Article

Ciphertext-Policy Attribute-Based Encryption with Outsourced Set Intersection in Multimedia Cloud Computing

Yanfeng Shi ¹ and Shuo Qiu ^{2,*} ¹ School of Computer Engineering, Nanjing Institute of Technology, Nanjing 211167, China; shiyf@njit.edu.cn² School of Software Engineering, Jinling Institute of Technology, Nanjing 211169, China

* Correspondence: shuoqiu@jit.edu.cn

Abstract: In a multimedia cloud computing system, suppose all cloud users outsource their own data sets to the cloud in the encrypted form. Each outsourced set is associated with an access structure such that a valid data user, Bob, with the credentials satisfying the access structure is able to conduct computing over outsourced encrypted set (e.g., decryption or other kinds of computing function). Suppose Bob needs to compute the set intersection over a data owner Alice's and his own outsourced encrypted sets. Bob's simple solution is to download Alice's and Bob's outsourced encrypted sets, perform set intersection operation, and decrypt the set intersection ciphertexts. A better solution is for Bob to delegate the cloud to calculate the set intersection, without giving the cloud any ability in breaching the secrecy of the sets. To solve this problem, this work introduces a novel primitive called *ciphertext-policy attribute-based encryption with outsourced set intersection* for multimedia cloud computing. It is the first cryptographic algorithm supporting a fully outsourced encrypted storage, computation delegation, fine-grained authorization security for ciphertext-policy model, without relying on an online trusted authority or data owners, and multi-elements set, simultaneously. We construct a scheme that provably satisfies the desirable security properties, and analyze its efficiency.

Citation: Shi, Y.; Qiu, S.Ciphertext-Policy Attribute-Based Encryption with Outsourced Set Intersection in Multimedia Cloud Computing. *Electronics* **2021**, *10*, 2685. <https://doi.org/10.3390/electronics10212685>

Academic Editor: Martin Reisslein

Received: 2 October 2021

Accepted: 30 October 2021

Published: 3 November 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: PSI; CP-ABE; outsourced computing; multimedia cloud computing

1. Introduction

The multimedia cloud computing has been a successful computing paradigm since it can offer mass data storage and elastic but powerful computing capabilities at an affordable price. Cloud users, e.g., data owners and data users, can outsource their multimedia data to the cloud, then access it and share it with other users at anytime from anywhere. However, since data outsourcing will undermine the control capability for data owners since the outsourced data resides in the cloud service providers' scope, it inevitably raises the security concerns regarding the data confidentiality, data integrity, and data access control. While it is natural to encrypt data before outsourcing it to the cloud, it indeed raises another issue about how to facilitate the cloud to perform some specific function over outsourced encrypted data without compromising data privacy.

In this paper, we study the problem of set intersection over outsourced encrypted data sets, which is a variant of the private set intersection (PSI) problem [1] that has been extensively studied. In the traditional setting of PSI, two data users, holding their own data sets at local, joint compute the data sets' intersection in a manner that they only obtain the information about the intersection but nothing else. PSI has been widely adopted to many practical applications, such as private data mining, private matching, credit information system, anti-epidemic information system, and educational resources sharing, etc.

Problem. For the multimedia cloud computing scenario, the cloud users (i.e., the data owners and the data users) no longer store their own data sets at local and calculate the set intersection by themselves. Instead, the cloud users expect to outsource them to the cloud in encrypted form and entrust the cloud to help them complete the set intersection

operation. While several works have studied PSI over outsourced encrypted data in the cloud [2–4], it certainly raises a difficulty for the data owners on how to enforce that their encrypted data can be used to perform by only some specific users. That is, how can the data owners limit the ability for the data users on performing set intersection by complying with certain control policies.

Motivation. For this problem, Shi et al. combined key-policy attribute-based encryption and PSI to propose KP-ABSI to support fine-grained authorization for PSI computation [5]. In their protocol, it restricts PSI operation permissions through access control policy, embedded in data users' secret keys. However, in KP-ABSI, the access structure is associated with data users instead of outsourced data sets. So, it is not particularly suitable for achieving access control over cloud data. Recently, Ali et al. proposed a ciphertext-policy attribute-based set intersection solution [6] to realize fine-grained authorization for outsourced data sets of data owners. Unfortunately, in their protocol, the data sets of the data users must be in plaintext form and the data users should execute PSI operations at local. It is not applicable to the case where all cloud users outsourced their encrypted data sets to the cloud and delegate PSI computation to the cloud. To our knowledge (cf. Section 2), none of the existing schemes consider this problem how to achieve the following objectives simultaneously: (1) All cloud users should outsource their data sets in the ciphertext form to the cloud. (2) The cloud users should be able to preserve data privacy while allowing the set intersection delegation to the cloud. (3) The cloud users should be able to accomplish fine-grained access control to the set operation permissions over their outsourced encrypted data sets. (4) It should not rely on the always online trusted authority and the data owners.

Our Contributions. For solving the above problem, this work introduces a novel notion called *ciphertext-policy attribute-based encryption with outsourced set intersection* (CP-ABSI) for multimedia cloud computing. In CP-ABSI, all cloud users encrypt their own data sets by customizing an access structure before outsourcing. An authorized data user (Bob), whose attributes satisfy the access structure of the data owner's (Alice) data set, can delegate the set intersection calculation between Alice's and Bob's data sets to the cloud by generating a token. This work presents the formal definition of CP-ABSI with security properties, then presents a specific scheme. There are five characteristic properties for our CP-ABSI scheme: (1) It supports fully outsourced encrypted data sets for all cloud users, including data owners and data users. (2) The cloud can be delegated to execute PSI operations and not acquire any additional information beyond the result of encrypted set intersection. (3) It combines PSI and CP-ABE to realize fine-grained authorization for outsourced set intersection operation. (4) The data owner and the trusted authority are not required to participate in the interaction. (5) It supports multi-elements data sets. It is suitable for multimedia cloud computing.

2. Related Work

Although PSI has been widely studied previously, all existing solutions did not resolve the problems considered in this work. We brief the relevant technologies below and compare them with our CP-ABSI as shown in Table 1.

Two-party Private Set Intersection. In this scenario, it only involves two participants, a client and a server, both of whom work interactively to accomplish the set intersection operation with their own data sets [7,8]. Unfortunately, these solutions cannot be adapted to cloud architecture; for that, both parties have to decrypt all of their outsourced encrypted data sets and do the heavy computations for PSI at local. Obviously, they cannot support fine-grained authorization and the data owner should be always online.

Table 1. Property summary for PSI schemes. “×” means not supported, “√” means supported, “CP” means ciphertext-policy and “KP” means key-policy. In the related work and our solution in this paper. Fully outsourced encrypted storage means that all cloud users, including data owners and data users, can outsource their data sets in encrypted form. Computation delegation means it supports delegating PSI operations to the cloud. Fine-grained authorization security (as defined in Definition 4) means that it supports fine-grained authorization security with key-policy or ciphertext-policy setting. Without always online trusted authority or data owner means that it is not required that the trusted authority or data owners are always online. Multi-elements set means it supports multiple elements data set.

Schemes	Fully Outsourced Encrypted Storage	Computation Delegation	Fine-Grained Authorization Security	Without Always Online Trusted Authority or Data Owner	Multi-Elements Set
Two-party Private Set Intersection [7,8]	×	×	×	×	√
Three-party Private Set Intersection [2–4,9]	√	√	×	×	√
Symmetric/Asymmetric Encryption with Equality Test [10,11]	√	√	×	√	×
KP-ABEET [12]	√	√	×	×	×
CP-ABEET [13,14]	√	√	×	×	×
AB-PSI [6]	×	×	CP	√	√
KP-ABSI [5]	√	√	KP	√	√
Our CP-ABSI	√	√	CP	√	√

Three-party Private Set Intersection. In this model, it includes three participants, i.e., the cloud servers, the data owners, and data users, where the data owners and the data users wish to outsource set intersection operations to the server. For example, refs. [2–4,9] considered outsourcing set intersection operations to the cloud while preserving the privacy of the data sets. In addition, symmetric/asymmetric encryption with equality test, e.g., [10,11], is a specific solution for one-element data sets. Unfortunately, all of these solutions ask for the data owner to always be online or have no authorization mechanism. Thus, traditional three-party PSI protocols are impractical for the cloud system.

Attribute-based Encryption (ABE). A reliable solution for realizing fine-grained authorization for outsourced data is ABE, introduced by Sahai and Waters [15]. This technique allows the data users with appropriate attributes to decrypt the ciphertexts under an access structure. There are two types of ABE: ciphertext-policy ABE (CP-ABE) in which the ciphertext is generated under an access structure (e.g., [16,17]) and key-policy ABE (KP-ABE) in which the private key is generated under an access structure (e.g., [18,19]). ABE also can be combined with other techniques to provide fine-grained operation and computation, e.g., attribute-based keyword search (e.g., [20]) or attribute-based proxy re-encryption (e.g., [21]).

Attribute-based Set Intersection. The first attribute-based set intersection solution is attribute-based encryption with equality test (ABEET). ABEET supports access control and set intersection computation delegation, but only for one-element data sets. Zhu et al. and Wang et al., respectively, present the first KP-ABEET and CP-ABEET [12,13]. After that, Cui et al. presented a performance enhanced ciphertext-policy construction [14]. Unfortunately, in these schemes, the trapdoor is generated with the master secret key held by the trusted authority and the data owner is required to participate in the delegating phase to send its token to the cloud. So, the trusted authority and the data owners should be always online. Even worse, in existing ABEET schemes, the cloud server without $\text{token}_{A,C}$ can obtain set intersection $A \cap C$ with $\text{token}_{A,B}$ for $A \cap B$ and $\text{token}_{B,C}$ for $B \cap C$. So, they actually cannot achieve fine-grained authorization security, which will be illustrated in Definition 4. It is impractical and secure enough for distributed cloud computing. The first attribute-

based solution for multi-elements set intersection is proposed by Mohammad Ali et. al. It achieves fine-grained authorization security and is not dependent on an online trusted authority or data owners. However, in Ali's scheme, data users must store the plaintext form of their set and execute PSI operations at local. It is not supporting fully outsourced encrypted storage and computation delegation, so it is not applied to our scenario where the data sets of all cloud users are encrypted in the cloud. Later, Shi et al. introduced a KP-ABSI scheme by combining KP-ABE and PSI to achieve fully outsourced encrypted storage and computation delegation. However, it is still not applied to our scenario since it implements access control for cloud users instead of cloud data sets [5].

For that, in this work, we combine PSI and CP-ABE to propose a novel primitive named ciphertext-policy attribute-based encryption with outsourced set intersection (CP-ABSI), implementing access control for ciphertext data sets. Compared with existing solutions, our CP-ABSI can support a fully outsourced encrypted storage, computation delegation, fine-grained authorization security for ciphertext-policy(CP) model, without relying on an online trusted authority or data owners, and multi-elements set, simultaneously.

3. Problem Formulation

3.1. System Model

Figure 1 highlights the system model, which consists of three entities—a cloud server, a trusted attribute authority, and the cloud users, e.g., a data owner (named Alice), an authorized data user (named Bob), and an unauthorized data user (named Calos). The cloud affords for the cloud users computing and storing supporting. The trusted attribute authority issues data users' secret keys in line with their attribute certificates. The data owner (i.e., Alice), which can be either individuals or organizations, encrypt their private data sets according to an access control policy and then outsource them to the cloud server. An authorized user (i.e., Bob), whose attributes satisfy Alice's and his own access structures simultaneously, can delegate set intersection operations over Alice's and his own encrypted data sets to the cloud. Whereas any unauthorized user, i.e., Calos, whose attributes do not satisfy the access structure of Alice, cannot outsource the set intersection operations to the cloud.

Figure 1. System model of CP-ABSI.

Here, we suppose that the cloud is honest but curious, meaning that it runs the protocols honestly but attempts to obtain privacy information about the data sets.

A cloud user is malicious and might collude with the semi-trusted cloud.

3.2. Formal Definition

Here is shown the formal definition of CP-ABSI (ciphertext-policy attribute-based encryption with outsourced set intersection), in which ciphertexts are generated under an access structure. T is denoted as an access policy and UA is denoted as an attribute set, respectively, in CP-ABSI. Let $F(T, UA) = 1$ if and only if T is satisfied by UA in CP-ABSI.

Definition 1. (CP-ABSI) A scheme for CP-ABSI has the following algorithms:

$(pk, msk) \leftarrow \text{Setup}(1^\ell)$: The trusted attribute authority bootstraps the master private key msk and the public parameters pk as input a security parameter ℓ .

$sk \leftarrow \text{KeyGen}(msk, UA)$: The trusted attribute authority generates secret keys sk for a data user.

$c \leftarrow \text{Enc}(D, T)$: Given T , a cloud user (data owner or data user) encrypts their private data set D . Then they outsource the ciphertexts to the cloud.

$\text{token} \leftarrow \text{TokenGen}(sk)$: A data user runs this algorithm to generate a token to delegate the set intersection operations between the data owner's and his own encrypted data sets to the cloud, where the token token will be sent to the cloud for this purpose.

$\text{rslt} \leftarrow \text{SI}(\text{token}, c, c')$: The cloud server runs this algorithm to calculate the set intersection rslt , only if UA associated to token meets both $F(T, UA) = 1$ and $F(T', UA) = 1$, where T and T' are specified by c of the data owner and c' of the data user, respectively.

The correctness of CP-ABSI scheme can be defined as: With $(pk, msk) \leftarrow \text{Setup}(1^\ell)$, $sk \leftarrow \text{KeyGen}(msk, UA)$, $\text{token} \leftarrow \text{TokenGen}(sk)$, $c \leftarrow \text{Enc}(D, T)$ for set D and $c' \leftarrow \text{Enc}(D', T')$ for set D' , if $F(T, UA) = 1$ and $F(T', UA) = 1$, rslt is the ciphertext of set intersection $D \cap D'$, where $\text{rslt} \leftarrow \text{SI}(c, c', \text{token})$.

3.3. Security Definitions

It is said that the CP-ABSI scheme is secure if it satisfies the following security properties: selective security against chosen-plaintext attack, one-wayness under a chosen-plaintext attack, and fine-grained authorization security.

Selective security against chosen-plaintext attack: Intuitively, this property says that a probabilistic polynomial-time (PPT) adversary $\mathcal{A}$, modeling malicious unauthorized data users, cannot obtain any useful information about the encrypted sets without being given the corresponding tokens, where the term “selective” means that it asks for the adversary $\mathcal{A}$ to choose T which parties to compromise before initializing the public parameters. This security property can be formally defined by the game between $\mathcal{A}$ and a challenger as follows.

Setup: $\mathcal{A}$ chooses T^* and sends it to the challenger. Then the challenger executes Setup to bootstrap msk and pk , it sets msk as the master private key and sends pk to $\mathcal{A}$ as a public key.

Phase 1: $\mathcal{A}$ can make queries for polynomial-many times as follows:

- $\mathcal{O}_{\text{KeyGen}}(UA)$: If $F(T^*, UA) = 1$, the challenger aborts; else, it executes $sk \leftarrow \text{KeyGen}(msk, pk, UA)$ and sends sk back to $\mathcal{A}$.
- $\mathcal{O}_{\text{TokenGen}}(UA)$: If $F(T^*, UA) = 1$, then the challenger aborts the simulation; otherwise, it executes $sk \leftarrow \text{KeyGen}(msk, UA)$, $\text{token} \leftarrow \text{TokenGen}(sk)$ and sends token to $\mathcal{A}$.

Challenge: $\mathcal{A}$ selects two data sets D_0 and D_1 where $|D_0| = |D_1|$ but $D_0 \neq D_1$, and sends them to the challenger. The challenger picks $\sigma \xleftarrow{R} \{0, 1\}$, executes $c^* = \text{Enc}(D_\sigma, T^*)$ and returns c^* back to $\mathcal{A}$.

Phase 2: $\mathcal{A}$ executes as in Phase 1.

Guess: A guess σ' is replied by $\mathcal{A}$. $\mathcal{A}$ wins the security game if $\sigma = \sigma'$.

Definition 2. If the advantage $|\Pr[\sigma' = \sigma] - 1/2|$ that $\mathcal{A}$ wins the selective security game is negligible, the CP-ABSI construction is selective secure.

One-wayness security: Intuitively, this property requires that any PPT adversary $\mathcal{A}$, modeling honest-but-curious cloud server, even with a token, is unable to acquire any useful information about the plaintexts corresponding to the ciphertexts in question. Note that, given a target ciphertext and the appropriate token, where the term “appropriate” means the credentials that generate the token satisfy the access control policy associated to the target ciphertext, $\mathcal{A}$ can purposefully choose and encrypt a data set element with public parameters, and compare their chosen ciphertext with the target ciphertext. Obviously, brute-force attack is inevitable, so it is just demanded that $\mathcal{A}$ have no better strategy. The one-wayness security can be defined by the following game.

Setup: The challenger produces msk, pk by executing Setup, sets msk as the master secret key, and sends pk to $\mathcal{A}$ as the public parameters.

Phase 1: $\mathcal{A}$ can query following oracles for polynomial-many times, the challenger book-keeps a list L_{UA} initially empty.

- $\mathcal{O}_{\text{KeyGen}}(\text{UA})$: The challenger replies $\text{sk} \leftarrow \text{KeyGen}(\text{msk}, \text{UA})$ to $\mathcal{A}$ and inserts UA to L_{UA} .
- $\mathcal{O}_{\text{TokenGen}}(\text{UA})$: The challenger replies $\text{token} \leftarrow \text{TokenGen}(\text{sk})$ (where $\text{sk} \leftarrow \text{KeyGen}(\text{msk}, \text{UA})$) to $\mathcal{A}$.

Challenge: $\mathcal{A}$ selects and sends T^* to the challenger, where $\forall \text{UA} \in L_{\text{UA}}, F(T^*, \text{UA}) = 0$. The challenger chooses UA^* such that $F(T^*, \text{UA}^*) = 1$, selects D^* uniformly at random, calculates $c^* \leftarrow \text{Enc}(D^*, T^*)$ and $\text{token}^* \leftarrow \text{TokenGen}(\text{KeyGen}(\text{UA}^*))$. Then return c^*, token^* back to $\mathcal{A}$.

Phase 2: It is similar to Phase 1, except that $F(T^*, \text{UA}) = 0$ when querying $\mathcal{O}_{\text{KeyGen}}(\text{UA})$.

Guess: A guess d is replied by $\mathcal{A}$. $\mathcal{A}$ wins the one-wayness security game if $d \in D^*$.

Definition 3. If the advantage $|\Pr[d \in D^*] - \frac{m|D^*|}{|\text{Msg}|}|$, in which Msg is the domain size of set elements and m is the number of guess/brute-force attacks $\mathcal{A}$ makes, that $\mathcal{A}$ wins the one-wayness security game is negligible, the CP-ABSI construction is one-wayness secure.

Fine-grained authorization: The adversary $\mathcal{A}$, modeling the honest-but-curious cloud server, cannot use the known tokens to obtain set intersection if the credentials that generate the tokens do not satisfy the access control policies associated to two ciphertexts simultaneously. Intuitively, $\mathcal{A}$ cannot use $\text{token}_{A,B}$ for $A \cap B$ and $\text{token}_{B,C}$ for $B \cap C$ to achieve set intersection $A \cap C$, if $\text{token}_{A,B}$ and $\text{token}_{B,C}$ whose associated attribute sets do not satisfy A and C 's access structure simultaneously. This can be formalized via the following fine-grained authorization security game.

Setup: The challenger produces msk, pk by executing Setup, sets msk as the master private key, and returns pk back to $\mathcal{A}$.

Phase 1: $\mathcal{A}$ can make polynomial-many oracle queries as follows, and the challenger bookkeeps two lists L_{UA} and L_{token} initially empty.

- $\mathcal{O}_{\text{KeyGen}}(\text{UA})$: The challenger replies $\text{sk} \leftarrow \text{KeyGen}(\text{msk}, \text{UA})$ back to $\mathcal{A}$ and adds UA to L_{UA} .
- $\mathcal{O}_{\text{TokenGen}}(\text{UA})$: The challenger replies $\text{token} \leftarrow \text{TokenGen}(\text{sk})$ (where $\text{sk} \leftarrow \text{KeyGen}(\text{msk}, \text{UA})$) to $\mathcal{A}$ and insert UA into L_{token} .

Challenge: $\mathcal{A}$ selects T_1^* and T_2^* , and sends them to the challenger. The challenger chooses two sets D_0, D_1 , selects $\sigma \xleftarrow{R} \{0, 1\}$, replies $c_1^* \leftarrow \text{Enc}(D_0, T_1^*)$ and $c_2^* \leftarrow \text{Enc}(D_{\sigma}, T_2^*)$ back to $\mathcal{A}$. It is required that

- $\forall \text{UA} \in L_{\text{UA}}, F(T_1^*, \text{UA})$ and $F(T_2^*, \text{UA})$ cannot be 1 simultaneously.

- $\forall \text{UA} \in L_{\text{token}}, F(T_1^*, \text{UA})$ and $F(T_2^*, \text{UA})$ cannot be 1 simultaneously.

Phase 2: It is similar to Phase 1, except that:

- For any query for $\mathcal{O}_{\text{KeyGen}}(\text{UA})$ or $\mathcal{O}_{\text{TokenGen}}(\text{UA})$, $F(T_1^*, \text{UA})$ and $F(T_2^*, \text{UA})$ cannot be 1 simultaneously.

Guess: A guess σ' is replied by $\mathcal{A}$. $\mathcal{A}$ wins the fine-grained authorization game if $\sigma = \sigma'$.

Definition 4. If the advantage $|\Pr[\sigma' = \sigma] - \frac{1}{2}|$ that $\mathcal{A}$ wins fine-grained authorization security game is negligible, the CP-ABSI construction is fine-grained authorization secure.

4. Scheme Construction

4.1. CP-ABSI Construction

The CP-ABSI scheme is described as follows.

Setup(1^ℓ): Given security parameter ℓ , it generates the public parameter and master key as follows:

- Set $(e, q, g, g_T, G, G_T) \leftarrow \text{BMapGen}(1^\ell)$.
- Let $H_1 : \{0, 1\}^* \rightarrow G$ and $H_2 : \{0, 1\}^* \rightarrow G$ be two secure hash functions.
- Choose $a, b \xleftarrow{R} \mathbb{Z}_p$, then make the public parameter as $\text{pk} = (e, G, G_T, g, g^a, g^b, H_1, H_2)$ and the master secret key as $\text{msk} = (a, b)$.

KeyGen($\text{msk}, \text{pk}, \text{UA}$): It picks $t \xleftarrow{R} \mathbb{Z}_p$, then calculates $X_1 = g^{at}$ and $X_2 = g^{bt}$. For each attribute $\text{at}_i \in \text{UA}$, it selects $c_i \xleftarrow{R} \mathbb{Z}_p$, and computes $Y_i = g^{abt} H_1(\text{at}_i)^{c_i}$ and $Z_i = g^{c_i}$. It sets the private key as $\text{sk} = (\text{UA}, X_1, X_2, \{(Y_i, Z_i) | \text{at}_i \in \text{UA}\})$.

Enc(D, T, pk): Given set $D = \{d_0, \dots, d_n\}$ where $d_j \in \{0, 1\}^*$, it encrypts D as follows: It picks $r_1, r_2 \xleftarrow{R} \mathbb{Z}_p$ and computes $A_1 = g^{br_1}$, $A_2 = g^{a(r_1+r_2)} H_2(d_j)$ for each d_j . It runs $\{q_v(0) | v \in \text{leaf}(T)\} \leftarrow \text{Share}(T, r_2)$ and computes $B_v = g^{q_v(0)}$, $C_v = H_1(\text{att}(v))^{q_v(0)}$ for each $v \in \text{leaf}(T)$. It sets the ciphertext as $c = \{c_0, \dots, c_n\}$, where $c_j = (T, A_1, A_2, \{B_v, C_v | v \in \text{leaf}(T)\})$.

TokenGen(sk, pk): It selects $k \xleftarrow{R} \mathbb{Z}_p$, computes $\hat{Y}_i = Y_i^k$, $\hat{Z}_i = Z_i^k$ for each $\text{at}_i \in \text{UA}$, and sets $\hat{X}_1 = X_1^k$, $\hat{X}_2 = X_2^k$. It sets the token as $\text{token} = (\text{UA}, \hat{X}_1, \hat{X}_2, \{\hat{Y}_i, \hat{Z}_i | \text{at}_i \in \text{UA}\})$.

SI(token, c, c'): As input $c = \{c_0, \dots, c_n\}$, $c' = \{c'_0, \dots, c'_m\}$ and a token token , it executes as follows:

- For $c_j \in c$, it arbitrarily chooses a subset $S \in \text{UA}$ satisfying T . If there is no such S , 0 is returned. Else, it calculates $E_v = e(\hat{Y}_i, B_v) / e(\hat{Z}_i, C_v) = e(g, g)^{abtkq_v(0)}$, where $\text{att}(v) = \text{at}_i$. Then it runs $E_{\text{root}} = e(g, g)^{abtkr_2} \leftarrow \text{Combine}(T, E_v | \text{att}(v) \in S)$, and computes $E_1 = e(A_1, \hat{X}_1) = e(g, g)^{abtkr_1}$ and $E_{2j} = e(A_2, \hat{X}_2) / (E_{\text{root}} E_1) = e(H_2(d_j), g)^{bkt}$. It sets $E = \{E_{20}, \dots, E_{2n}\}$.
- For $c'_j \in c'$, it arbitrarily chooses a subset $S' \in \text{UA}$ satisfying T' . If there is no such S' , 0 is returned. Else, it calculates $E'_v = e(\hat{Y}_i, B'_v) / e(\hat{Z}_i, C'_v) = e(g, g)^{abtkq'_v(0)}$, where $\text{att}(v) = \text{at}_i$. Then it runs $E'_{\text{root}} = e(g, g)^{abtkr'_2} \leftarrow \text{Combine}(T', E'_v | \text{att}(v) \in S')$, and computes $E'_1 = e(A'_1, \hat{X}_1) = e(g, g)^{abtkr'_1}$ and $E'_{2j} = e(A'_2, \hat{X}_2) / (E'_{\text{root}} E'_1) = e(H_2(d'_j), g)^{bkt}$. It sets $E' = \{E'_{20}, \dots, E'_{2m}\}$.
- It outputs the set intersection $\text{rslt} = \{c_j | c_j \in c, E_{2j} \in E \cap E'\}$.

we can verify the correctness for CP-ABSI construction easily according to the protocols. Then its security properties will be analyzed.

4.2. Security Analysis

Theorem 1. The CP-ABSI scheme is selective secure against chosen-plaintext attack in the generic bilinear group model as specified in Definition 2.

Proof. It is first to show that the scheme is secure when $|D| = 1$, and then extend it to the case $|D| > 1$.

Since in the challenge phase of the game, $\mathcal{A}$'s aim is to distinguish $g^{a(r_1+r_2)}H_2(d_0)$ and $g^{a(r_1+r_2)}H_2(d_1)$, where $D_0 = \{d_0\}$ and $D_1 = \{d_1\}$. The probabilities of $\mathcal{A}$ distinguishing $g^{a(r_1+r_2)}H_2(d_0)$ from g^θ and distinguishing g^θ from $g^{a(r_1+r_2)}H_2(d_1)$ where $\theta \xleftarrow{R} \mathbb{Z}_p$ are the same. Therefore, if $\mathcal{A}$ distinguishes $g^{a(r_1+r_2)}H_2(d_0)$ from $g^{a(r_1+r_2)}H_2(d_1)$ with probability μ , then $\mathcal{A}$ distinguishes between $g^{a(r_1+r_2)}H_2(d_0)$ and g^θ with probability $\mu/2$. We consider the following adjusted game, in which $\mathcal{A}$ attempts to discriminate between $g^{a(r_1+r_2)}$ and g^θ with $\theta \in \mathbb{Z}_p$.

Setup: The challenger randomly picks $a, b \xleftarrow{R} \mathbb{Z}_p$, sets public parameter as $(G, G_T, e, p, g, g^a, g^b)$, and transfers them to $\mathcal{A}$. It bookkeeps two initially empty lists $List_{H_1}(at_j, \alpha_j)$ and $List_{H_2}(d, \gamma)$. $\mathcal{A}$ can query polynomially many times for $\mathcal{O}_{H_1}$ and $\mathcal{O}_{H_2}$.

$\mathcal{O}_{H_1}(at_j)$: If at_j has not been queried before, it picks $\alpha_j \xleftarrow{R} \mathbb{Z}_p$, replies g^{α_j} to $\mathcal{A}$ and inserts (at_j, α_j) into $List_{H_1}$; Else, it retrieves α_j from $List_{H_1}$ and replies g^{α_j} to $\mathcal{A}$.

$\mathcal{O}_{H_2}(d)$: If an element d was queried before, it searches γ from $List_{H_2}$ and sends g^γ back to $\mathcal{A}$; else, it picks $\gamma \xleftarrow{R} \mathbb{Z}_p$, inserts (d, γ) into $List_{H_2}$, then sends g^γ back to $\mathcal{A}$.

Phase 1: $\mathcal{A}$ can query the oracles polynomially many times as follows.

$\mathcal{O}_{KeyGen}(UA)$: The challenger selects $t^{(u)} \xleftarrow{R} \mathbb{Z}_p$. For each $at_j \in UA$, it chooses $c_j^{(u)} \xleftarrow{R} \mathbb{Z}_p$, and sets $sk = (UA, X_1 = g^{at^{(u)}}, X_2 = g^{bt^{(u)}} \{Y_j = g^{abt^{(u)} + \alpha_j c_j^{(u)}}, Z_j = g^{c_j^{(u)}} | at_j \in UA\})$.

$\mathcal{O}_{token}(UA)$: The challenger chooses $k^{(u)} \xleftarrow{R} \mathbb{Z}_p$ and calculates $token = (UA, \hat{X}_1 = X_1^{k^{(u)}} = g^{at^{(u)}k^{(u)}}, \hat{X}_2 = X_2^{k^{(u)}} = g^{bt^{(u)}k^{(u)}}, \{\hat{Y}_j = Y_j^{k^{(u)}} = g^{abt^{(u)}k^{(u)} + \alpha_j c_j^{(u)}k^{(u)}}, \hat{Z}_j = Z_j^{k^{(u)}} = g^{c_j^{(u)}k^{(u)}} | at_j \in UA\})$.

Challenge: $\mathcal{A}$ chooses and transfers an access policy T^* to the challenger, which is not satisfied by any UA that was queried to $\mathcal{O}_{KeyGen}$. The challenger picks two equal length messages (d_0, d_1) uniformly at random from the message space, selects $r_1, r_2 \xleftarrow{R} \mathbb{Z}_p$ and $\sigma \xleftarrow{R} \{0, 1\}$, and runs $\{q_v(0) | v \in \text{leaf}(T^*)\} \leftarrow \text{Share}(T^*, r_2)$. If $\sigma = 0$, then the challenger computes $c^* = (A_1 = g^{br_1}, A_2 = g^\theta, \{B_v = g^{q_v(0)}, C_v = g^{\alpha_j q_v(0)} | v \in \text{leaf}(T^*)\})$, where $\text{att}(v) = at_j$. Otherwise, it calculates $c^* = (A_1 = g^{br_1}, A_2 = g^{a(r_1+r_2)}, \{B_v = g^{q_v(0)}, C_v = g^{\alpha_j q_v(0)} | v \in \text{leaf}(T^*)\})$, where $\text{att}(v) = at_j$. At last, it returns c^* to $\mathcal{A}$.

Phase 2: $\mathcal{A}$ queries the oracles as in Phase 1.

Guess: $\mathcal{A}$ outputs a guess σ' of σ .

In the generic model, if it is distinguishable between $e(g, g)^{a(r_1+r_2)}$ and g^θ for $\mathcal{A}$, it should build $e(g, g)^{\Gamma a(r_1+r_2)}$ with certain g^Γ . Then it will be shown that the probability that $\mathcal{A}$ build $e(g, g)^{\Gamma a(r_1+r_2)}$ for certain g^Γ is negligible, possibly taking advantage of the oracles' outputs. Table 2 lists all the possible queries to group oracle G_T .

Let us discuss the way building $e(g, g)^{\Gamma a(r_1+r_2)}$ with certain Γ . Since only the term br_1 has the element r_1 , the only way of building $e(g, g)^{\Gamma a(r_1+r_2)}$ for $\mathcal{A}$ is building $e(g, g)^{\Gamma'abr_2}$, where $\Gamma = \Gamma'b$. Obviously, to construct $\Gamma'abr_2$, $\mathcal{A}$ needs to use $q_v(0)$, $\alpha_j q_v(0)$, $c_j^{(u)}$, $c_j^{(u)}k^{(u)}$, $abt^{(u)} + \alpha_j c_j^{(u)}$ and $abt^{(u)}k^{(u)} + \alpha_j c_j^{(u)}k^{(u)}$. However, only under the condition that the attribute set satisfies T^* can r_2 be reconstructed.

Thus far, it is shown that $\mathcal{A}$ can gain a negligible advantage in the modified game.

Intuitively, for the general case of $|\mathcal{D}| > 1$, we can also prove that the scheme is selective secure against chosen-plaintext attack as in [5]. $\square$

Table 2. Possible group oracle G_T queries in the selective security game.

Queries 1	Queries 2	Queries 3	Queries 4
a	$c_j^{(u)}$	$c_j^{(u)}k^{(u)}$	$q_v(0)$
b	$at^{(u)}$	$at^{(u)}k^{(u)}$	$\alpha_j q_v(0)$
α_j	$bt^{(u)}$	$bt^{(u)}k^{(u)}$	
br_1	$abt^{(u)} + \alpha_j c_j^{(u)}$	$abt^{(u)}k^{(u)} + \alpha_j c_j^{(u)}k^{(u)}$	

Theorem 2. The CP-ABSI scheme is one-wayness secure if hash function H_2 is one-way.

Proof. Suppose there is a PPT adversary $\mathcal{A}$ winning the game for one-wayness with a non-negligible advantage μ , a challenger can be simulated to break one-wayness for H_2 .

Given $H_2(d^*) = y^*$, the challenger executes the following game.

Setup: It chooses $a, b \xleftarrow{R} \mathbb{Z}_p$, sets $\text{msk} = (a, b)$, and transfers $\text{pk} = (G, G_T, e, p, g, g^a, g^b)$ to $\mathcal{A}$.

Phase 1: It keeps an initially empty list L_{UA} . $\mathcal{A}$ is able to query the following oracles in polynomially many times:

$\mathcal{O}_{\text{KeyGen}}(\text{UA})$: For an attribute set UA, it returns $\text{sk} \leftarrow \text{KeyGen}(\text{msk}, \text{pk}, \text{UA})$ to $\mathcal{A}$ and inserts UA into L_{UA} .

$\mathcal{O}_{\text{token}}(\text{UA})$: For an attribute set UA, it replies $\text{token} \leftarrow \text{TokenGen}(\text{KeyGen}(\text{msk}, \text{pk}, \text{UA}), \text{pk})$.

Challenge: $\mathcal{A}$ transfers to the challenger an access control policy T^* , where $\forall \text{UA} \in L_{\text{UA}}, F(\text{UA}, T^*) = 0$. The challenger selects $j \xleftarrow{R} \mathbb{Z}_p$ and generates a data set $D^* = (d_0, d_1, \dots, d_{|D^*|-1})$, where $d_0, \dots, d_{j-1}, d_{j+1}, \dots, d_{|D^*|-1}$ are chosen from Msg at random and d_j is set as $d_j = d^*$ implicitly. It sets the ciphertext $c = \{c_k\}_{k \in [0, |D^*|-1]}$ as follows:

- If $k \neq j$, c_k is the same with that in the real scheme;
- If $k = j$, set $A_1 = g^{br_1}, A_2 = g^{a(r_1+r_2)}y^*$, by choosing $r_1, r_2 \xleftarrow{R} \mathbb{Z}_p$ at random and setting $d_j = d^*$ implicitly. Then run $\{q_v(0) | v \in \text{leaf}(T^*)\} \leftarrow \text{Share}(T^*, r_2)$ and compute $B_v = g^{q_v(0)}, C_v = H_1(\text{att}(v))^{q_v(0)}$ for each $v \in \text{leaf}(T^*)$. At last it sets $c_k = (T^*, A_1, A_2, \{B_v, C_v | v \in \text{leaf}(T^*)\})$

The challenger chooses an attribute set UA^* such that $F(\text{UA}^*, T^*) = 1$, and forwards to the adversary $\mathcal{A}$ $\text{token}^* \leftarrow \text{TokenGen}(\text{KeyGen}(\text{msk}, \text{pk}, \text{UA}^*), \text{pk})$ and $c^* = (c)$.

Phase 2: It is the same as that defined by the game.

Guess: The challenger wins if $\mathcal{A}$ replies a guess d satisfying $d = d^*$.

It is shown that if $\mathcal{A}$ outputs $d \in D^*$ with probability $|\Pr[d \in D^*] - \frac{m|D^*|}{|\text{Msg}|}| = \mu$, then $\Pr[d \in D^*] = \frac{m|D^*|}{|\text{Msg}|} + \mu$. Moreover, $\Pr[d : d = d^*] \geq \frac{\Pr[d \in D^*]}{|D^*|} = \frac{1}{|D^*|}(\frac{m|D^*|}{|\text{Msg}|} + \mu)$, where $|D^*|$ is the data set size. Therefore, if $\mathcal{A}$ wins the game for one-wayness with a non-negligible advantage μ , the challenger can break the one-wayness for H_2 with a non-negligible probability that is at least $\frac{1}{|D^*|}(\frac{m|D^*|}{|\text{Msg}|} + \mu)$. $\square$

Theorem 3. The CP-ABSI scheme is fine-grained authorization secure in the generic bilinear group model.

Proof. We first show that the CP-ABSI construction is fine-grained authorization secure while $|D_0| = |D_1| = 1$.

Setup: The challenger picks $a, b \xleftarrow{R} \mathbb{Z}_p$ and transfers $pk = (G, G_T, e, g, p, g^a, g^b)$ to $\mathcal{A}$. The challenger bookkeeps lists $List_{H_1}(at_j, \alpha_j)$ and $List_{H_2}(D, \gamma)$, which are empty initially, and $\mathcal{A}$ can query $\mathcal{O}_{H_1}$ and $\mathcal{O}_{H_2}$ in polynomially many times as follows:

$\mathcal{O}_{H_1}(at_j)$: Given the attribute at_j , if at_j has not been queried before, it picks $\alpha_j \xleftarrow{R} \mathbb{Z}_p$, replies g^{α_j} , and inserts (at_j, α_j) into $List_{H_1}$; Else, it replies g^{α_j} by searching α_j from $List_{H_1}$.

$\mathcal{O}_{H_2}(D)$: If D has been queried before, then it searches γ from $List_{H_2}$ and sends g^γ back to $\mathcal{A}$; otherwise, it picks $\gamma \xleftarrow{R} \mathbb{Z}_p$, replies g^γ to $\mathcal{A}$, and inserts (D, γ) into $List_{H_2}$.

Phase 1: The challenger bookkeeps two initially empty lists L_{UA} and L_{token} . $\mathcal{A}$ can query the oracles polynomially many times.

$\mathcal{O}_{KeyGen}(UA)$: It picks $t^{(u)} \xleftarrow{R} \mathbb{Z}_p$. For each $at_j \in UA$, it chooses $c_j^{(u)} \xleftarrow{R} \mathbb{Z}_p$, and sets $sk = (UA, X_1 = g^{at^{(u)}}, X_2 = g^{bt^{(u)}}, \{Y_j = g^{abt^{(u)} + \alpha_j c_j^{(u)}}, Z_j = g^{c_j^{(u)}} | at_j \in UA\})$. It returns sk to $\mathcal{A}$ and adds UA to L_{UA} .

$\mathcal{O}_{token}(UA)$: The challenger selects $k^{(u)} \xleftarrow{R} \mathbb{Z}_p$, runs $\mathcal{O}_{KeyGen}(UA)$. For each $at_j \in UA$, it chooses $c_j^{(u)} \xleftarrow{R} \mathbb{Z}_p$, and sets $token = (UA, \hat{X}_1 = X_1^{k^{(u)}} = g^{at^{(u)}k^{(u)}}, \hat{X}_2 = X_2^{k^{(u)}} = g^{bt^{(u)}k^{(u)}}, \{\hat{Y}_j = Y_j^{k^{(u)}} = g^{abt^{(u)}k^{(u)} + \alpha_j c_j^{(u)}k^{(u)}}, \hat{Z}_j = Z_j^{k^{(u)}} = g^{c_j^{(u)}k^{(u)}} | at_j \in UA\})$. It sends $token$ to $\mathcal{A}$ and inserts UA into L_{token} .

Challenge: The adversary $\mathcal{A}$ selects access control policies T_1^*, T_2^* and sends them to the challenger. We require that (1) $\forall UA \in L_{UA}$, $F(T_1^*, UA)$ and $F(T_2^*, UA)$ cannot output 1 simultaneously, and (2) $\forall UA \in L_{UA}$, $F(T_1^*, UA)$ and $F(T_2^*, UA)$ cannot output 1 simultaneously. The challenger chooses two data sets $(D_0 = (d_{0,0}, \dots, d_{0,n}), D_1 = (d_{1,0}, \dots, d_{1,n}))$ of equal length.

For each $d_{0,i}, 0 \leq i \leq n$, the challenger selects $r_1, r_2 \xleftarrow{R} \mathbb{Z}_p$ and runs $\{q_v(0) | v \in \text{leaf}(T_1^*)\} \leftarrow \text{Share}(T_1^*, r_2)$. It computes $c_{1,i}^* = (A_1 = g^{br_1}, A_2 = g^{a(r_1+r_2)+\beta_{0,i}}, B_v = g^{q_v(0)}, \{C_v = g^{\alpha_j q_v(0)} | v \in \text{leaf}(T_1^*)\})$, where $\text{att}(v) = at_j$.

It selects $\sigma \xleftarrow{R} \{0, 1\}$ and for each $d_{\sigma,w}, 0 \leq w \leq n$, it selects $r_3, r_4 \xleftarrow{R} \mathbb{Z}_p$ and runs $\{q'_v(0) | v \in \text{leaf}(T_2^*)\} \leftarrow \text{Share}(T_2^*, r_4)$. It calculates $cipher_{2,w}^* = (A'_1 = g^{br_3}, A'_2 = g^{a(r_3+r_4)+\beta_{\sigma,w}}, B'_v = g^{q'_v(0)}, \{C'_v = g^{\alpha_j q'_v(0)} | v \in \text{leaf}(T_2^*)\})$, where $\text{att}(v) = at_j$.

At last, it returns $c_1^* = (c_{1,0}^*, \dots, c_{1,n}^*), c_2^* = (c_{2,0}^*, \dots, c_{2,n}^*)$ to $\mathcal{A}$.

Phase 2: It is the same as that defined by the game.

Guess: A guess σ' of σ is output by $\mathcal{A}$.

In this game, $\mathcal{A}$ attempts to determine whether D_σ is equal to D_0 . Since each message in data sets is encrypted independently, $\mathcal{A}$ needs to determine whether $d_{\sigma,w}$ is equal to $d_{0,i}$ for $i, w \in \{0, \dots, n\}$. The only way is to judge whether or not g^{β_σ} is same as g^{β_0} . Since in the generic model, the only way that $\mathcal{A}$ can determine β_σ is equal to β_0 is that there are two queries $v = \gamma_1 + \gamma_2 \beta_{0,i}$ and $v' = \gamma_1 + \gamma_2 \beta_{\sigma,w}$ for some γ_1 and γ_2 into G or G_T . We artificially add the query $v - v' = \gamma_2(\beta_0 - \beta_{\sigma,w})$ to $\mathcal{A}$'s queries. Then, it will be shown that $\mathcal{A}$ never makes any query for $g^{\gamma_2(\beta_0 - \beta_{\sigma,w})}$. We can see that β_0 only appears in the term $a(r_1 + r_2) + \beta_{0,i}$ and $\beta_{\sigma,w}$ only appears in the term $a(r_3 + r_4) + \beta_{\sigma,w}$. So, $\mathcal{A}$ needs to construct $g^{\gamma_2 a(r_1+r_2-r_3-r_4)}$. Table 3 lists all the queries to group oracle G_T .

In the above game, r_1 only appears in the term br_1 and br_3 only appears in the term br_3 . Let $\gamma_2 = b\gamma'_2$ for some γ'_2 , and then $\mathcal{A}$ needs to construct $\gamma'_2 ab(r_2 - r_4)$. Since r_2 is independent with r_4 , $\mathcal{A}$ needs to construct $\gamma'_2 abr_2$ and $\gamma'_2 abr_4$. The only way to construct $\gamma'_2 abr_2$ needs to use $(q_v(0), \alpha_j q_v(0))$ and $(c_j^{(u)}, abt^{(u)} + \alpha_j c_j^{(u)}, abt^{(u)}k^{(u)} + \alpha_j c_j^{(u)}k^{(u)}, c_j^{(u)}k^{(u)})$

related to key or token queries labeled with $(UA^{(u)}, t^{(u)})$. $q_v(0)$ is r_2 's secret share associated to T_1 . So, $abt^{(u)}r_2$ or $abt^{(u)}k^{(u)}r_2$ can be only reconstructed if $F(UA^{(u)}, T_1)$. Let Ω is a set of $t^{(u)}$ such that γ'_2abr_2 for $\gamma' = t^{(u)}c$ or $\gamma' = k^{(u)}t^{(u)}c$, where g^c is known to $\mathcal{A}$, can be reconstructed. For any $t^{(u)} \in \Omega$, $\gamma'_2abr_4 = abr_4t^{(u)}c$ or $\gamma'_2abr_4 = abr_4t^{(u)}k^{(u)}c$ can be reconstructed only if the key or token query labeled with $(UA^{(u)}, t^{(u)})$ satisfies $F(UA^{(u)}, T_2)$. However, there is no key or token query labeled with $(UA^{(u)}, t^{(u)})$ satisfying not only $F(UA^{(u)}, T_1) = 1$ but also $F(UA^{(u)}, T_2) = 1$. Therefore, there is no γ'_2 such that γ'_2abr_2 and γ'_2abr_4 can both be reconstructed.

Consequently, $g^{\gamma_2a(r_1+r_2-r_3-r_4)}$ and $g^{\gamma_2(\beta_0-\beta_\sigma)}$ for some γ_2 can be constructed by $\mathcal{A}$ with negligible probability, and it is fine-grained authorization secure for $|D_0| = |D_1| = 1$.

As with the evidence in Theorem 1, it can be shown that our solution is also fine-grained authorization secure for $|D_0| = |D_1| > 1$. This completes the proof. $\square$

Table 3. Possible queries for group oracle G_T in fine-grained authorization game.

Queries 1	Queries 2	Queries 3	Queries 4	Queries 5
a	$c_j^{(u)}$	$c_j^{(u)}k^{(u)}$	$q_v(0)$	$\alpha_j q'_v(0)$
b	$at^{(u)}$	$at^{(u)}k^{(u)}$	$\alpha_j q_v(0)$	$a(r_1 + r_2) + \beta_0$
α_j	$bt^{(u)}$	$bt^{(u)}k^{(u)}$	br_3	$a(r_3 + r_4) + \beta_\sigma$
$abt^{(u)} + \alpha_j c_j^{(u)}$	$abt^{(u)}k^{(u)} + \alpha_j c_j^{(u)}k^{(u)}$	br_1	$q'_v(0)$	

5. Efficiency Analysis

Here, we measure the performance of CP-ABSI and other related works about attribute-based set intersection by the asymptotic complexity. Pairing operation time is denoted by P , hash operation time is denoted by H , group exponentiation time in G is denoted by E , and group exponentiation time in G_T is denoted by E_T . The multiplication operation is ignored since it has much higher efficiency compared to the above operations.

For analyzing the performance of CP-ABSI clearly, firstly we compared it with KP-ABSI [5], as in Table 4, as follows:

Table 4. Efficiency comparison between KP-ABSI and our CP-ABSI. Note that here, both sets are n in size, the number of a data user's attributes is S , the number of attributes involved in the access tree is N .

Scheme	KP-ABSI [5]	CP-ABSI
KeyGen	$(3N + 2)E + NH$	$(2S + 3)E + SH$
Enc	$(S + 3)nE + (S + 1)nH$	$(2N + 2)nE + (N + 1)nH$
TokenGen	$(2N + 2)E$	$(2S + 4)E$
SI	$(2S + 2) \cdot 2nP + S \cdot 2nE_T$	$(2N + 2) \cdot 2nP + N \cdot 2nE_T$

It shows that KeyGen in the CP-ABSI scheme is more expensive than that in the KP-ABSI scheme, while Enc in the CP-ABSI scheme is more expensive than that in the KP-ABSI scheme. In both schemes, the TokenGen algorithm and the SI algorithm incur almost the same cost. SI algorithm can be delegated to the cloud server. However, our CP-ABSI can specify the access structure over outsourced data sets directly, while KP-ABSI can only access control for users because of its key-policy setting.

Additionally, as we showed in Section 2, on one hand, Ali's solution [6] requires the data user to store complete sets of big data and spend more time than $nE + nH$ for PSI operation at local. This has a great burden on the computing and storage for the cloud users. On the other hand, ABEET solutions [12–14] cannot support multi-elements data

sets and fine-grained authorization security. Even worse, they require the data owners and trusted authority to always be online. So, the communication cost is too high and they are not applicable to the distributed cloud scenario.

Thus, our CP-ABSI is the most practical and secure solution than existing attribute-based set intersection schemes for cloud computing.

6. Conclusions

We introduced a novel application of multimedia cloud computing: ciphertext-policy attribute-based encryption with outsourced set intersection. This application has five properties: (1) All cloud users outsource their encrypted data sets to a cloud, where each outsourced data set has an associated access control policy. (2) A data user who has the credentials that satisfy the data owner's access control policy is able to delegate to the cloud the set intersection operation between his own and the data owner's data sets. (3) The cloud helps data users calculate the set intersection without being capable of learning any additional useful information and without being given the data user's credentials. (4) It is not required that the data owners and the trusted authority are always online. (5) It supports multi-elements data sets.

There are many interesting problems for future research in multimedia cloud computing. First, our schemes do not hide what credentials the data user possesses. How to hide such information is an open problem. Second, our main contribution is not for high performance, but for new cryptographic functions and security in this work. So, it is also interesting to design a high performance CP-ABSI scheme in future work.

Author Contributions: Conceptualization, Y.S.; methodology, Y.S.; formal analysis, S.Q.; writing—original draft preparation, Y.S.; writing—review and editing, Y.S. and S.Q. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Philosophy and Social Science Foundation of the Jiangsu Higher Education Institutions of China “Research on Blockchain-based Intelligent Credit Information System and its Privacy Preservation Mechanism” (Grants No. 2021SJA0448), the Natural Science Foundation of Jiangsu Province (Grants No. BK20210928), and the Scientific Research Foundation of Nanjing Institute of Technology (Grants No. YKJ201980), and the Higher Education Research Project of Nanjing Institute of Technology “Research on Security Sharing Mechanism of Compound Teaching Resources based on Blockchain” (Grants No. 2021ZC13).

Data Availability Statement: All data is available to the public.

Acknowledgments: This paper has benefited from the comments of the anonymous reviewers.

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

References

1. Freedman, M.J.; Nissim, K.; Pinkas, B. Efficient private matching and set intersection. In *Advances in Cryptology-EUROCRYPT 2004*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 1–19.
2. Wang, Q.; Zhou, F.; Xu, J.; Peng, S. Tag-based Verifiable Delegated Set Intersection over Outsourced Private Datasets. In *IEEE Transactions on Cloud Computing*; IEEE: Piscataway, NJ, USA, 2020; p. 1. [\[CrossRef\]](#)
3. Abadi, A.; Terzis, S.; Dong, C. VD-PSI: Verifiable delegated private set intersection on outsourced private datasets. In *International Conference on Financial Cryptography and Data Security*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 149–168.
4. Yang, X.; Luo, X.; Xu, A.W.; Zhang, S. Improved outsourced private set intersection protocol based on polynomial interpolation. *Concurr. Comput. Pract. Exp.* **2018**, *30*, e4329. [\[CrossRef\]](#)
5. Shi, Y.; Qiu, S. Delegated Key-Policy Attribute-Based Set Intersection over Outsourced Encrypted Data Sets for CloudIoT. *Secur. Commun. Netw.* **2021**, *2021*, 5595243. [\[CrossRef\]](#)
6. Ali, M.; Mohajeri, J.; Sadeghi, M.R.; Liu, X. Attribute-based fine-grained access control for outsourced private set intersection computation. *Inf. Sci.* **2020**, *536*, 222–243. [\[CrossRef\]](#)
7. Pinkas, B.; Rosulek, M.; Trieu, N.; Yanai, A. Psi from paxos: Fast, malicious private set intersection. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 739–767.
8. Lv, S.; Ye, J.; Yin, S.; Cheng, X.; Feng, C.; Liu, X.; Li, R.; Li, Z.; Liu, Z.; Zhou, L. Unbalanced private set intersection cardinality protocol with low communication cost. *Future Gener. Comput. Syst.* **2020**, *102*, 1054–1061. [\[CrossRef\]](#)

9. Zhu, H.; Chen, M.; Sun, M.; Liao, X.; Hu, L. Outsourcing Set Intersection Computation Based on Bloom Filter for Privacy Preservation in Multimedia Processing. *Secur. Commun. Netw.* **2018**, *2018*, 5841967. [\[CrossRef\]](#)
10. Duong, D.H.; Fukushima, K.; Kiyomoto, S.; Roy, P.S.; Susilo, W. A lattice-based public key encryption with equality test in standard model. In *Australasian Conference on Information Security and Privacy*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 138–155.
11. Lee, H.T.; Ling, S.; Seo, J.H.; Wang, H. Public key encryption with equality test from generic assumptions in the random oracle model. *Inf. Sci.* **2019**, *500*, 15–33. [\[CrossRef\]](#)
12. Zhu, H.; Wang, L.; Ahmad, H.; Niu, X. Key-policy attribute-based encryption with equality test in cloud computing. *IEEE Access* **2017**, *5*, 20428–20439. [\[CrossRef\]](#)
13. Wang, Q.; Peng, L.; Xiong, H.; Sun, J.; Qin, Z. Ciphertext-policy attribute-based encryption with delegated equality test in cloud computing. *IEEE Access* **2017**, *6*, 760–771. [\[CrossRef\]](#)
14. Cui, Y.; Huang, Q.; Huang, J.; Li, H.; Yang, G. Ciphertext-policy attribute-based encrypted data equality test and classification. *Comput. J.* **2019**, *62*, 1166–1177. [\[CrossRef\]](#)
15. Sahai, A.; Waters, B. Fuzzy identity-based encryption. In *Advances in Cryptology—EUROCRYPT 2005*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 457–473.
16. Liu, Z.; Duan, S.; Zhou, P.; Wang, B. Traceable-then-revocable ciphertext-policy attribute-based encryption scheme. *Future Gener. Comput. Syst.* **2019**, *93*, 903–913. [\[CrossRef\]](#)
17. Ma, H.; Wang, Z.; Guan, Z. Efficient Ciphertext-Policy Attribute-Based Online/Offline Encryption with User Revocation. *Secur. Commun. Netw.* **2019**, *2019*, 8093578. [\[CrossRef\]](#)
18. Li, J.; Yu, Q.; Zhang, Y.; Shen, J. Key-policy attribute-based encryption against continual auxiliary input leakage. *Inf. Sci.* **2019**, *470*, 175–188. [\[CrossRef\]](#)
19. Zhang, J.; Gao, H. A compact construction for non-monotonic key-policy attribute-based encryption. *Int. J. High Perform. Comput. Netw.* **2019**, *13*, 321–330. [\[CrossRef\]](#)
20. Zheng, Q.; Xu, S.; Ateniese, G. VABKS: Verifiable attribute-based keyword search over outsourced encrypted data. In *Proceedings of the IEEE INFOCOM 2014—IEEE Conference on Computer Communications*, Toronto, ON, Canada, 27 April–2 May 2014; pp. 522–530.
21. Li, Z.; Sharma, V.; Ma, C.; Ge, C.; Susilo, W. Ciphertext-policy attribute-based proxy re-encryption via constrained PRFs. *Sci. China Inf. Sci.* **2021**, *64*, 1–2. [\[CrossRef\]](#)