \le h_{max}$. □

7 Performance Evaluation

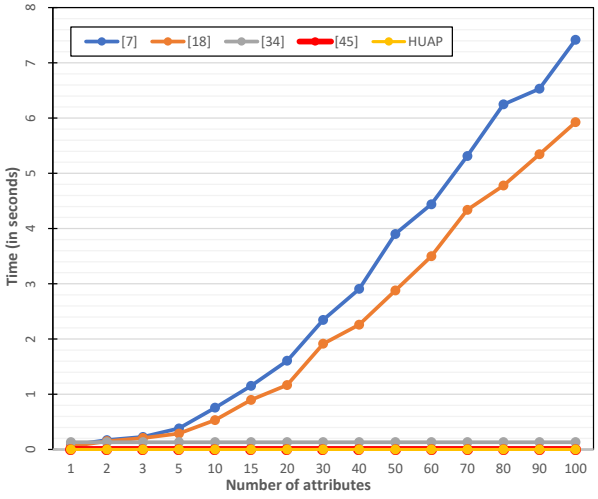
In this section, we evaluate the effectiveness of our proposed scheme by comparing its computational and storage complexities with schemes presented in [7, 18, 34, 45]. The main reason for considering these schemes for comparison is that they are similar to our proposed scheme in several aspects, such as functional capabilities and cryptographic algorithms. In this table, n, m, l, and a denote the total number of attributes in the universe, the number of AND-gates in an access policy, the number of rows in an LSSS access policy matrix, and the maximum number of users in the system, respectively. $N = \sum_{i=1}^{n} n_i$ indicates the total number of possible values of all attributes. Also, M, E, P, and R represent a modular multiplication, a modular exponentiation, a bilinear pairing, and a random element selection, respectively. The results indicate that our scheme performs as efficiently as [45] in terms of online encryption while outperforming the other schemes (i.e. [7, 18, 34]). This is because the other schemes fail to split the encryption function, and therefore the whole encryption function must be executed by the message generator device after the message is ready to be encrypted. Additionally, thanks to the constant number of pairing operations in the decryption algorithm, the decryption efficiency

Table 3. Storage and efficiency comparison

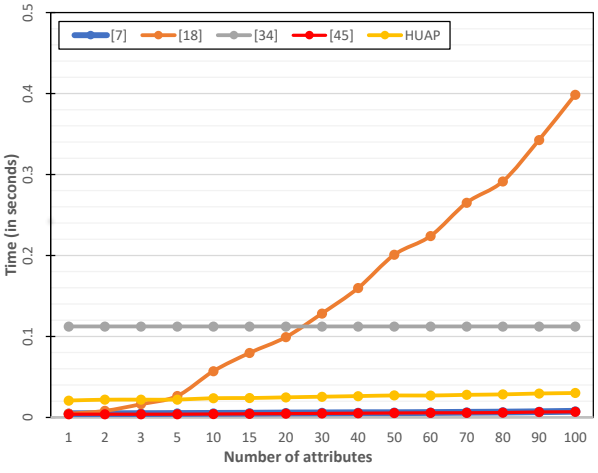
Scheme	System public key size	Attribute secret key size	Ciphertext size	Encryption cost	Decryption cost (decryption phase)
[7]	$3\|\mathbb{G}\|+1\|\mathbb{G}_T\|$	$(3n+4)\|\mathbb{G}\|$	$(3N+3)\|\mathbb{G}\|+2\|\mathbb{G}_T\|$	$M_{\mathbb{G}}(3n+1)+2E_{\mathbb{G}_T}+E_{\mathbb{G}}(3n+3)+R_{\mathbb{G}}(3N)$	$M_{\mathbb{G}}(2n)+4M_{\mathbb{G}_T}+4P$
[18]	$(3N+4)\|\mathbb{G}\|+1\|\mathbb{G}_T\|$	$(4n+4)\|\mathbb{G}\|$	$(3N+3)\|\mathbb{G}\|+2\|\mathbb{G}_T\|$	$M_{\mathbb{G}}(n+1)+2E_{\mathbb{G}_T}+E_{\mathbb{G}}(3n+3)+R_{\mathbb{G}}(3N)$	$M_{\mathbb{G}_T}(3n+1)+P(3n+1)$
[34]	$(2a+11)\|\mathbb{G}\|$	$(5n+2)\|\mathbb{G}\|$	$(6l+3)\|\mathbb{G}\|$	$M_{\mathbb{G}}(2l+2)+1E_{\mathbb{G}_T}+E_{\mathbb{G}}(8l+4)+1P$	$M_{\mathbb{G}_T}(5l+3)$ $+E_{\mathbb{G}_T}(l+2)+P(6l+3)$
[45]	$(n+5)\|\mathbb{G}\|+1\|\mathbb{G}_T\|$	$3\|\mathbb{G}\|+(n+2)\|\mathbb{Z}_p\|$	$(n+4)\|\mathbb{G}\|+1\|\mathbb{G}_T\|+(n+2)\|\mathbb{Z}_p\|$	DO (offline): $M_{\mathbb{G}}(n)+1E_{\mathbb{G}_T}+E_{\mathbb{G}}(2n+2)$ DO (online): $1M_{\mathbb{G}_T}$	$M_{\mathbb{G}}(n)+3M_{\mathbb{G}_T}$ $+E_{\mathbb{G}}(n+3)+3P$
HUAP	$5\|\mathbb{G}\|+1\|\mathbb{G}_T\|$	$(3n+4)\|\mathbb{G}\|$	message ciphertext : $2\|\mathbb{G}\|+1\|\mathbb{G}_T\|$ policy ciphertext : $(5mN+6m)\|\mathbb{G}\|$ $+(2m)\|\mathbb{G}_T\|$	Dev (offline): $1E_{\mathbb{G}_T}+2E_{\mathbb{G}}$ Dev (online): $1M_{\mathbb{G}_T}$ DO: $M_{\mathbb{G}}(5mn+m)+E_{\mathbb{G}_T}(3m)$ $+E_{\mathbb{G}}(5mn+5m)+R_{\mathbb{G}}(5mN)$	$M_{\mathbb{G}}(4n+2)$ $+9M_{\mathbb{G}_T}+10P$

of our scheme outperforms [18] and is comparable to that of [7], [34], and [45].

We have implemented the simulation experiment on a virtual machine equipped with Intel Core i7-3632QM CPU (2 core 2.20 GHz) and 2 GB memory running Linux Kernel 5.4.0. The experiment is implemented with the PBC library of version 0.5.14 for underlying cryptographic operations. The evaluation results of executing encryption and decryption algorithms are presented in Figure 4, where we have set $l = 10$, $a = 10$, and $n_i = 10$ for $1 \leq i \leq n$. To be specific, we have compared the cost of online encryption that is performed by Dev in our scheme with the cost of encryption in other schemes. Moreover, as the number of random element selections in [7], [18], and HUAP appreciably affects the encryption cost, it is enumerated in Table 3 for these schemes. From Figure 4a, the online encryption time in our scheme is constant as the number of attributes in the universe is increased. In the meanwhile, from Figure 4b, in our scheme, growing the total number of attributes in the universe does not change the decryption time significantly, as the number of pairing operations in the decryption algorithm is constant. Note that although the complexity of the encryption/decryption algorithm in [34] remains at a constant value when the total number of attributes in the universe is increased, it grows linearly with respect to the number of rows in the access policy matrix.



(a) Encryption



(b) Decryption

Figure 4. Performance comparison

8 Conclusion

In this paper, a ciphertext-policy attribute-based access control scheme has been proposed. In the proposed scheme, the access policies are hidden, and hence unauthorized data users cannot learn which attribute set satisfies an access policy. The scheme also enables data owners to efficiently outsource a major part of the access policy update process to a cloud service provider. In particular, the process does not require generating a new re-encryption key. Moreover, to reduce the computational cost for resource-constrained devices, this scheme divides the encryption algorithm into two offline and online phases. Furthermore, in the proposed scheme, the decryption process is very fast and requires only a constant number of bilinear pairing operations. The proposed scheme is proven to be secure in the random oracle model. Our simulation results indicate that our proposed scheme effectively decreases computational overhead

at the data collector device and data user sides.

References

[1] M. Ali, S.U. Khan, and A.V. Vasilakos. Security in cloud computing: Opportunities and challenges. *Information sciences*, 305:357–383, 2015.

[2] A. Sahai and B. Waters. Fuzzy identity-based encryption. In *Annual international conference on the theory and applications of cryptographic techniques*, pages 457–473. Springer, 2005.

[3] V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *Proceedings of the 13th ACM conference on Computer and communications security*, pages 89–98, 2006.

[4] J. Bethencourt, A. Sahai, and B. Waters. Ciphertext-policy attribute-based encryption. In *2007 IEEE symposium on security and privacy (SP'07)*, pages 321–334. IEEE, 2007.

[5] Y. Zhang, R.H. Deng, S. Xu, J. Sun, Q. Li, and D. Zheng. Attribute-based encryption for cloud computing access control: A survey. *ACM Computing Surveys (CSUR)*, 53(4):1–41, 2020.

[6] Y. Zhang, D. Zheng, and R.H. Deng. Security and privacy in smart health: Efficient policy-hiding attribute-based access control. *IEEE Internet of Things Journal*, 5(3):2130–2145, 2018.

[7] Y. Zhang, X. Chen, J. Li, D.S. Wong, H. Li, and I. You. Ensuring attribute privacy protection and fast decryption for outsourced data security in mobile cloud computing. *Information Sciences*, 379:42–61, 2017.

[8] A. Kapadia, P.P. Tsang, and S.W. Smith. Attribute-based publishing with hidden credentials and hidden policies. In *NDSS*, volume 7, pages 179–192, 2007.

[9] T. Nishide, K. Yoneyama, and K. Ohta. Attribute-based encryption with partially hidden encryptor-specified access structures. In *International conference on applied cryptography and network security*, pages 111–129. Springer, 2008.

[10] J. Li, K. Ren, B. Zhu, and Z. Wan. Privacy-aware attribute-based encryption with user accountability. In *International Conference on Information Security*, pages 347–362. Springer, 2009.

[11] J. Lai, R.H. Deng, and Y. Li. Fully secure cipertext-policy hiding cp-abe. In *International conference on information security practice and experience*, pages 24–39. Springer, 2011.

[12] J. Hao, C. Huang, J. Ni, H. Rong, M. Xian, and X.S. Shen. Fine-grained data access control with attribute-hiding policy for cloud-based iot. *Computer Networks*, 153:1–10, 2019.

[13] L. Zhang, G. Hu, Y. Mu, and F. Rezaeibagha. Hidden ciphertext policy attribute-based encryption with fast decryption for personal health record system. *IEEE Access*, 7:33202–33213, 2019.

[14] X. Liang, Z. Cao, H. Lin, and J. Shao. Attribute based proxy re-encryption with delegating capabilities. In *Proceedings of the 4th International Symposium on Information, Computer, and Communications Security*, pages 276–286, 2009.

[15] S. Luo, J. Hu, and Z. Chen. Ciphertext policy attribute-based proxy re-encryption. In *International Conference on Information and Communications Security*, pages 401–415. Springer, 2010.

[16] H.J. Seo and H.W. Kim. Attribute-based proxy re-encryption with a constant number of pairing operations. *Journal of information and communication convergence engineering*, 10(1):53–60, 2012.

[17] J. Li, S. Wang, Y. Li, H. Wang, H. Wang, H. Wang, J. Chen, and Z. You. An efficient attribute-based encryption scheme with policy update and file update in cloud computing. *IEEE Transactions on Industrial Informatics*, 15(12):6500–6509, 2019.

[18] Y. Zhang, J. Li, X. Chen, and H. Li. Anonymous attribute-based proxy re-encryption for access control in cloud computing. *Security and Communication Networks*, 9(14):2397–2411, 2016.

[19] S. Belguith, N. Kaaniche, and G. Russello. CUPS: Secure opportunistic cloud of things framework based on attribute-based encryption scheme supporting access policy update. *Security and Privacy*, 3(4):e85, 2020.

[20] Y. Jiang, W. Susilo, Y. Mu, and F. Guo. Ciphertext-policy attribute-based encryption supporting access policy update and its extension with preserved attributes. *International Journal of Information Security*, 17(5):533–548, 2018.

[21] 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*, 111:899–918, 2020.

[22] S. Hohenberger and B. Waters. Online/offline attribute-based encryption. In *International workshop on public key cryptography*, pages 293–310. Springer, 2014.

[23] P. Datta, R. Dutta, and S. Mukhopadhyay. Fully secure online/offline predicate and attribute-based encryption. In *International Conference on Information Security Practice and Experience*, pages 331–345. Springer, 2015.

[24] Y. Liu, Y. Zhang, J. Ling, and Z. Liu. Secure and fine-grained access control on e-healthcare records in mobile cloud computing. *Future Generation Computer Systems*, 78:1020–1026, 2018.

[25] J. Li, Y. Zhang, X. Chen, and Y. Xiang. Secure attribute-based data sharing for resource-limited users in cloud computing. *Computers & Security*, 72:1–12, 2018.

[26] K. Huang, X. Wang, and Z. Lin. Practical multi-authority attribute-based access control for edge-cloud-aided internet of things. *Security and Communication Networks*, 2021, 2021.

[27] M. La Manna, P. Perazzo, and G. Dini. Seabrew: A scalable attribute-based encryption revocable scheme for low-bitrate iot wireless networks. *Journal of Information Security and Applications*, 58:102692, 2021.

[28] M. Ali, J. Mohajeri, M.R. Sadeghi, and X. Liu. Attribute-based fine-grained access control for outscored private set intersection computation. *Information Sciences*, 536:222–243, 2020.

[29] M. Ali, J. Mohajeri, M.R. Sadeghi, and X. Liu. A fully distributed hierarchical attribute-based encryption scheme. *Theoretical Computer Science*, 815:25–46, 2020.

[30] Maryam Zarezadeh, Maede Ashouri Taluki, and Mohammad Siavashi. Attribute-based access control for cloud-based electronic health record (ehr) systems. *ISeCure*, 12(2), 2020.

[31] Aniseh Najafi, Majid Bayat, and Hamid Haj Seyyed Javadi. Privacy preserving attribute-based encryption with conjunctive keyword search for e-health records in cloud. *ISeCure*, 13(2), 2021.

[32] Sajjad Palanki and Alireza Shafieinejad. Attribute-based encryption with efficient attribute revocation, decryption outsourcing, and multi-keyword searching in cloud storage. *ISeCure*, 14(3), 2022.

[33] X. Yan, H. Ni, Y. Liu, and D. Han. Privacy-preserving multi-authority attribute-based encryption with dynamic policy updating in phr. *Computer Science and Information Systems*, 16(3):831–847, 2019.

[34] H. Xiong, Y. Zhao, L. Peng, H. Zhang, and K.H. Yeh. Partially policy-hidden attribute-based broadcast encryption with secure delegation in edge computing. *Future Generation Computer Systems*, 97:453–461, 2019.

[35] S. Even, O. Goldreich, and S. Micali. On-line/off-line digital signatures. *Journal of Cryptology*, 9(1):35–67, 1996.

[36] M. Blaze, G. Bleumer, and M. Strauss. Divertible protocols and atomic proxy cryptography. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 127–144. Springer, 1998.

[37] Q. Liu, G. Wang, and J. Wu. Time-based proxy re-encryption scheme for secure data sharing in a cloud environment. *Information sciences*, 258:355–370, 2014.

[38] Q. Liu, G. Wang, and J. Wu. Clock-based proxy re-encryption scheme in unreliable clouds. In *2012 41st International Conference on Parallel Processing Workshops*, pages 304–305. IEEE, 2012.

[39] H. Li, D. Liu, K. Alharbi, S. Zhang, and X. Lin. Enabling fine-grained access control with efficient attribute revocation and policy updating in smart grid. *TIIS*, 9(4):1404–1423, 2015.

[40] Q. Huang, L. Wang, and Y. Yang. Decent: Secure and fine-grained data access control with policy updating for constrained iot devices. *World Wide Web*, 21(1):151–167, 2018.

[41] K. Sethi, A. Pradhan, and P. Bera. Practical traceable multi-authority cp-abe with outsourcing decryption and access policy updation. *Journal of Information Security and Applications*, 51:102435, 2020.

[42] Jingwei Wang, Xinchun Yin, Jianting Ning, Shengmin Xu, Guowen Xu, and Xinyi Huang. Secure updatable storage access control system for ehrs in the cloud. *IEEE Transactions on Services Computing*, 2022.

[43] X. Yan, G. He, J. Yu, Y. Tang, and M. Zhao. Offline/online outsourced attribute-based encryption with partial policy hidden for the internet of things. *Journal of Sensors*, 2020, 2020.

[44] H. Tian, X. Li, H. Quan, C.C. Chang, and T. Baker. A lightweight attribute-based access control scheme for intelligent transportation system with full privacy protection. *IEEE Sensors Journal*, 2020.

[45] J. Sun, H. Xiong, X. Liu, Y. Zhang, X. Nie, and R.H. Deng. lightweight and privacy-aware fine-grained access control for iot-oriented smart health. *IEEE Internet of Things Journal*, 7(7):6566–6575, 2020.

[46] Chenbin Zhao, Li Xu, Jiguo Li, He Fang, and Yinghui Zhang. Toward secure and privacy-preserving cloud data sharing: Online/offline multiauthority cp-abe with hidden policy. *IEEE Systems Journal*, 16(3):4804–4815, 2022.

[47] Yinghui Zhang, Xuanni Wei, Jin Cao, Jianting Ning, Zuobin Ying, and Dong Zheng. Blockchain-enabled decentralized attribute-based access control with policy hiding for smart healthcare. *Journal of King Saud University-Computer and Information Sciences*, 34(10):8350–8361, 2022.

[48] Jing Cheng and Mi Wen. An efficient attribute encryption scheme with privacy-preserving policy in smart grid. *International Journal of Network Security*, 25(1):140–150, 2023.

[49] M. Chegenizadeh, M. Ali, J. Mohajeri, and M.R. Aref. An anonymous attribute-based access control system supporting access structure update.

In *2019 16th International ISC (Iranian Society of Cryptology) Conference on Information Security and Cryptology (ISCISC)*, pages 85–91. IEEE, 2019.

[50] Mahdi MahdaviOliaee and Zahra Ahmadian. Fine-grained flexible access control: ciphertext policy attribute based encryption for arithmetic circuits. *Journal of Computer Virology and Hacking Techniques*, pages 1–14, 2022.

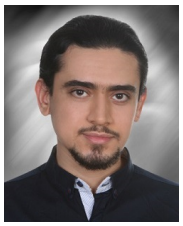
[51] Mahdi Mahdavi, Mohammad Hesam Tadayon, Mohammad Sayad Haghighi, and Zahra Ahmadian. Iot-friendly, pre-computed and outsourced attribute based encryption. *Future Generation Computer Systems*, 150:115–126, 2024.

[52] D. Boneh and M. Franklin. Identity-based encryption from the weil pairing. In *Annual international cryptology conference*, pages 213–229. Springer, 2001.

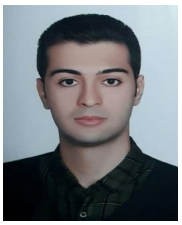
[53] D. Boneh, X. Boyen, and H. Shacham. Short group signatures. In *Annual international cryptology conference*, pages 41–55. Springer, 2004.

[54] O. Blazy, E. Conchon, M. Klingler, and D. Sauveron. An iot attribute-based security framework for topic-based publish/subscribe systems. *IEEE Access*, 9:19066–19077, 2021.

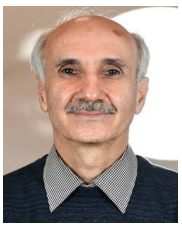
[55] L. Cheung and C. Newport. Provably secure ciphertext policy abe. In *Proceedings of the 14th ACM conference on Computer and communications security*, pages 456–465, 2007.



Mostafa Chegenizadeh received his B.Sc. and M.Sc. degrees in Electrical Engineering from the Sharif University of Technology, Tehran, Iran, in 2017 and 2019, respectively. With a solid grounding in cryptography, which serves as the fundamental building block for Blockchain Systems, his current research as a Ph.D. Candidate at the University of Zurich focuses on Blockchain Analytics, Cryptocurrencies, and Cryptographic Protocols.



Mohammad Ali received the M.Sc. degree in Applied Mathematics, in 2016, and the Ph.D. degree in mathematics from Amirkabir University of Technology, Tehran, Iran, in 2020. He is currently an assistant professor with the Department of Mathematics and Computer Science, Amirkabir University of Technology. His research interests include Provable Security Cryptography, Post-Quantum Cryptography, Cloud Security, and IoT Security.



Javad Mohajeri is an Assistant Professor with the Electronics Research Institute, Sharif University of Technology, Tehran, Iran, where he is an Adjunct Assistant Professor with the Electrical Engineering Department. His current research interests include Data Security and the Design and Analysis of Cryptographic Protocols and Algorithms. Javad is a Founding Member of the Iranian Society of Cryptology. Also, he has been the committee program chair of the second International ISC Conference on Information Security and Cryptology, and a committee program member of the third – 20th of this conference.

Mohammad Reza Aref received the B.Sc. degree from School of Electrical and Computer Engineering, University of Tehran, in 1975. The M.Sc. and Ph.D. degrees from Stanford University, Stanford, CA, USA, in 1976 and 1980, respectively. He Came back to Iran in 1980 and was actively engaged in academic and political affairs. He was a Faculty member of Isfahan University of Technology (1982-1997). He has been a professor of Electrical Engineering at Sharif University of Technology Since 1997.