Hindawi
Security and Communication Networks
Volume 2021, Article ID 5595243, 11 pages
https://doi.org/10.1155/2021/5595243

Research Article

Delegated Key-Policy Attribute-Based Set Intersection over Outsourced Encrypted Data Sets for CloudIoT

Yanfeng Shi[1] **and Shuo Qiu**[2]

[1]*School of Computer Engineering, Nanjing Institute of Technology, Nanjing 211167, China*
[2]*School of Software Engineering, Jinling Institute of Technology, Nanjing 211169, China*

Correspondence should be addressed to Shuo Qiu; shuoqiu@jit.edu.cn

Received 8 January 2021; Revised 11 February 2021; Accepted 24 February 2021; Published 18 March 2021

Academic Editor: Zhaoqing Pan

Private set intersection (PSI) is a fundamental cryptographic primitive, allowing two parties to calculate the intersection of their data sets without exposing additional private information. In cloud-based IoT system, IoT-enabled devices would like to outsource their data sets in their encrypted form to the cloud. In this scenario, how to delegate the set intersection computation over outsourced encrypted data sets to the cloud and how to achieve the fine-grained access control for PSI without divulging any additional information to the cloud are still open problems. With that in mind, in this work, we combine key-policy attribute-based encryption (KP-ABE) and PSI to introduce such a novel concept, called *delegated key-policy attribute-based set intersection over outsourced encrypted data sets (KP-ABSI)*, to solve this problem. Then we propose a first concrete KP-ABSI scheme and analyze its efficiency.

1. Introduction

Internet of Things (IoT) is enabling Smart City initiatives all over the world. Recently, IoT-based applications have been widely developed, such as smart grid and smart healthcare [1, 2]. With the growth of IoT, enormous amount of data is generated by IoT-enabled devices. They need to be stored, processed, and accessed. Thus, Alessio et al. firstly merged cloud and IoT to introduce a new paradigm named CloudIoT to solve the issues [3]. For cloud-based IoT, the research on the security and privacy for the big data of IoT is a hot spot.

Private set intersection (PSI), firstly proposed by [4], is a special case of secure multiparty computation. It enables two parties to calculate the set intersection of their data sets under the condition of privacy preservation. It is applied to many practical scenarios, such as IoT and internet-based personal health record (PHR) systems. In traditional PSI solutions, the data users hold their own data sets. However, in CloudIoT computing, data users (i.e., IoT-enabled devices) with limited computing power and storage resources would like to outsource their data sets to the cloud. For confidentiality and privacy, data sets should be encrypted before outsourcing. Cloud service providers provide flexible services to fulfill cloud users' demand.

Based on this, we research on PSI over outsourced encrypted data sets in the CloudIoT system. In this scenario, the data users (i.e., the IoT-enabled devices) will encrypt and outsource their data sets to the cloud and then delegate the cloud to perform the set intersection. It has been studied by some works [5–7]. But it indeed raises a concern on how to enforce fine-grained access control for limiting the cloud's capability on computing set intersection. For this, Mohammad Ali et. al combined ciphertext-policy attribute-based encryption and private set intersection to propose an attribute-based private set intersection [8]. However, in their solution, the data user, who requests the set intersection operation, should hold the data sets in plaintext form. It does not really focus on outsourced encrypted data sets. Besides, there is still no solution for key-policy setting.

In this paper, we firstly combined key-policy attribute-based encryption and private set intersection to introduce a

novel concept called *delegated key-policy attribute-based set intersection over outsourced encrypted data sets* (KP-ABSI). KP-ABSI focuses on the problem of set intersection over outsourced data sets in the cloud paradigm. It allows data owners to specify some attributes set on his/her data set and encrypt it before outsourcing, respectively. A data user with proper access control policy (satisfied by the attribute set specified by the data owner and himself/herself) can generate a token to delegate the cloud sever to perform the set intersection over his/her and the data owner's outsourced encrypted data sets. We formally give the definition and security notion for KP-ABSI and propose a concrete construction.

Our KP-ABSI scheme has three distinctive properties: (1) Our solution realizes fine-grained authorization for set intersection over encrypted outsourced data sets by combining KP-ABE and PSI. (2) The cloud server cannot obtain any information about the plaintexts beyond the result of set intersection, which is also with the form of ciphertexts. (3) Compared with existing PSI schemes, our schemes do not require interaction with the data owner or the trusted authority.

2. Related Work

Although the scholars have carried out extensive research on PSI, the existing solutions cannot solve the problems considered in this paper. In the following part, we will briefly introduce the related works. In general, they can be divided into three categories as follows.

Two-Party Private Set Intersection. The traditional PSI has two participants, a data owner and a data user. Both of them hold their own data sets and interactively compute the set intersection [9, 10]. However, two-party PSI does not apply to cloud computing because two parties must hold their data sets by themselves.

Three-Party Private Set Intersection. Typically, three-party PSI involves three participants: a data owner, a data user, and the cloud server. The data user and the data owner would like to outsource their data sets to the cloud and delegate set intersection computation to the cloud. [5, 7, 11]. Moreover, public key encryption with equality test [12–15] can also be used to attain this goal. However, in these solutions, there is not any authorization mechanism and the data owner online is required to authorize the data user. So, they are not practical in the cloud computing.

Attribute-Based Encryption. ABE, which is introduced by Sahai and Waters, achieves fine-grained access control for outsourced data [16]. There are two variants of ABE: key-policy attribute-based encryption (KP-ABE) where the decryption key is associated with the access control policy (e.g., [17–19]) and ciphertext-policy attribute-based encryption (CP-ABE) where the ciphertext is associated with the access control policy (e.g., [20–22]). In 2017, Zhu et al. presented a key-policy attribute-based encryption with equality test, which can be utilized to do the set intersection over outsourced encrypted data set of one element. After that, Wang et al. proposed the first ciphertext-policy attribute-based encryption with equality test scheme [23, 24]. Later, Cui et al. improved its efficiency [25]. However, attribute-based encryption with equality test is only for one element. For this, in 2020, Mohammad Ali et. al firstly combined CP-ABE and PSI to propose an attribute-based set intersection scheme [8]. It achieves fine-grained access control for set intersection computation. Unfortunately, their solution requires the data user to hold his/her data set in the plaintext form. It did not really focus on outsourced encrypted data sets in the cloud computing. Moreover, there are no key-policy setting solutions for attribute-based set intersection.

Thus, in this paper, we combine KP-ABE with PSI to introduce a novel primitive-delegated key-policy attribute-based set intersection over outsourced encrypted data sets (KP-ABSI). For fairness, we summarize the properties of KP-ABSI scheme in Table 1.

3. Problem Formulation

3.1. System Model. The system model for KP-ABSI is shown in Figure 1. There are three participants: the trusted attribute authority, the cloud users (e.g., data owner Alice, authorized data user Bob, and unauthorized data user Carlos), and the cloud server. The trusted attribute authority primarily initiates the public parameters and issues private keys for data users according to their access control polices. Cloud server provides powerful storage and computing services for cloud users. The cloud users outsource their private data sets to the cloud server. Specifically, a cloud user, Alice, outsources her data set to the cloud in encrypted form, where the encryption is conducted according to some attribute set UAtt_A. An authorized user, Bob, whose access control policy is satisfied by the attribute set UAtt_A, can delegate to the cloud the computation of set intersection between Alice's outsourced encrypted sets and his own outsourced encrypted sets (Bob naturally has the private key to decrypt his own outsourced encrypted data). Meanwhile, any unauthorized user, Carlos, is neither able to decrypt Alice's outsourced encrypted data sets nor able to delegate the cloud to perform the set intersection operation.

In this model, we assume that the cloud is semitrusted (i.e., honest-but-curious), which means that the cloud honestly executes the protocol for two honest users, but tries to learn useful information beyond the ciphertexts through set intersection operations. Cloud users may be malicious and may collude with each other. We even allow a malicious user, say Bob, to collude with the semitrusted cloud. However, in this case, we cannot require that the cloud be not able to decrypt the honest user's, say Alice, ciphertext data set when the malicious and colluding user, Bob, has the private key for decrypting Alice's data set (e.g., Bob can simply give his private key to the cloud).

3.2. Functional Definition. In this part, we introduce the formal definition for delegated key-policy attribute-based set intersection over outsourced encrypted data sets (KP-ABSI),

Table 1: Property summary for PSI solutions in the literature and KP-ABSI in this paper. PSI delegation means that the data owners can delegate set intersection operations to the cloud. Outsourced encrypted data set means it can do PSI over outsourced encrypted data sets in the cloud. Fine-grained authorization means that it supports attribute-based access control policy.

Schemes	PSI delegation	Outsourced encrypted data sets	Fine-grained authorization
Two-party PSI [9, 10]	×	×	×
Three-party PSI [5, 12]	✓	✓	×
AB-PSI [8]	✓	×	✓
KP-ABSI	✓	✓	✓

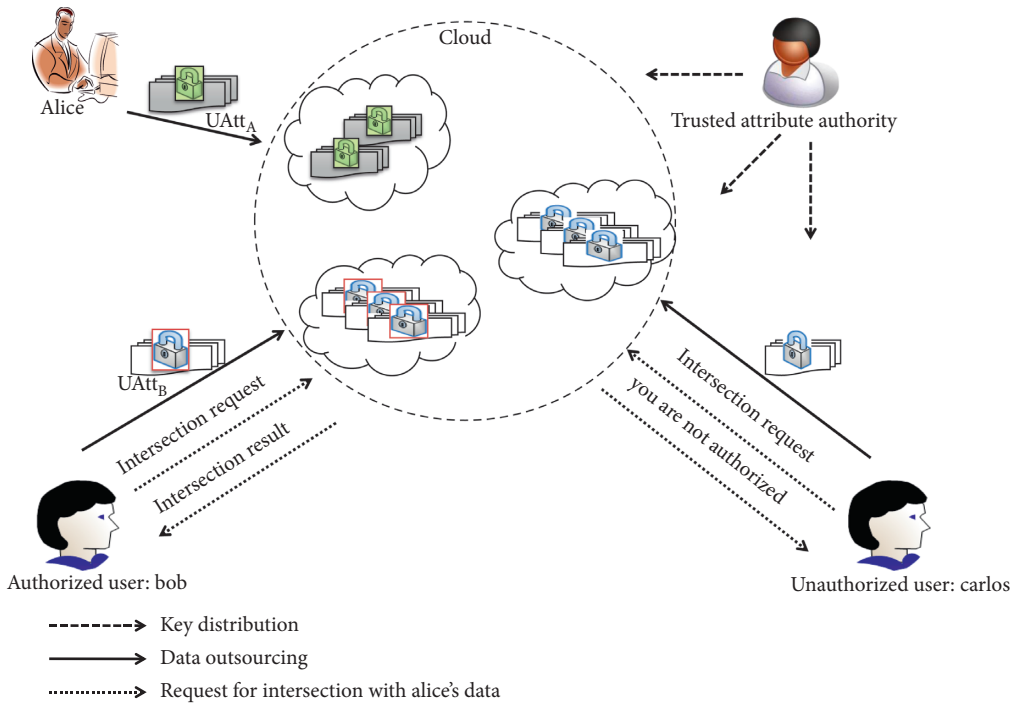

Figure 1: System model of delegated key-policy attribute-based set intersection over outsourced encrypted data sets.

where private keys are associated with access control policies. For convenience, we denote by UAtt an attribute set and by T an access policy in KP-ABSI. Let $F(\mathrm{UAtt}, T) = 1$ if and only if UAtt satisfies T in KP-ABSI.

Definition 1. Delegated key-policy attribute-based set intersection over outsourced encrypted data sets (KP-ABSI) includes five algorithms as follows:

$(\mathrm{pk}, \mathrm{mk}) \longleftarrow \mathrm{Setup}(1^{\ell})$: Given a security parameter ℓ as input, the trusted attribute authority initializes the system public parameters pk and the master secret key mk.

$\mathrm{sk} \longleftarrow \mathrm{KeyGen}(\mathrm{mk}, T)$: Given the master secret key mk and an access control policy T, the trusted attribute authority issues private keys sk for a data user.

$\mathrm{cph} \longleftarrow \mathrm{Enc}(D, \mathrm{UAtt})$: Given an attribute set UAtt, a data user encrypts his/her private data set D to the ciphertext cph. The resulting ciphertexts will be outsourced to the cloud.

$\mathrm{tkn} \longleftarrow \mathrm{TokenGen}(\mathrm{sk})$: With his/her private key sk, the data user generates a token tkn and delegates the set intersection computation to the cloud.

$\mathrm{rslt} \longleftarrow \mathrm{SI}(\mathrm{tkn}, \mathrm{cph}, \mathrm{cph}')$: The cloud utilizes tkn to compute, on behalf of two data users, the set intersection rslt only if the access control policy T corresponding to tkn satisfies both $F(\mathrm{UAtt}, T) = 1$ and $F(\mathrm{UAtt}', T) = 1$, where the attribute sets UAtt and UAtt$'$ are, respectively, specified by cph and cph$'$. We say that a KP-ABSI scheme is correct if the following holds: Given $(\mathrm{pk}, \mathrm{mk}) \longleftarrow \mathrm{Setup}(1^{\ell})$, $\mathrm{sk} \longleftarrow \mathrm{KeyGen}(\mathrm{mk}, T)$, $\mathrm{tkn} \longleftarrow \mathrm{TokenGen}(\mathrm{sk})$, and $\mathrm{cph} \longleftarrow \mathrm{Enc}(D, \mathrm{UAtt})$ for set D and $\mathrm{cph}' \longleftarrow \mathrm{Enc}(D', I_{\mathrm{Enc}}')$ for set D', if $F(\mathrm{UAtt}, T) = 1$ and $(\mathrm{UAtt}', T) = 1$, then rslt is the encrypted form of set intersection $D \cap D'$, where $\mathrm{rslt} \longleftarrow \mathrm{SI}(\mathrm{cph}, \mathrm{cph}', \mathrm{tkn})$.

3.3. Security Definitions. The security for KP-ABSI can be expressed by the three properties as follows.

3.3.1. Selective Security against Chosen-Plaintext Attack. It indicates that a probabilistic polynomial-time (PPT) adversary $\mathcal{A}$, without being given the corresponding tokens, is not able to obtain any useful information about the encrypted data sets. Notice that "selective" means that

adversary $\mathcal{A}$ should choose a target attribute set UAtt^* which it wants to challenge before the public parameters are generated. The security definition for selective security against chosen-plaintext attack can be formalized via the following game between an adversary $\mathcal{A}$ and a challenger.

Setup: $\mathcal{A}$ selects a target attribute set UAtt^* and sends it to the challenger. The challenger runs Setup algorithm to initialize pk and mk, sends pk to $\mathcal{A}$, and sets mk as the master private key.

Phase 1: The adversary $\mathcal{A}$ can make polynomial queries for the following oracles:

$\mathcal{O}_{\mathrm{KeyGen}}(T)$: If $F(\mathrm{UAtt}^*, T) = 1$, the challenger aborts; otherwise, the challenger returns $\mathrm{sk} \longleftarrow \mathrm{KeyGen}(\mathrm{mk}, \mathrm{pk}, T)$ to $\mathcal{A}$.

$\mathcal{O}_{\mathrm{TokenGen}}(T)$: If $F(\mathrm{UAtt}^*, T) = 1$, the challenger aborts; otherwise, the challenger calculates $\mathrm{sk} \longleftarrow \mathrm{KeyGen}(\mathrm{mk}, T)$ and returns $\mathrm{tkn} \longleftarrow \mathrm{TokenGen}(\mathrm{sk})$ to $\mathcal{A}$.

Challenge: The adversary $\mathcal{A}$ randomly gives two data sets D_0 and D_1, where $|D_0| = |D_1|$ but $D_0 \neq_R D_1$, to the challenger. Then, the challenger picks $\sigma \longleftarrow \{0, 1\}$ at random, builds the challenge ciphertext $\mathrm{cph}^* = \mathrm{Enc}(D_\sigma, \mathrm{UAtt}^*)$, and sends cph^* to $\mathcal{A}$.

Phase 2: Same as Phase 1.

Guess: $\mathcal{A}$ eventually outputs a guess σ' of σ. If $\sigma = \sigma'$, we say that $\mathcal{A}$ wins the game.

Definition 2. We say that a KP-ABSI scheme is selective secure against chosen-plaintext attack, if any PPT adversary $\mathcal{A}$ wins the above game with a negligible advantage, where the advantage can be described as $|\Pr[\sigma' = \sigma] - 1/2|$.

3.3.2. One-Way Security against Chosen-Plaintext Attack. It says that a PPT adversary $\mathcal{A}$, even given an appropriate token, cannot obtain the plaintexts corresponding to the ciphertexts. Note that the term "appropriate" means that the access control policy that generates the token is satisfied by the attribute set associated with the target ciphertext. Of course, $\mathcal{A}$ can choose a plaintext data set of its choice, encrypt it with public keys, and then utilize the token to check whether or not the target ciphertext is equal to the ciphertext of his choice. In other words, this type of brute-force attack is inherent to the set intersection problem and we can only demand that $\mathcal{A}$ cannot have any attack strategy significantly better than the brute-force attack, as captured by this property via the following game between an adversary $\mathcal{A}$ and a challenger.

Setup: The challenger runs Setup to initialize $(\mathrm{pk}, \mathrm{mk})$, sends pk to $\mathcal{A}$, and sets mk as the master private key.

Phase 1: The adversary $\mathcal{A}$ can make polynomial queries for the following oracles. Meanwhile, the challenger maintains a list L_T, which is initially empty.

$\mathcal{O}_{\mathrm{KeyGen}}(T)$: The challenger returns $\mathrm{sk} \longleftarrow \mathrm{KeyGen}(\mathrm{mk}, T)$ to $\mathcal{A}$ and records T to L_T.

$\mathcal{O}_{\mathrm{TokenGen}}(T)$: The challenger calculates $\mathrm{sk} \longleftarrow \mathrm{KeyGen}(\mathrm{mk}, T)$ and returns $\mathrm{tkn} \longleftarrow \mathrm{TokenGen}(\mathrm{sk})$ to $\mathcal{A}$.

Challenge: $\mathcal{A}$ gives a target attribute set UAtt^* to the challenger, where, $\forall T \in L_T$, $F(\mathrm{UAtt}^*, T) = 0$. The challenger selects an access control T^* such that $F(\mathrm{UAtt}^*, T^*) = 1$, picks D^* uniformly at random, runs $\mathrm{cph}^* \longleftarrow \mathrm{Enc}(D^*, \mathrm{UAtt}^*)$ and $\mathrm{tkn}^* \longleftarrow \mathrm{TokenGen}(\mathrm{KeyGen}(T^*))$, and returns $\mathrm{cph}^*, \mathrm{tkn}^*$ to $\mathcal{A}$.

Phase 2: $\mathcal{A}$ executes the same as in Phase 1, except that $F(\mathrm{UAtt}^*, T) = 0$ when querying $\mathcal{O}_{\mathrm{KeyGen}}(T)$.

Guess: $\mathcal{A}$ outputs a guess d. If $d \in D^*$, we say that $\mathcal{A}$ wins the game.

Definition 3. We say that a KP-ABSI scheme achieves one-way security against chosen-plaintext attack if, for any PPT adversary $\mathcal{A}$, the advantage of $\mathcal{A}$ winning the game is negligible, where the advantage is defined as $|\Pr[d \in D^*] - (m|D^*|/|\mathrm{Msg}|)|$, where m is the number of guess/brute-force attacks $\mathcal{A}$ makes, and Msg is the message space of set elements.

3.3.3. Fine-Grained Authorization Security. This property says that the cloud is unable to utilize the given tokens to conduct set intersection over ciphertexts if no access control policy (associated with the data user's private key) that generates the tokens is satisfied by both of the attribute sets associated with the two ciphertexts. More specifically, consider the token tkn_1 that can be used to conduct set intersection over ciphertexts cph_1 and cph_2 and the token tkn_2 that can be used to conduct set intersection over ciphertexts cph_2 and cph_3. If the access control policy that is used to generate tkn_1 is not satisfied by the attribute set associated with ciphertext cph_3, and the access control policy that is used to generate tkn_2 is not satisfied by the attribute set associated with ciphertext cph_1; then the cloud cannot do the set intersection computation over ciphertexts cph_1 and cph_3 by using tkn_1 and/or tkn_2. The definition for fine-grained authorization security can be described via a game between an adversary $\mathcal{A}$ and a challenger.

Setup: The challenger runs Setup to initialize $(\mathrm{pk}, \mathrm{mk})$, sends pk to $\mathcal{A}$, and sets mk as the master secret key.

Phase 1: The adversary $\mathcal{A}$ makes polynomial queries for the following oracles. Meanwhile, the challenger maintains two lists L_T and L_{tkn}, which are empty initially.

$\mathcal{O}_{\mathrm{KeyGen}}(T)$: The challenger returns $\mathrm{sk} \longleftarrow \mathrm{KeyGen}(\mathrm{mk}, T)$ to $\mathcal{A}$ and records T to L_T.

$\mathcal{O}_{\mathrm{TokenGen}}(T)$: The challenger runs $\mathrm{sk} \longleftarrow \mathrm{KeyGen}(\mathrm{mk}, T)$ and $\mathrm{tkn} \longleftarrow \mathrm{TokenGen}(\mathrm{sk})$, returns tkn back to $\mathcal{A}$, and records T to L_{tkn}.

Challenge: $\mathcal{A}$ gives two target attribute sets UAtt_1^* and UAtt_2^* to the challenger. Then the challenger chooses two data sets D_0, D_1, picks a bit $\sigma \overset{R}{\longleftarrow} \{0, 1\}$ randomly,

runs $\text{cph}_1^* \longleftarrow \text{Enc}(D_0, \text{UAtt}_1^*)$ and $\text{cph}_2^* \longleftarrow \text{Enc}(D_\sigma, \text{UAtt}_2^*)$, and returns $\text{cph}_1^*, \text{cph}_2^*$ to $\mathcal{A}$. Here, we require that

$\forall T \in L_T$, $F(\text{UAtt}_1^*, T)$ and $F(\text{UAtt}_2^*, T)$ do not output 1 simultaneously;
$\forall T \in L_{\text{tkn}}$, $F(\text{UAtt}_1^*, T)$ and $F(\text{UAtt}_2^*, T)$ do not output 1 simultaneously.

Phase 2: $\mathcal{A}$ executes the same as in Phase 1, except for the following:

When querying $\mathcal{O}_{\text{KeyGen}}(T)$, $F(\text{UAtt}_1^*, T)$ and $F(\text{UAtt}_2^*, T)$ do not output 1 simultaneously.
When querying $\mathcal{O}_{\text{TokenGen}}(T)$, $F(\text{UAtt}_1^*, T)$ and $F(\text{UAtt}_2^*, T)$ do not output 1 simultaneously.

Guess: The adversary $\mathcal{A}$ eventually outputs a guess σ'. If $\sigma = \sigma'$, we say that $\mathcal{A}$ wins the game.

Definition 4. If, for any PPT adversary $\mathcal{A}$, the advantage of $\mathcal{A}$ winning the game is negligible, where the advantage can be expressed as $|\Pr[\sigma' = \sigma] - (1/2)|$, we say that a KP-ABSI scheme achieves fine-grained authorization security.

4. Scheme Construction

4.1. Basic Idea. To illustrate the idea, let a data user's private key be (g^{at}, g^{bt}), which can be generated by running $\{q_v(0) \mid v \in \text{lvs}(T)\} \longleftarrow \text{Share}(T, \text{abt})$ for some random t and setting $(g^{q_v(0)} H_1(\text{att}(v))^{t_v}, g^{t_v})$, where t_v is a random number with respect to leaf $v \in \text{lvs}(T)$. A set element d is encrypted into two parts:

The first part is related to d; namely,

$$\left(g^{br_1}, g^{a(r_1+r_2)} H_2(d), g^{r_2}\right), \tag{1}$$

where g is a generator of G, r_1 and r_2 are two random numbers, H_1, H_2 are two hash functions, and a, b are private keys.

The second part is related to the attribute set corresponding to the access control policy in question, namely, $H_1(at_i)^{r_2}$ for $at_i \in \text{UAtt}$.

A data user can generate the token as $\left\{g^{atk}, g^{btk}, (g^{q_v(0)k} H_1(\text{att}(v))^{kt_v}, g^{kt_v})\right\}$, by which the cloud is able to translate the ciphertext into an intermediate form $e(H_2(d), g)^{btk}$ once the attribute set UAtt satisfies the access control policy T.

4.2. KP-ABSI Construction

Setup(1^ℓ): Given the security parameter ℓ as input, the public parameters and the master secret key can be generated as follows:

Let $(e, q, g, g_T, G, G_T) \longleftarrow \text{BMapGen}(1^\ell)$.

Let H_1: $\{0,1\}^* \longrightarrow G$ and H_2: $\{0,1\}^* \longrightarrow G$ be two secure hash functions that are modeled as random oracles.
Select $a, b \xleftarrow{R} \mathbb{Z}_p$ and set the public parameters and the master secret key as

$$\begin{aligned} \text{pk} &= \left(e, G, G_T, g, g^a, g^b, H_1, H_2\right), \\ \text{mk} &= (a, b). \end{aligned} \tag{2}$$

KeyGen(mk, T): Given access tree T, this algorithm selects $t \xleftarrow{R} \mathbb{Z}_p$, computes $X_1 = g^{at}$ and $X_2 = g^{bt}$, and runs $\{q_v(0) \mid v \in \text{lvs}(T)\} \longleftarrow \text{Share}(T, abt)$. Then, for each leaf $v \in \text{lvs}(T)$, the algorithm selects $t_v \xleftarrow{R} \mathbb{Z}_p$ and sets $Y_v = g^{q_v(0)} H_1(\text{att}(v))^{t_v}$ and $Z_v = g^{t_v}$. The secret key is

$$\text{sk} = \left(T, X_1, X_2, \{(Y_v, Z_v) \mid v \in \text{lvs}(T)\}\right). \tag{3}$$

Enc(D, UAtt): Given set for outsourcing, $D = \{d_0, \ldots, d_n\}$, this algorithm encrypts the set as follows: for each d_j, it selects $r_1, r_2 \xleftarrow{R} \mathbb{Z}_p$, sets $A_1 = g^{br_1}$, $A_2 = g^{a(r_1+r_2)} H_2(d_j)$, and $A_3 = g^{r_2}$, and computes $B_i = H_1(at_i)^{r_2}$ for each $at_i \in \text{UAtt}$. The ciphertext of d_j is

$$\text{cph}_j = \left(\text{UAtt}, A_1, A_2, A_3, \{B_i \mid at_i \in \text{UAtt}\}\right). \tag{4}$$

The set of ciphertexts is $\text{cph} = \{\text{cph}_0, \ldots, \text{cph}_n\}$.

TokenGen(sk): Given secret key sk, this algorithm selects $k \xleftarrow{R} \mathbb{Z}_p$, sets $\widehat{X}_1 = X_1^k = g^{atk}$ and $\widehat{X}_2 = X_2^k = g^{btk}$, and computes $\widehat{Y}_v = Y_v^k$ and $\widehat{Z}_v = Z_v^k$ for leaf $v \in \text{lvs}(T)$. The token is

$$\text{tkn} = \left(T, \widehat{X}_1, \widehat{X}_2, \left\{\left(\widehat{Y}_v, \widehat{Z}_v\right) \mid v \in \text{lvs}(T)\right\}\right). \tag{5}$$

SI$(\text{tkn}, \text{cph}, \text{cph}')$: Given $\text{cph} = \{\text{cph}_0, \ldots, \text{cph}_n\}$, $\text{cph}' = \{\text{cph}_0', \ldots, \text{cph}_m'\}$, and tkn, this algorithm is executed as follows:

Given $\text{cph}_j \in \text{cph}$, it selects an attribute set $S \in \text{UAtt}$ satisfying T. If S does not exist, it returns 0. Otherwise, it computes

$$E_v = \frac{e\left(\widehat{Y}_v, A_3\right)}{e\left(\widehat{Z}_v, B_i\right)} = e(g, g)^{q_v(0) k r_2}, \tag{6}$$

for all $v \in \text{lvs}(T)$ with $\text{att}(v) = at_i$, computes

$$\begin{aligned} E_{\text{root}} &= e(g, g)^{abtkr_2} \longleftarrow \text{Combine}\left(T, E_v \mid \text{att}(v) \in S\right), \\ E_1 &= e\left(A_1, \widehat{X}_1\right) = e(g, g)^{abtkr_1}, \\ E_{2_j} &= e\frac{\left(A_2, \widehat{X}_2\right)}{\left(E_{\text{root}} E_1\right)} = e\left(H_2\left(d_j\right), g\right)^{btk}, \end{aligned} \tag{7}$$

and sets $E = \{E_{20}, \ldots, E_{2n}\}$.

Given $\mathrm{cph}_j' \in \mathrm{cph}'$, it selects an attribute set $S' \in \mathrm{UAtt}'$ satisfying T. If S' does not exist, it returns 0. Otherwise, it computes $E_v' = e(\widehat{Y}_v, A_3')/e(\widehat{Z}_v, B_i) = e(g,g)^{q_v(0)kr_2'}$, for all $v \in \mathrm{lvs}(T)$ with $\mathrm{att}(v) = at_i$, computes

$$E_{\mathrm{root}}' = e(g,g)^{abtkr_2'} \longleftarrow \mathrm{Combine}\left(T, E_v' \mid \mathrm{att}(v) \in S'\right),$$

$$E_1' = e\left(A_1', \widehat{X}_1\right) = e(g,g)^{abtkr_1'},$$

$$E_{2j}' = \frac{e\left(A_2, \widehat{X}_2\right)}{\left(E_{\mathrm{root}}' E_1'\right)} = e\left(H_2\left(d_j'\right), g\right)^{btk}, \tag{8}$$

and sets $E' = \{E_{20}', \ldots, E_{2m}'\}$.

Output the set intersection $\mathrm{rslt} = \left\{\mathrm{cph}_j \mid \mathrm{cph}_j \in \mathrm{cph}, E_{2j} \in E \cap E'\right\}$.

The correctness of the above KP-ABSI scheme can be verified by following the protocol. In what follows, we analyze its security.

4.3. Security Analysis

Theorem 1. *Under the DLN assumption, the above KP-ABSI scheme achieves the selective security against chosen-plaintext attacks in the random oracle as specified in Definition 2.*

Firstly, we prove that our scheme achieves the security goal when $|D| = 1$ (i.e., the message space has a single element) and then extends the proof to the case $|D| > 1$.

Proof. We show that if there is a PPT adversary $\mathcal{A}$ that wins the selective security game with a nonnegligible advantage μ, then a challenger that can solve the DLN problem with the advantage at least $\mu/2$ can be constructed. Specifically, given a DLN instance $(g, h, f, f^{r_1}, g^{r_2}, Q)$, where $g, f, h, Q \xleftarrow{R} G$ and $r_1, r_2 \xleftarrow{R} \mathbb{Z}_p$ are unknown, the game is simulated by the challenger as follows.

Setup: The adversary $\mathcal{A}$ gives an attribute set UAtt^* to the challenger. Then the challenger produces the bilinear map $e: G \times G \longrightarrow G_T$, constructs $g^b = f$ and $g^a = h$ with a and b unknown, and sets $\mathrm{pk} = (e, g, f, h, G, G_T)$. The challenger sends pk to $\mathcal{A}$. The challenger maintains two lists $\mathrm{List}_{H_1}(at_i, \alpha_i, \beta_i)$ and $\mathrm{List}_{H_2}(d, \gamma)$, which are initially empty. $\mathcal{A}$ can query $\mathcal{O}_{H_1}$ and $\mathcal{O}_{H_2}$ polynomially many times as follows:

$\mathcal{O}_{H_1}(at_i)$: Given attribute at_i, the challenger responds as follows:

The case at_i was queried before: it retrieves α_i, β_i from List_{H_1} and returns $f^{\alpha_i} g^{\beta_i}$ to $\mathcal{A}$.

The case at_i was not queried before: if $at_i \in \mathrm{UAtt}^*$, it selects $\beta_i \xleftarrow{R} \mathbb{Z}_p$, adds $(at_i, \alpha_i = 0, \beta_i)$ to List_{H_1}, and returns g^{β_i} to $\mathcal{A}$; otherwise, it selects $\alpha_i, \beta_i \xleftarrow{R} \mathbb{Z}_p$, adds $(at_i, \alpha_i, \beta_i)$ to List_{H_1}, and returns $f^{\alpha_i} g^{\beta_i}$ to $\mathcal{A}$.

$\mathcal{O}_{H_2}(d)$: Given a message d as input, if d was queried before, it retrieves γ from List_{H_2} and returns g^γ to $\mathcal{A}$; otherwise, it picks $\gamma \xleftarrow{R} \mathbb{Z}_p$ randomly, records (d, γ) to List_{H_2}, and returns g^γ to $\mathcal{A}$.

Phase 1: The adversary $\mathcal{A}$ makes polynomial queries for the following oracles as follows:

$\mathcal{O}_{\mathrm{KeyGen}}(T)$: If $F(\mathrm{UAtt}^*, T) = 1$, the simulation is aborted; otherwise, the challenger produces sk according to the two following procedures:

$\mathrm{PolySat}(T_v, \mathrm{UAtt}^*, \lambda_v)$: Given a secret value λ_v, this procedure builds the polynomial for each node of subtree T_v, where $F(\mathrm{UAtt}^*, T_v) = 1$. Suppose that the threshold value of node v is k_v; it lets $q_v(0) = \lambda_v$ and chooses $k_v - 1$ coefficients uniformly at random to uniquely determine the polynomial q_v. Then it recursively runs $\mathrm{PolySat}(T_{v'}, \mathrm{UAtt}^*, \lambda_{v'})$ to build the polynomial for each child node v' of v, by letting $\lambda_{v'} = q_v(\mathrm{index}(v'))$.

$\mathrm{PolyUnsat}(T_v, \mathrm{UAtt}^*, g^{\lambda_v})$: Given an element $g^{\lambda_v} \in G$, where λ_v is unknown to the challenger, this procedure is to build the polynomial for each node of subtree T_v, where $F(\mathrm{UAtt}^*, T_v) = 0$. Assume that the threshold value of node v is k_v and V is the set of children of node v such that, $\forall v' \in V, F(\mathrm{UAtt}^*, T_{v'}) = 1$. Since $F(\mathrm{UAtt}^*, T_v) = 0$, we have $|V| < k_v$. For each $v' \in V$, it selects $\lambda_{v'} \xleftarrow{R} \mathbb{Z}_p$ and sets $\lambda_{v'} = q_v(\mathrm{index}(v'))$. It then determines the other $k_v - |V|$ points of polynomial q_v such that $g^{q_v(0)} = g^{\lambda_v}$. For each child node v' of node v, it executes the following:

(i) If v' is a node with $F(at^*, T_{v'}) = 1$, it runs $\mathrm{PolySat}(T_{v'}, \mathrm{UAtt}^*, q_v(\mathrm{index}(v')))$, where $q_v(\mathrm{index}(v'))$ is known.

(ii) If v' is a node with $F(at^*, T_{v'}) = 0$, it runs $\mathrm{PolySat}(T_{v'}, \mathrm{UAtt}^*, g^{\lambda_{v'}})$, where $g^{\lambda_{v'}} = g^{q_v(\mathrm{index}(v'))}$ is known.

Based on the two procedures above, the challenger executes $\mathrm{PolyUnsat}(T, \mathrm{UAtt}^*, g^a)$ by implicitly defining $q_{\mathrm{root}}(0) = a$. Note that, for each $v \in \mathrm{lvs}(T)$, the challenger knows $q_v(0)$ if $\mathrm{att}(v) \in \mathrm{UAtt}^*$ and knows $g^{q_v(0)}$ otherwise. Therefore, it generates credentials as follows by selecting $t \xleftarrow{R} \mathbb{Z}_p$ and setting $X_1 = h^t$ and $X_2 = f^t$ (while noting that $btq_v(0)$ is the secret share of abt):

If $\mathrm{att}(v) = at_j$ for some $at_j \in \mathrm{UAtt}^*$, it selects $t_v \xleftarrow{R} \mathbb{Z}_p$ and sets $Y_v = f^{tq_v(0)} g^{\beta_j t_v} = g^{btq_v(0)} H_1(\mathrm{att}(v))^{t_v}$ and $Z_v = g^{t_v}$. If $\mathrm{att}(v) \notin \mathrm{UAtt}^*$, where $\mathrm{att}(v) = at_j$ for some j, it selects $t_v' \xleftarrow{R} \mathbb{Z}_p$ and sets $Y_v = f^{tt_{v'}\alpha_j} (g^{q_v(0)})^{-(\beta_j/\alpha_j)} g^{t_{v'}\beta_j}$ and $Z_v = g^{-(q_v(0)/\alpha_j)} g^{t_{v'}}$. Note that (Y_v, Z_v) is valid because the challenger implicitly sets $t_v = -(q_v(0)/\alpha_j) + t_v'$ and the following:

$$\begin{aligned} Y_v &= f^{tt_v'\alpha_j} \left(g^{q_v(0)}\right)^{-\left(\beta_j/\alpha_j\right)} g^{t_v'\beta_j} = f^{tq_v(0)} \left(f^{\alpha_j} g^{\beta_j}\right)^{-\left(q_v(0)/\alpha_j\right)+t_v'} \\ &= f^{tq_v(0)} H_1\left(at_j\right)^{t_v}, \\ Z_v &= g^{-\left(q_v(0)/\alpha_j\right)} g^{t_v'} = g^{t_v}. \end{aligned} \tag{9}$$

$\mathcal{O}_{\mathrm{tkn}}(T)$: The challenger queries $\mathcal{O}_{\mathrm{KeyGen}}(T)$ to obtain sk and returns $\mathrm{tkn} \longleftarrow \mathrm{TokenGen}(\mathrm{sk})$ to $\mathcal{A}$.

Challenge: The adversary $\mathcal{A}$ gives two messages $D_0 = \{d_0\}$ and $D_1 = \{d_1\}$ of equal length to the challenger. Then the challenger randomly picks $\sigma \xleftarrow{R} \{0,1\}$, encrypts d_σ to $\mathrm{cph}^* = (\mathrm{UAtt}^*, f^{r_1}, QH_2(d_\sigma), g^{r_2}, \{(g^{r_2})^{\beta_j} \mid at_j \in \mathrm{UAtt}^*\})$, and sends cph^* to $\mathcal{A}$.

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

Guess: The adversary $\mathcal{A}$ will eventually output a guess σ' of σ. If $\sigma' = \sigma$, the challenger outputs $Q = h^{r_1+r_2}$; otherwise, it outputs $Q \neq h^{r_1+r_2}$.

The simulation is completed. In Challenge phase, if $Q = h^{r_1+r_2}$, then cph^* is indeed a valid ciphertext of D_σ and the probability that $\mathcal{A}$ outputs $\sigma' = \sigma$ is $(1/2) + \mu$. Otherwise, if Q is a random element from G, then cph^* is a random group element and the probability that $\mathcal{A}$ outputs $\sigma' = \sigma$ is $(1/2)$. In conclusion, the probability that the challenger correctly guesses $Q \stackrel{?}{=} h^{r_1+r_2}$ is $(1/2)((1/2) + (1/2) + \mu) = (1/2) + (\mu/2)$. That is, if $\mathcal{A}$ wins the game with the advantage μ, then the challenger solves the DLN problem with the advantage $(\mu/2)$.

So far, we have shown that our KP-ABSI scheme is selective secure against chosen-plaintext attack when $|D| = 1$. In what follows, we prove that our KP-ABSI scheme achieves the selective security against chosen-plaintext attack for the general case of $|D| > 1$.

Suppose that $\mathcal{A}$ gives two sets $D_0 = (d_0, \ldots, d_n)$ and $D_1 = (d_0', \ldots, d_n')$. Denote by $\mathrm{cph}^{(i)}$ the encryption of $(d_0, \ldots, d_i, d_{i+1}', \ldots, d_n')$, meaning that $\mathrm{cph}^{(n)}$ is the encryption of D_0 and $\mathrm{cph}^{(-1)}$ is the encryption of D_1. The Challenge phase is extended to accommodate an additional adversary $\mathcal{A}'$ as follows:

$\mathcal{A}'$ picks a random index $i \xleftarrow{R} [0, n]$ and presents (d_i, d_i') to the challenger. Then the challenger sends back cph_i to $\mathcal{A}'$ by encrypting d_i if $\sigma = 0$ or returns d_i' if $\sigma = 1$.

$\mathcal{A}'$ encrypts $(d_0, \ldots, d_{i-1})$ and $(d_{i+1}', \ldots, d_n')$ and returns $(\mathrm{cph}_0, \ldots, \mathrm{cph}_n)$ to $\mathcal{A}$. $\mathcal{A}'$ outputs $\mathcal{A}$'s output σ'.

Note that $\mathcal{A}'$ sends to $\mathcal{A}$ the ciphertext $\mathrm{cph}^{(i)}$ if $\sigma = 0$ and the ciphertext $\mathrm{cph}^{(i-1)}$ if $\sigma = 1$. Denote by $\mathcal{A}(\mathrm{cph}^{(i)})$ the guess of $\mathcal{A}$ with ciphertexts $\mathrm{cph}^{(i)}$. Then we show the probability that $\mathcal{A}'$ wins the game. Note that

$$\begin{aligned} \Pr[\mathcal{A}' \text{ outputs } 0 \mid \sigma = 0] &= \sum_{j=0}^{n} \frac{1}{n+1} \Pr\left[\mathcal{A}\left(\mathrm{cph}^{(j)}\right) = 0\right], \\ \Pr[\mathcal{A}' \text{ outputs } 0 \mid \sigma = 1] &= \sum_{j=0}^{n} \frac{1}{n+1} \Pr\left[\mathcal{A}\left(\mathrm{cph}^{(j-1)}\right) = 1\right]. \end{aligned} \tag{10}$$

Therefore, the probability that $\mathcal{A}'$ wins the game is

$$\begin{aligned} &\frac{1}{2}\Pr[\mathcal{A}'(\mathrm{cph}) = 0 \mid \sigma = 0] + \frac{1}{2}\Pr[\mathcal{A}'(\mathrm{cph}) = 1 \mid \sigma = 1] \\ &= \sum_{j=0}^{n} \frac{1}{2(n+1)} \Pr\left[\mathcal{A}\left(\mathrm{cph}^{(j)}\right) = 0\right] \\ &\quad + \sum_{j=0}^{n} \frac{1}{2(n+1)} \Pr\left[\mathcal{A}\left(\mathrm{cph}^{(j-1)}\right) = 1\right] \\ &= \frac{n}{2(n+1)} + \frac{1}{2(n+1)} \Pr\left[\mathcal{A}\left(\mathrm{cph}^{(n)}\right) = 0\right] \\ &\quad + \frac{1}{2(n+1)} \Pr\left[\mathcal{A}\left(\mathrm{cph}^{(-1)}\right) = 1\right] \\ &= \frac{n}{2(n+1)} + \frac{1}{(n+1)} \left(\frac{1}{2} \Pr\left[\mathcal{A}\left(\mathrm{cph}^{(n)}\right) = 0\right]\right. \\ &\quad \left. + \frac{1}{2} \Pr\left[\mathcal{A}\left(\mathrm{cph}^{(-1)}\right) = 1\right]\right) \\ &\leq \frac{1}{2} + \varepsilon, \end{aligned} \tag{11}$$

where ε is negligible because the advantage that $\mathcal{A}'$ wins the game is negligible. Thus, the probability that $\mathcal{A}$ distinguishes cph^{-1} from cph^n is

$$\frac{1}{2} \Pr\left[\mathcal{A}\left(\mathrm{cph}^{(n)}\right) = 0\right] + \frac{1}{2} \Pr\left[\mathcal{A}\left(\mathrm{cph}^{(-1)}\right) = 1\right] \leq \frac{1}{2} + (n+1)\varepsilon. \tag{12}$$

That is, the advantage of $\mathcal{A}$ distinguishing cph^{-1} from cph^n is at most $(n+1)\varepsilon$. Therefore, the scheme achieves the selective security against chosen-plaintext attack when $|D| \geq 1$. □

Theorem 2. *Given one-way hash function H_2, the above KP-ABSI scheme achieves the one-way security against chosen-plaintext attack as specified in Definition 3.*

Proof. We prove this theorem by showing that if there is a PPT adversary $\mathcal{A}$ winning the one-way security game against chosen-plaintext attack with a nonnegligible advantage μ,

then a challenger breaking the one-way hash function H_2 can be simulated.

Given $H_2(d^*) = y^*$, the challenger can simulate the one-way security game as follows:

Setup: The challenger randomly picks $a, b \xleftarrow{R} \mathbb{Z}_p$, produces $\mathrm{pk} = (e, G, G_T, g, p, g^a, g^b)$, $\mathrm{mk} = (a, b)$, and sends pk to $\mathcal{A}$.

Phase 1: The challenger maintains a list L_T, which is empty initially. $\mathcal{A}$ makes polynomial queries for the following oracles:

$\mathcal{O}_{\mathrm{KeyGen}}(T)$: Given an access control policy T, it returns $\mathrm{sk} \longleftarrow \mathrm{KeyGen}(\mathrm{mk}, \mathrm{pk}, T)$ to $\mathcal{A}$ and adds T to L_T.

$\mathcal{O}_{\mathrm{tkn}}(T)$: Given an access control policy T, it runs $\mathrm{tkn} \longleftarrow \mathrm{TokenGen}(\mathrm{KeyGen}(\mathrm{mk}, \mathrm{pk}, T), \mathrm{pk})$ and returns tkn to $\mathcal{A}$.

Challenge: $\mathcal{A}$ gives an attribute set UAtt^* to the challenger, where, $\forall T \in L_T$, $F(\mathrm{UAtt}^*, T) = 0$. The challenger picks $j \xleftarrow{R} \mathbb{Z}_p$ and sets a data set $D^* = (d_0, d_1, \ldots, d_{|D^*|-1})$, where $d_0, \ldots, d_{j-1}, d_{j+1}, \ldots, d_{|D^*|-1}$ are randomly chosen from the message space Msg and d_j is implicitly set as $d_j = d^*$ and generates the ciphertext $\mathrm{cph} = \{\mathrm{cph}_k\}_{k \in [0, |D^*|-1]}$ as follows:

If $k \neq j$, cph_k is generated the same as in the real construction.

If $k = j$,

$$\mathrm{cph}_k = \Big(\mathrm{UAtt}^*, A_1 = g^{br_1}, A_2 = g^{a(r_1+r_2)} y^*, A_3 = g^{r_2}, B_i = \{H_1(at_i)^{r_2} \mid at_i \in \mathrm{UAtt}^*\}\Big), \tag{13}$$

by randomly choosing $r_1, r_2 \xleftarrow{R} \mathbb{Z}_p$ and implicitly setting $d_j = d^*$.

The challenger chooses an access tree T^* satisfying $F(\mathrm{UAtt}^*, T^*) = 1$, runs $\mathrm{tkn}^* \longleftarrow \mathrm{TokenGen}(\mathrm{KeyGen}(\mathrm{mk}, \mathrm{pk}, T^*), \mathrm{pk})$, and returns $(\mathrm{cph}^* = (\mathrm{cph}), \mathrm{tkn}^*)$ to the adversary $\mathcal{A}$.

Phase 2: The adversary $\mathcal{A}$ executes the same as in Phase 1 while complying with the necessary requirements defined by the game.

Guess: $\mathcal{A}$ will eventually output a guess d to the challenger. The challenger wins if $d = d^*$.

The simulation is completed. If the probability that $\mathcal{A}$ outputs $d \in D^*$ is $|\Pr[d \in D^*] - (m|D^*|/|\mathrm{Msg}|)| = \mu$, then $\Pr[d \in D^*] = (m|D^*|/|\mathrm{Msg}|) + \mu$. Since the data set size is $|D^*|$, $\Pr[d\colon d = d^*] \geq (\Pr[d \in D^*]/|D^*|) = (1/|D^*|)((m|D^*|/|\mathrm{Msg}|) + \mu)$. Therefore, if $\mathcal{A}$ wins the one-way security game against chosen-plaintext attack with a non-negligible advantage μ, the one-way hash function H_2 can be broken by the challenger with a nonnegligible probability at least $(1/|D^*|)((m|D^*|/|\mathrm{Msg}|) + \mu)$. □ □

Theorem 3. *The KP-ABSI achieves fine-grained authorization security in the generic bilinear group model as specified in Definition 4.*

Proof. Similar to the proof for Theorem 1, firstly, we prove that our KP-ABSI scheme achieves fine-grained authorization when the challenge size $|D_0| = |D_1| = 1$ and then extend the proof to the case of challenge size $|D_0| = |D_1| > 1$.

Setup: The challenger randomly picks $a, b \xleftarrow{R} \mathbb{Z}_p$, produces $\mathrm{pk} = (e, G, G_T, g, p, g^a, g^b)$, and sends pk to $\mathcal{A}$. The challenger maintains two lists $\mathrm{List}_{H_1}(at_j, \alpha_j)$ and $\mathrm{List}_{H_2}(d, \beta)$, which are empty initially. $\mathcal{A}$ can make polynomial queries for the following $\mathcal{O}_{H_1}$ and $\mathcal{O}_{H_2}$.

$\mathcal{O}_{H_1}(at_j)$: Given an attribute at_j, if at_j was queried before, the challenger returns g^{α_j} by retrieving α_j from List_{H_1}; otherwise, the challenger picks $\alpha_j \xleftarrow{R} \mathbb{Z}_p$, records (at_j, α_j) to List_{H_1}, and returns g^{α_j} to $\mathcal{A}$.

$\mathcal{O}_{H_2}(d)$: Given a message d, if d was queried before, the challenger returns g^{β}; otherwise, the challenger picks $\beta \xleftarrow{R} \mathbb{Z}_p$, records (d, β) to List_{H_2}, and returns g^{β} to $\mathcal{A}$.

Phase 1: The challenger keeps the two lists, L_T and L_{tkn}, which are initially empty. $\mathcal{A}$ can make polynomial queries for the following oracles.

$\mathcal{O}_{\mathrm{KeyGen}}(T)$: The challenger selects $t^{(u)} \xleftarrow{R} \mathbb{Z}_p$ and runs $\{q_v(0)^{(u)} \mid v \in \mathrm{lvs}(T)\} \longleftarrow \mathrm{Share}(T, abt^{(u)})$. For each node $v \in \mathrm{lvs}(T)$, the challenger chooses $t_v^{(u)} \xleftarrow{R} \mathbb{Z}_p$ and sets

$$\mathrm{sk} = \Big(T, X_1 = g^{at^{(u)}}, X_2 = g^{bt^{(u)}}, \Big\{Y_v = g^{q_v(0)^{(u)} + \alpha_j t_v^{(u)}}, Z_v = g^{t_v^{(u)}} \mid v \in \mathrm{lvs}(T)\Big\}\Big), \tag{14}$$

where $\mathrm{att}(v) = at_j$. The challenger sends sk to $\mathcal{A}$ and records T to L_T.

$\mathcal{O}_{\mathrm{tkn}}(T)$: The challenger runs $\mathcal{O}_{\mathrm{KeyGen}}(T)$, selects $k^{(u)} \xleftarrow{R} \mathbb{Z}_p$, and sets

Table 2: tkn_1 and tkn_2 that can be obtained by $\mathscr{A}$, where $F(\text{UAtt}_1^*, T_1) = 1$ and $F(\text{UAtt}_2^*, T_2) = 1$.

$\text{tkn}_1 = \left(T_1, X_1 = g^{atk}, X_2 = g^{btk}, \left\{Y_v = g^{k(q_v(0)+\alpha_j t_v)}, Z_v = g^{kt_v} \mid v \in \text{lvs}(T_1)\right\}\right)$
$\text{tkn}_2 = \left(T_2, X_1' = g^{at'k}, X_2' = g^{bt'k}, \left\{Y_v' = g^{k(q_v'(0)+\alpha_j t_v')}, Z_v = g^{kt_v'} \mid v \in \text{lvs}(T_2)\right\}\right)$

$$\text{tkn} = \left(T, \left\{\widehat{Y}_v = Y_v^{k^{(u)}} = g^{q_v(0)^{(u)}k^{(u)}+\alpha_j t_v^{(u)}k^{(u)}}, \widehat{Z}_v = Z_v^{k^{(u)}} = g^{t_v^{(u)}k^{(u)}} \mid v \in \text{lvs}(T)\right\}, \widehat{X}_1 = X_1^{k^{(u)}} = g^{at^{(u)}k^{(u)}}, \widehat{X}_2 = X_2^{k^{(u)}} = g^{bt^{(u)}k^{(u)}}\right), \tag{15}$$

where $\text{att}(v) = at_j$. It returns tkn to $\mathscr{A}$ and adds T to L_{tkn}.

Challenge: $\mathscr{A}$ chooses two attribute sets UAtt_1^* and UAtt_2^* with the following restrictions: (1) $\forall T \in L_T$, $F(\text{UAtt}_1^*, T)$ and $F(\text{UAtt}_2^*, T)$ cannot output 1 at the same time, and, (2) $\forall T \in L_{\text{tkn}}$, $F(\text{UAtt}_1^*, T)$ and $F(\text{UAtt}_2^*, T)$ cannot output 1 at the same time. Then, $\mathscr{A}$ sends UAtt_1^* and UAtt_2^* to the challenger. The challenger chooses two sets $(D_0 = \{d_0\}, D_1 = \{d_1\})$ of equal length. For d_0, the challenger selects $r_1, r_2 \xleftarrow{R} \mathbb{Z}_p$ and computes

$$\text{cph}_1^* = \left(A_1 = g^{br_1}, A_2 = g^{a(r_1+r_2)+\beta_0}, A_3 = g^{r_2}, \left\{B_j = g^{\alpha_j r_2} \mid at_j \in \text{UAtt}_1^*\right\}\right). \tag{16}$$

The challenger randomly picks $\sigma \xleftarrow{R} \{0,1\}$ and $r_3, r_4 \xleftarrow{R} \mathbb{Z}_p$ for d_σ and sets

$$\text{cph}_2^* = \left(A_1' = g^{br_3}, A_2' = g^{a(r_3+r_4)+\beta_\sigma}, A_3' = g^{r_4}, \left\{B_j' = g^{\alpha_j r_4} \mid at_j \in \text{UAtt}_2^*\right\}\right). \tag{17}$$

Phase 2: Same as Phase 1.

Guess: Finally, $\mathscr{A}$ will eventually output a guess σ' of σ.

If $\mathscr{A}$ can determine whether d_0 is equal to d_σ or not, $\mathscr{A}$ also can determine whether g^{β_σ} is equal to g^{β_0} or not. The only way for $\mathscr{A}$ to achieve this is to construct a query $\Gamma_2(\beta_0 - \beta_\sigma)$ for some Γ_2. To prove Theorem 3, we will show that $\mathscr{A}$ can never construct a query for $\Gamma_2(\beta_0 - \beta_\sigma)$.

Table 2 shows all the possible queries of G by means of the bilinear map and group elements given to the adversary. Note that β_0, β_σ only incurs in terms $a(r_1 + r_2) + \beta_0$ and $a(r_3 + r_4) + \beta_\sigma$, respectively. Thus, $\mathscr{A}$ must construct $g^{\Gamma_2 a(r_1+r_2-r_3-r_4)}$ for obtaining $g^{\Gamma_2(\beta_0-\beta_\sigma)}$. Moreover, since (r_1, r_2) and (r_3, r_4) are independent, $\mathscr{A}$ must construct $g^{\Gamma_2 a(r_1+r_2)}$ and $g^{\Gamma_2 a(r_3+r_4)}$ for the same Γ_2. Then we show that adversary $\mathscr{A}$ can never build $g^{\Gamma_2 a(r_1+r_2)}$ and $g^{\Gamma_2 a(r_3+r_4)}$ for the same Γ_2.

To construct $\Gamma_2 a(r_1 + r_2)$ for $\mathscr{A}$, as r_1 only appears in the term br_1, we let $\Gamma_2 = \Gamma_2' b$ for some Γ_2'. That is, $\mathscr{A}$ needs to construct the term $\Gamma_2' abr_2$. In order to get that, the only way of constructing $\Gamma_2' abr_2$ is to apply tkn_1 in Table 2 with $\left\{B_j = g^{\alpha_j r_2} \mid at_j \in \text{UAtt}_1^*\right\}$ of cph_1^*, which will result in $e(g,g)^{abtkr_2}$, meaning that $\mathscr{A}$ can construct the query $abtkr_2$. That is, Γ_2 can be written as $\Gamma_2 = \Gamma_2' b = \Gamma_2'' tkb$ for a known constant Γ_2''. Similarly, we can show that Γ_2 can be written as $\Gamma_2 = \Gamma_2' b = \Gamma_2'' t' kb$ for a known constant Γ_2'' to build $\Gamma_2 a(r_3 + r_4)$. Since t and t' are unknown to $\mathscr{A}$, then Γ_2' cannot be constructed, since $\mathscr{A}$ cannot find a known constant Γ_2'' that is the product of t and t'.

In conclusion, $\mathscr{A}$ is able to construct $g^{\Gamma_2 a(r_1+r_2)}$ and $g^{\Gamma_2 a(r_3+r_4)}$ for the same Γ_2 with a negligible probability and get a negligible advantage in the fine-grained authorization game.

Similar to the proof in Theorem 1, an adversary $\mathscr{A}'$ can be simulated and it can be proved that if $\mathscr{A}$ can break the fine-grained authorization security for $|D_0| = |D_1| > 1$, then $\mathscr{A}'$ can break the fined-grained authorization security for $|D_0| = |D_1| = 1$. This completes the proof. □

4.4. Efficiency Analysis. Now we evaluate the efficiency of the schemes in terms of the asymptotic computational complexity. The asymptotic complexity is measured in terms of operations: H denotes the operation of mapping a bit-string to an element of G, E denotes the group exponentiation operation in G, E_T denotes the group exponentiation operation in G_T, and P denotes the pairing operation. We ignore the multiplication operations because they are much more efficient than the operations mentioned above (Table 3).

We can see that TokenGen incurs small cost when compared with SI. This implies that the data user should use the token to outsource the set intersection operations to the cloud.

Table 3: N is the number of attributes involved in the access tree in question, S is the number of a data user's attributes, and n is the size of sets (here we assume that both sets have the same size n).

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

5. Conclusions

In this paper, we present a novel cryptographic primitive: delegated key-policy attribute-based set intersection over outsourced encrypted data sets (KP-ABSI). It simultaneously achieves the following: (1) Each data owner outsources his/her data set in encrypted form to a cloud, where the outsourced data set is associated with an attribute set. (2) A data user is associated with an access control policy that is satisfied by the attribute sets of two encrypted data sets (owned by two data owners, respectively) and can delegate to the cloud the set intersection computation over the two data owners' outsourced encrypted data sets. (3) The cloud can conduct the set intersection operation on behalf of the data user without being able to obtain any useful information about the data owners' plaintext data set.

Thus, our scheme can solve the PSI problem in CloudIoT system. Of course, in our solution, the cloud is semihonest. How to build a construction in the malicious model is still an open problem.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by Program for the Scientific Research Foundation of Nanjing Institute of Technology (YKJ201980), Program for Natural Science Research Projects of Universities (19KJB520033), and Program for Scientific Research Foundation for Talented Scholars of Jinling Institute of Technology (JIT-B-201726).

References

[1] A. Kobusińska, C. Leung, C. H. Hsu, S. Raghavendra, and V. Chang, "Emerging trends, issues and challenges in internet of things, big data and cloud computing," *Future Generation Computer Systems*, vol. 87, pp. 416–419, 2018.

[2] Y. Yang, L. Wu, G. Yin, L. Li, and H. Zhao, "A survey on security and privacy issues in internet-of-things," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1250–1258, 2017.

[3] A. Botta, W. De Donato, V. Persico, and A. Pescapé, "Integration of cloud computing and internet of things: a survey," *Future Generation Computer Systems*, vol. 56, pp. 684–700, 2016.

[4] M. J. Freedman, K. Nissim, and B. Pinkas, "Efficient private matching and set intersection," in *Advances in Cryptology-EUROCRYPT 2004*, pp. 1–19, Springer, Berlin, Germany, 2004.

[5] Q. Wang, F. Zhou, J. Xu, and S. Peng, "Tag-based verifiable delegated set intersection over outsourced private datasets," *IEEE Transactions on Cloud Computing*, 2020.

[6] A. Abadi, S. Terzis, R. Metere, and C. Dong, "Efficient delegated private set intersection on outsourced private datasets," *IEEE Transactions on Dependable and Secure Computing*, vol. 16, no. 4, pp. 608–624, 2017.

[7] B. Pinkas, T. Schneider, and M. Zohner, "Scalable private set intersection based on ot extension," *ACM Transactions on Privacy and Security*, vol. 21, no. 2, pp. 1–35, 2018.

[8] M. Ali, J. Mohajeri, M. R. Sadeghi, and X. Liu, "Attribute-based fine-grained access control for outscored private set intersection computation," *Information Sciences*, vol. 536, 2020.

[9] B. Pinkas, M. Rosulek, N. Trieu, and A. Yanai, "PSI from PaXos: fast, malicious private set intersection," in *Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 739–767, Springer, Zagreb, Croatia, October 2020.

[10] O. Ruan and H. Mao, "Efficient private set intersection using point-value polynomial representation," *Security and Communication Networks*, vol. 2020, Article ID 8890677, 12 pages, 2020.

[11] X. Yang, X. Luo, X. A. Wang, and S. Zhang, "Improved outsourced private set intersection protocol based on polynomial interpolation," *Concurrency and Computation: Practice and Experience*, vol. 30, no. 1, p. e4329, 2018.

[12] K. Zhang, J. Chen, H. T. Lee, H. Qian, and H. Wang, "Efficient public key encryption with equality test in the standard model," *Theoretical Computer Science*, vol. 755, pp. 65–80, 2019.

[13] M. Zeng, J. Chen, K. Zhang, and H. Qian, "Public key encryption with equality test via hash proof system," *Theoretical Computer Science*, vol. 795, pp. 20–35, 2019.

[14] H. T. Lee, S. Ling, J. H. Seo, and H. Wang, "Public key encryption with equality test from generic assumptions in the random oracle model," *Information Sciences*, vol. 500, pp. 15–33, 2019.

[15] D. H. Duong, K. Fukushima, S. Kiyomoto, P. S. Roy, and W. Susilo, "A lattice-based public key encryption with equality test in standard model," in *Proceedings of the Australasian Conference on Information Security and Privacy*, pp. 138–155, Springer, Christchurch, New Zealand, July 2019.

[16] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Advances in Cryptology–EUROCRYPT 2005*, pp. 457–473, Springer, Berlin, Germany, 2005.

[17] J. Li, Q. Yu, Y. Zhang, and J. Shen, "Key-policy attribute-based encryption against continual auxiliary input leakage," *Information Sciences*, vol. 470, pp. 175–188, 2019.

[18] Y. Liu, L. Wang, X. Shen, L. Li, and D. An, "Space-efficient key-policy attribute-based encryption from lattices and two-dimensional attributes," *Security and Communication Networks*, vol. 2020, Article ID 2345369, 11 pages, 2020.

[19] J. Zhang and H. Gao, "A compact construction for non-monotonic key-policy attribute-based encryption," *International Journal of High Performance Computing and Networking*, vol. 13, no. 3, pp. 321–330, 2019.

[20] Z. Liu, S. Duan, P. Zhou, and B. Wang, "Traceable-then-revocable ciphertext-policy attribute-based encryption

scheme," *Future Generation Computer Systems*, vol. 93, pp. 903–913, 2019.

[21] Q. M. Malluhi, A. Shikfa, V. D. Tran, and V. C. Trinh, "Decentralized ciphertext-policy attribute-based encryption schemes for lightweight devices," *Computer Communications*, vol. 145, pp. 113–125, 2019.

[22] H. Ma, Z. Wang, and Z. Guan, "Efficient ciphertext-policy attribute-based online/offline encryption with user revocation," *Security and Communication Networks*, vol. 2019, Article ID 8093578, 11 pages, 2019.

[23] H. Zhu, L. Wang, H. Ahmad, and X. Niu, "Key-policy attribute-based encryption with equality test in cloud computing," *IEEE Access*, vol. 5, pp. 20428–20439, 2017.

[24] Q. Wang, L. Peng, H. Xiong, J. Sun, and Z. Qin, "Ciphertext-policy attribute-based encryption with delegated equality test in cloud computing," *IEEE Access*, vol. 6, pp. 760–771, 2017.

[25] Y. Cui, Q. Huang, J. Huang, H. Li, and G. Yang, "Ciphertext-policy attribute-based encrypted data equality test and classification," *The Computer Journal*, vol. 62, no. 8, pp. 1166–1177, 2019.