Verifiable Outsourced Ciphertext-Policy Attribute-Based Encryption for Mobile Cloud Computing

Zhiyuan Zhao[1], Jianhua Wang[1]
[1] Zhengzhou Information Science and Technology Institute
Zhengzhou 450001, Henan – P. R. China
[e-mail: 759731637@qq.com]
*Corresponding author: Zhiyuan Zhao

Received November 24, 2016; revised March 13, 2017; accepted March 30, 2017; published June 30, 2017

Abstract

With the development of wireless access technologies and the popularity of mobile intelligent terminals, cloud computing is expected to expand to mobile environments. Attribute-based encryption, widely applied in cloud computing, incurs massive computational cost during the encryption and decryption phases. The computational cost grows with the complexity of the access policy. This disadvantage becomes more serious for mobile devices because they have limited resources. To address this problem, we present an efficient verifiable outsourced scheme based on the bilinear group of prime order. The scheme is called the verifiable outsourced computation ciphertext-policy attribute-based encryption scheme (VOC-CP-ABE), and it provides a way to outsource intensive computing tasks during encryption and decryption phases to CSP without revealing the private information and leaves only marginal computation to the user. At the same time, the outsourced computation can be verified by two hash functions. Then, the formal security proofs of its (selective) CPA security and verifiability are provided. Finally, we discuss the performance of the proposed scheme with comparisons to several related works.

Keywords: Mobile cloud computing, Ciphertext-policy attribute-based encryption, Access policy, Verifiable, Outsourced computation

This research was supported by the National key Research Program of China“Collaborative Precision Positioning Project”（No.2016YFB0501900）

https://doi.org/10.3837/tiis.2017.06.024 **ISSN : 1976-7277**

1. Introduction

Cloud computing is a promising technology that is transforming the traditional Internet computing paradigm and IT industry [1]. With the rapid development of cloud computing, hundreds of thousands of enterprises have cut out their IT departments. Instead, they put their data in a cloud storage centre and customize their services to deploy their business projects online, and billions of dollars in IT expenses are saved.

For most current cloud users, once data are outsourced to a cloud service provider, they have to trust the cloud service providers (CSP), which means they have lost control of the data. This is especially concerning when all or part of the outsourced data is sensitive and should only be accessed by authorized data consumers at remote locations. Thus, it is important to encrypt the data and design a flexible access control mechanism for this encrypted data. Attribute-based encryption (ABE) is a promising public-key primitive that has been used for cryptographically enforced access control in untrusted storage. Sahai and Waters [2] first introduced the attribute-based encryption scheme. It has two variants depending on how access control is enforced: key-policy ABE (KP-ABE), where the decryption key is associated with an access control policy [3], and ciphertext-policy ABE (CP-ABE), where the ciphertext is associated with an access control policy [4]. Since the above methods were proposed, relevant personnel have carried out a large amount of research work [5-7].

With the development of wireless access technologies and the popularity of mobile intelligent terminals, cloud computing is expected to expand to mobile environments, where mobile devices and sensors are used as the information collection nodes for the cloud. Nevertheless, one of the main efficiency drawbacks of ABE is that the computational cost during the encryption and decryption phases, needing largely pairing and exponentiation computation, grows with the complexity of the access policy. This is a huge limitation on a mobile intelligent terminal (limited computational capability and low battery). To address this problem, outsourced ABE, which provides a way to outsource intensive computing tasks during encryption and decryption to CSP without revealing the private information and leaves only marginal computation to the user, has been proposed [8-10]. It has a wide range of applications. For example, when Bob went on a business trip, he had to use a mobile phone, which has limited computational capability and low battery capacity, to complete basic encryption or decryption to protect sensitive data residing in a public cloud. Outsourced ABE allows Bob to perform heavy encryption and decryption by “borrowing” the computational resources from CSP. Therefore, Bob can successfully complete the work task.

In ABE, outsourcing complex operations to a cloud server becomes an important and popular problem. Green et al. [11] proposed a scheme for this problem by introducing the notion of ABE with outsourced decryption, which largely eliminates the decryption overhead for users. However, the scheme provides no guarantee on the correctness of the transformation done by the cloud server. Furthermore, Benjamin et al. [12] addressed the problem of secure outsourcing for widely applicable linear algebraic computations. Nevertheless, the proposed protocols require expensive operations of homomorphic encryption. Li et al. [13] presented an ABE scheme that allows encryption to be securely outsourced to third-party service providers. The scheme does not consider outsourced decryption. Li et al. [14] proposed outsourced attribute-based encryption that supports both outsourced key issuing and decryption. Zhou et al. [15] presented privacy preserving cipher policy attribute-based encryption (PP-CP-ABE), which allows for secure outsourcing of both decryption and encryption to third-party service

providers. However, the root node of this access tree must be an AND gate in the scheme. Li et al. [16] proposed securely outsourced attribute-based encryption with checkability, which supports both outsourced key issuing and decryption. The authors gave an efficient method to check the correctness of the outsourced decryption in a distributed system. It requires more than one key generation service provider (KGSP), and at least one KGSP honestly takes the right ciphertext as input. Otherwise, their verification model suffers from the same attack as in Green et al.'s scheme. Armknecht et al. [17] proposed the notion of outsourced proofs of retrievability (OPOR), in which users can task an external auditor to perform and verify POR with the cloud provider. Recently, Lai et al. [18] proposed a concrete construction for ABE with verifiable decryption, which achieves both security and verifiability without random oracles. Li et al. [19] proposed a new verifiable outsourcing scheme with constant ciphertext length. They proved that their scheme is secure and verifiable in the standard model.

Contribution: In this paper, we present an efficient verifiable outsourced ABE scheme based on the bilinear group of prime order. The scheme is called the verifiable outsourced computation ciphertext-policy attribute-based encryption scheme (VOC-CP-ABE). In detail, we first construct a CP-ABE scheme with outsourced encryption and decryption based on the scheme in [15], for which the root node of the access policy must be an AND gate. However, our scheme is able to delegate encryption for any policy. Then, we use the key-encapsulated mechanism to improve OC-CP-ABE in Verifiability, that is, VOC-CP-ABE, for which the CP-ABE scheme encrypts a symmetric session key, and the message is encrypted by the symmetric session key. At the same time, we use two hash functions to obtain two hash values. To verify the integrity of the symmetric encrypted ciphertext, we use a hash value on the concatenation of the ciphertext. The second hash value is then used to verify the correctness of the outsourced decryption. Then, we provide formal security proofs of its (selective) CPA security and verifiability. This scheme is based on the prime order bilinear group. Compared with the scheme [10] based on the composite order bilinear group, our scheme has the following advantages. 1, For the CSP, a large amount of computing resources can be saved, yielding greater economic benefits. 2, The computational complexity of the two schemes is constant. However, for the limited computing resources of the mobile terminal, the scheme based on the composite order bilinear group still requires more calculation. In addition, we compare the performances of this scheme and other schemes through theoretical analysis and experimental simulation.

The rest of this paper is organized as follows. Section 2 describe some preliminaries. In Section 3, we present the system model and security definition. The proposed construction and its security analysis are presented in Section 4. In Section 5, we discuss the performance of the proposed scheme with comparisons to several related works. Finally, we conclude our work in Section 6.

2. Preliminaries

The bilinear maps, the access structure and the syntax of symmetric encryption are provided in this section.

2.1 Bilinear Maps

We present a few facts related to groups with efficiently computable bilinear maps. Assume that there is an efficient algorithm $\prod$ for generating bilinear groups. The algorithm $\prod$, by inputting a security parameter k, outputs a tuple, $\mathbb{G} = [p, G, G_T, g \in G, e]$. Here, G and G_T are

multiplicative groups of prime order p. g is a generator of G. e is a bilinear map, $e: G \times G \to G_T$. $e(g,g)$ is the generator of G_T. Z_p is the group of large prime order p. The bilinear map e has the following properties:

- Bilinearity: For all $x, y \in G$ and $a, b \in Z_p$, $e(x^a, y^b) = e(x,y)^{ab}$.
- Non-degeneracy: $e(g,g) \neq 1$, where 1 is the identity element of G_T.

We say that G is a bilinear group if the group operation in G and the bilinear map $e: G \times G \to G_T$ are both efficiently computable.

2.2 Access Structure

Let $\{P_1, P_2, \cdots, P_n\}$ be a set of parties. A collection $\mathbb{A} \subseteq 2^{\{P_1, P_2, \cdots, P_n\}}$ is monotone for $\forall B, C$ if $\forall B \in \mathbb{A}$, $B \subseteq C$, then $C \in \mathbb{A}$. An access structure (monotone access structure) is a collection (monotone collection) of nonempty subsets of $\{P_1, P_2, \cdots, P_n\}$, i.e., $\mathbb{A} \subseteq 2^{\{P_1, P_2, \cdots, P_n\}} \setminus \{\varnothing\}$. The sets in $\mathbb{A}$ are called authorized sets, and the sets not in $\mathbb{A}$ are called unauthorized sets.

2.3 Symmetric Encryption

A symmetric encryption (SE) scheme with key space $\mathcal{K}$ consists of two probabilistic polynomial time (PPT) algorithms: $SE.Enc(K, M)$, mapping a key $K \in \mathcal{K}$ and a message $M \in \mathcal{M}$ to a ciphertext C, and $SE.Dec(K, C)$, recovering M from C using K. A semantically secure one-time SE can be simply constructed from any pseudorandom generator using the one-time pad encryption scheme. The advantage $Adv_{\mathcal{A}}(\lambda)$ of an adversary $\mathcal{A}$ is as follow:

$$Adv_{\mathcal{A}}(\lambda) := \left| \Pr\left[b' = b \left(\begin{array}{c} (M_0, M_1) \leftarrow \mathcal{A}(1^{\lambda}), b \leftarrow \{0,1\}, K \leftarrow \mathcal{K} \\ C \leftarrow SE.Enc(K, M_b), b' \leftarrow \mathcal{A}(C) \end{array} \right) \right] - \frac{1}{2} \right|.$$

We say that a one-time symmetric encryption scheme is semantically secure if the advantage $Adv_{\mathcal{A}}(\lambda)$ of any PPT adversary $\mathcal{A}$ is negligible in λ.

3. System and Adversary Model

3.1 Access Policy Tree

The proposed construction is based on the CP-ABE scheme, so user secret keys are associated with a set of descriptive attributes w, while ciphertexts are associated with an encryption policy that is specified as an access tree T. Let us briefly review the concept of an access tree in [4] as well as [15] before describing our construction. Let T be a tree representing an access structure. Each non-leaf node of the tree represents a threshold gate, described by its children and a threshold value. Each leaf node of the tree is described by an attribute. A user is able to decrypt a ciphertext with a given key if and only if there is an assignment of attributes from the private key to the leaf nodes of the tree such that the tree is satisfied.

To outsource the computation of encryption and preserve the data privacy, the access tree T in [15] is split into T_{ESP} and T_{DO}, while the user is required to specify a hybrid tree connected by an AND gate. According to the above tree, we also build a hybrid access tree T as shown in **Fig. 1**. In our hybrid tree, we define a default policy, that is, $T = T_{E-CSP} \wedge \{\xi\}$. The advantage of

our construction is that, by introducing a default policy, it is able to delegate encryption for any policy. In the tree, $\wedge$ and $\vee$ denote AND and OR gates, respectively, and att_i denotes the attribute. To facilitate working with the access tree, we define a few notations and functions as follows.

- num_x is the number of children of a node x, and k_x is its threshold value; then, $0 < k_x < num_x$. When $k_x = 1$, the threshold gate is an OR gate, and when $k_x = num_x$, it is an AND gate. Each leaf node x of the tree is described by an attribute and a threshold value $k_x = 1$.

- The access tree T also defines an ordering between the children of every node; that is, the children of a node are numbered from 1 to num. The function $\mathcal{A}$ returns such a number associated with node x. The function L returns the parent of node x in the tree. $Att(x)$ returns the attribute associated with leaf node x.

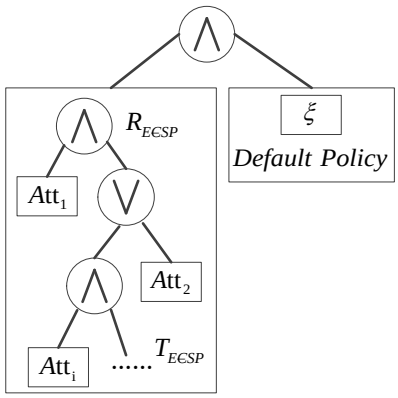


Fig. 1. Hybrid Access Policy Tree

3.2 System Model

The notation used in this paper is listed in **Table 1**. Let $x \| y$ denote the concatenation of two strings x and y.

Table 1. Notation used in this paper

Acronym	Descriptions
DO	Data Owner
DU	Data User
T	Access Policy Tree
E-CSP	Encryption-Cloud Service Provider
D-CSP	Decryption-Cloud Service Provider
S-CSP	Storage-Cloud Service Provider
TA	Trust Authority

This proposed system mainly includes DO, DU, E-CSP, D-CSP, S-CSP and TA, as shown in **Fig. 2**. In our proposed scheme, DO and DU can be a mobile wireless device or a sensor that can store/request information in/from the Cloud. To guarantee that our proposed VOC-CP-ABE scheme is secure, the presented system model has the following properties: (1) the data must be encrypted before sending to S-CSP; (2) E-CSP provides encryption service to the data owner without knowing the actual data encryption key; (3) D-CSP provides decryption service to users without knowing the data content; and (4) even though E-CSP, D-CSP and S-CSP collude, the data content cannot be revealed.

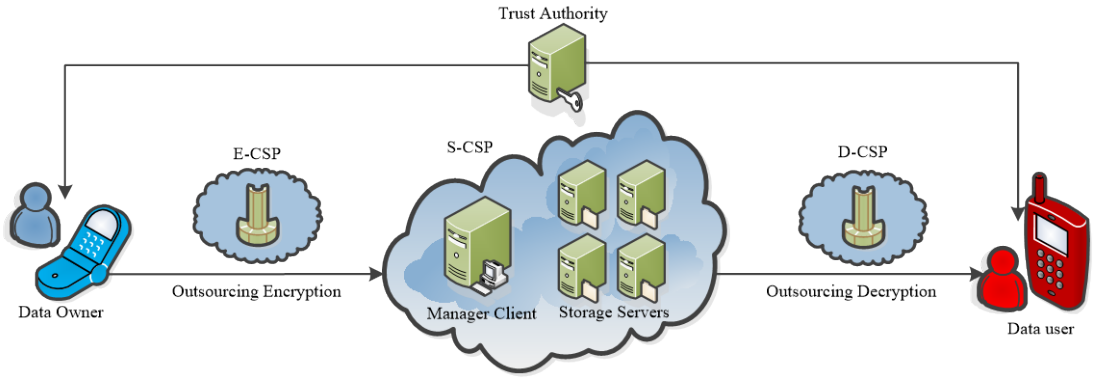


Fig. 2. Architecture for outsourced attribute-based encryption

As shown in **Fig. 2**, S-CSP, E-CSP and D-CSP are the core components of the proposed system. E-CSP and D-CSP provide outsourcing encryption and decryption computing services, and S-CSP provides storage services. The cloud is semi-trusted, as the cloud only provides computing and storage services with assistance for data security; however, the data are blinded to the cloud.

First, we give an overview of the VOC-CP-ABE (verifiable outsourced computation ciphertext-policy attribute-based encryption) scheme as follows:

$\boldsymbol{Setup(1^{\lambda})}$: The setup algorithm takes as input a security parameter λ and outputs a public key PK and a master secret key MSK.

$\boldsymbol{KeyGen(MSK,S)}$: The key generation algorithm run by TA takes as input the master secret key of TA and the set of attributes *S* for the user and then generates the secret key SK.

$\boldsymbol{Encrypt(PK,M,T)}$: The encryption algorithm requires communication between DO and E-CSP and contains the $Encrypt_{DO}$ and $Encrypt_{E-CSP}$ algorithms. It takes as input PK, the message $M \in \mathcal{M}$ and the access policy tree $T = T_{E-CSP} \wedge \{\xi\}$ and outputs a ciphertext CT and the verification mark VM.

$\boldsymbol{KeyBlind(SK)}$: The key blind algorithm is run by DU. It takes as input the secret key SK and then generates the blinded secret key SK' and the unique retrieval key RK.

$\boldsymbol{Decrypt_{D-CSP}(SK',CT')}$: The $Decrypt_{D-CSP}$ algorithm takes as input the blinded secret key SK' and partially ciphertext $CT' \subset CT$. It outputs intermediate decrypted ciphertext IDC.

$\boldsymbol{Decrypt_{DU}(IDC,RK,CT'',VM)}$: The $Decrypt_{DU}$ algorithm takes as input the intermediate decrypted ciphertext IDC, the unique retrieval key RK, the partially ciphertext $CT'' \subset CT$ and the verification mark VM. It outputs the message M, or it returns $\perp$ and halts immediately.

3.3 Adversary Model

We assume that the symmetric encryption algorithm and one-way hash function used in this paper are secure and that the Discrete Logarithm Problem (DL) on both groups G and G_T is hard. In addition, TA is responsible for distributing cryptographic keys, and it is well guarded and trustable. We consider the cloud service providers to be honest but curious [20]. More specifically, they follow the protocol but try to determine as much private information as possible. The channels between users and E-CSP/D-CSP/S-CSP are secure. We thus consider

the adversary: Adversary $\mathcal{A}$ refers to some corrupted users colluding with E-CSP, D-CSP and S-CSP, who can obtain private keys of corrupted users, the transformed ciphertexts CT_{E-CSP} stored at E-CSP, the blinded private keys SK' stored at D-CSP, and the ciphertexts stored at S-CSP. It intends to decrypt uncorrupted users' ciphertexts at S-CSP.

RCCA Security Game for the Adversary. We describe the RCCA security game for adversary $\mathcal{A}$ based on the replayable chosen-ciphertext attack (RCCA) security in [11].

Setup: The challenger $\mathcal{C}$ runs the *Setup* algorithm and gives the public parameters *PK* to the adversary $\mathcal{A}$.

Phase 1: $\mathcal{C}$ initializes an empty table T, an empty set D and an integer $j=0$. Proceeding adaptively, the adversary can repeatedly make any of the following queries:

- ***Creat(S)***: $\mathcal{C}$ sets $j:=j+1$. It runs *KenGen* with S to obtain SK, runs *KeyBlind* with SK to obtain SK' and RK, runs $Encrypt_{E-CSP}$ to obtain CT_{E-CSP}, and then stores in table T the entry $(j,S,SK,SK',RK,CT_{E-CSP})$.

Note: Create can be repeatedly queried with the same input.

- ***Corrupt.SK(i)***: $\mathcal{C}$ checks whether the i^{th} entry (i,S,SK) exists in table T. If $f(T^*,S)=1$, that is, the attribute set S does satisfy the access policy tree T^*, it returns $\perp$. Otherwise, it sets $D:=D\bigcup\{S\}$ and returns SK. If no such entry exists, it returns $\perp$.

- ***Corrupt.SK'(i)***: $\mathcal{C}$ checks whether the i^{th} entry (i,S,SK') exists in table T. If so, return SK'; otherwise, return $\perp$.

- ***Corrupt.$CT_{E\text{-}CSP}$(i)***: $\mathcal{C}$ checks whether the i^{th} entry (i,S,CT_{E-CSP}) exists in table T. If so, return CT_{E-CSP}; otherwise, return $\perp$.

- ***Decrypt(i,CT,VM)***: $\mathcal{C}$ checks whether the i^{th} entry (i,S,SK,SK',RK) exists in table T. If so, return the output of the decryption on CT; otherwise, return $\perp$.

Challenge: The adversary $\mathcal{A}$ submits two equal length messages M_0 and M_1. In addition, $\mathcal{A}$ gives a value T^* such that for all $S\in D$, $f(T^*,S)\neq 1$. That is, the attribute set S does not satisfy the access policy tree T^*. The challenger $\mathcal{C}$ flips a random coin b and runs *Encrypt*, which contains the $Encrypt_{DO}$ and $Encrypt_{E-CSP}$ algorithms, to encrypt M_b under T^*. The resulting ciphertexts (CT^*_{E-CSP},CT^*,VK^*) are given to $\mathcal{A}$.

Phase 2: Phase 1 is repeated with the following restrictions:

- $\mathcal{A}$ cannot issue a Corrupt query that would result in a value S that satisfies $f(T^*,S)=1$ being added to D.
- If a decryption response would be either M_0 or M_1, then the challenger $\mathcal{C}$ responds with the special message $\perp$.

Guess: The adversary $\mathcal{C}$ outputs a guess b' of b.

The advantage of $\mathcal{A}$ is defined as $Adv^{RCCA}_{\mathcal{A}}(\lambda):=\left|\Pr[b=b']-1/2\right|$.

Definition 1. *The VOC-CP-ABE scheme is RCCA secure if all probabilistic polynomial-time adversaries have at most a negligible advantage in the security game defined above. That is,* $Adv^{RCCA}_{\mathcal{A}}(\lambda)$ *is negligible in* λ.

CPA Security. We say that a VOC-CP-ABE scheme is CPA secure (or secure against chosen-plaintext attacks) if we remove the Decrypt oracle in both Phases 1 and 2.

Selective Security. We say that a VOC-CP-ABE scheme is selectively secure if we add an Init stage before Setup where the adversary commits to the challenge access structure T^* at the beginning.

Verifiability of Outsourced Computation. Verifiability guarantees that a user can efficiently check if the transformation is done correctly. It is described between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$.

Setup: The challenger $\mathcal{C}$ runs the *Setup* algorithm and gives the public parameters *PK* to the adversary $\mathcal{A}$.

Phase 1: $\mathcal{C}$ can adaptively query the Oracles as in ***Phase 1*** above.

Challenge: The adversary $\mathcal{A}$ submits a message M^* and a value T^*. The challenger $\mathcal{C}$ computes a challenge ciphertext $Encrypt(PK, M^*, T^*) = (CT^*, VM^*)$ and sends it to $\mathcal{A}$.

Phase 2: The same as Phase 1.

Guess: The adversary $\mathcal{C}$ outputs a value S^* such that $f(T^*, S^*) = 1$ and an intermediate decrypted ciphertext IDC.

We say that $\mathcal{A}$ succeeds in the game if $Decrypt(IDC, RK, VM^*, CT^*) \notin \{M^*, \perp\}$. The advantage of $\mathcal{A}$ is defined as $Adv_{\mathcal{A}}^{Ver}(\lambda) := \Pr[\mathcal{A}\, Wins]$.

Definition 2 (Verifiability): *The VOC-CP-ABE scheme with outsourced decryption is (privately) verifiable if for all PPT adversary* $\mathcal{A}$, *the advantage* $Adv_{\mathcal{A}}^{Ver}(\lambda) := \Pr[\mathcal{A}\, Wins]$ *is negligible in* λ.

4. Proposed CP-ABE Scheme with Verifiable Outsourced Computation

In this section, we first construct a CP-ABE scheme with outsourced encryption and decryption based on the scheme in [15]. It is able to delegate encryption for any policy in our scheme. Then, we use the key-encapsulated mechanism to improve the OC-CP-ABE (outsourced computation ciphertext-policy attribute based encryption) scheme in verifiability. Finally, we provide a formal security proof of its (selective) CPA security and verifiability.

4.1 Proposed CP-ABE Scheme with Outsourced Computation

The proposed construction contains five polynomial-time algorithms: *Setup*, *KenGen*, $Encrypt(Encrypt_{E-CSP}, encrypt_{DO})$, *KeyBlind*, and $Decrypt(Decrypt_{D-CSP}, Decrypt_{DU})$. These algorithms are presented in detail in the following.

Setup(1^{λ}): The setup algorithm is executed by the authority. Select a bilinear group G of prime order q with generator g and two random integers $\alpha, \beta \in Z_q$, and compute $h = g^{\beta}$. Define a hash function $H: \{0,1\} \to G$ modelled as a random oracle. Finally, output the public key $PK = \langle G, g, h, e(g,g)^{\alpha}, H \rangle$ and the master key $MSK = \langle \beta, g^{\alpha} \rangle$, which is only known by the authority.

***KeyGen*(*MSK*, *S*)**: For each user's private key request, the authority runs the key generation algorithm. Choose $r \in Z_p$ and $r_j \in Z_p$ for each attribute $j \in S \bigcup \{Att(\xi)\}$. Then, $D = g^{\frac{(\alpha+r)}{\beta}}$ is calculated. For the attribute $j \in S \bigcup \{Att(\xi)\}$, calculate $D_{j0} = g^r H(j)^{r_j}$ and $D_{j1} = g^{r_j}$. The private key is $SK = \left\langle D, \{D_{j0}, D_{j1}\}_{j \in S \bigcup \{Att(\xi)\}} \right\rangle$.

***Encrypt*(*PK*, *M*, *T*)**: The encryption algorithm requires communication between DO and E-CSP and contains the $Encrypt_{DO}$ and $Encrypt_{E-CSP}$ algorithms. In the access policy tree $T = T_{E-CSP} \wedge \{\xi\}$, a default attribute ξ is appended to each user's attribute set.

DO first picks an integer $s \in Z_p$ and randomly specifies a 1-degree polynomial $q_R(x)$ such that $q_R(0) = s$. Furthermore, let $s_1 = q_R(1)$ and $s_2 = q_R(2)$. Then, DO sends $\{s_1, T_{E-CSP}\}$ to E-CSP.

E-CSP then runs $Encrypt_{E-CSP}(s_1, T_{E-CSP})$. For $\forall x \in T_{E-CSP}$, randomly choose a $(k_x - 1)$-degree polynomial $q(x)$ in a top-down manner. We note that the polynomial $q(x)$ is chosen with the restriction that $q(0) = s_1$ if x is the root node in T_{E-CSP}; otherwise, $q(0) = q_{parent(x)}(index(x))$. For $\forall y \in Y_{E-CSP}$, compute $C_{y0} = g^{q_y(0)}$ and $C_{y1} = H(Att(y))^{q_y(0)}$. The transformed ciphertext is $CT_{E-CSP} = \left\langle \{C_{y0}, C_{y1}\}_{y \in Y_{E-CSP}} \right\rangle$, where Y_{E-CSP} is the set of leaf nodes in T_{E-CSP}.

At the same time, DO runs $Encrypt_{DO}(s_2, \xi)$. Compute $C_{\xi 0} = g^{s_2}$ and $C_{\xi 1} = H(Att(\xi))^{s_2}$ for ξ. Then, $C = Me(g,g)^{\alpha s}$ and $C' = h^s$ are calculated. DO sends $\{C, C', C_{\xi 0}, C_{\xi 1}\}$ to E-CSP.

After receiving the message $\{C, C', C_{\xi 0}, C_{\xi 1}\}$ from DO, E-CSP generates the following ciphertext: $CT = \left\langle T, C, C' = h^s, C_{\xi 0}, C_{\xi 1}, \{C_{y0}, C_{y1}\}_{y \in Y_{E-CSP}} \right\rangle$.

***KeyBlind*(*SK*)**: DU chooses a random $\delta \in Z_p$ to blind his private key and then calculates $D' = D^{\delta} = g^{\frac{\delta(\alpha+r)}{\beta}}$. DU holds the unique retrieval key $RK = \delta$. The blinded private key is $SK' = \left\langle D', \{D_{j0}, D_{j1}\}_{j \in S \bigcup \{Att(\xi)\}} \right\rangle$.

***Decrypt*$_{D\text{-}CSP}$(*SK*′,*CT*′)**: Suppose that the private key's attribute set $S \bigcup \{Att(\xi)\}$ satisfies the hybrid tree T inserted into the ciphertext CT. Then, DU sends SK' to D-CSP and requests S-CSP to send the ciphertext to D-CSP. After receiving the request, S-CSP sends $CT' = \left\langle T, C', C_{\xi 0}, C_{\xi 1}, \{C_{y0}, C_{y1}\}_{y \in Y_{E-CSP}} \right\rangle$ to D-CSP, where $CT' \subset CT$. We define a recursive algorithm $DecryptNode(CT, SK, x)$, which is the same as that defined in [4], where $i = Att(x)$.

For ξ:

$$DecryptNode(CT', SK', \xi) = \frac{e(C_{\xi 0}, D_{\xi 0})}{e(C_{\xi 1}, D_{\xi 1})} = \frac{e(g^{s_2}, g^r H(Att(\xi))^{r_\xi})}{e(H(Att(\xi))^{s_2}, g^{r_\xi})} = e(g,g)^{rs_2}$$

For $\forall y \in Y_{E-CSP}$:

$$DecryptNode(CT', SK', y) = \frac{\prod e(C_{y0}, D_{j0})}{\prod e(C_{y1}, D_{j1})} = \frac{e(g^{q_y(0)}, g^r H(Att(\xi))^{r_j})}{e(H(Att(\xi))^{q_y(0)}, g^{r_j})} = e(g,g)^{rq_y(0)} = F_y$$

In the tree T_{E-CSP}, we now consider the recursive case when x is a non-leaf node. For all nodes y that are children of x, the algorithm calls $DecryptNode(CT, SK, y)$ and stores the output as F_y. Let S_x be an arbitrary k_x-sized set of child nodes y. D-CSP calculates:

$$F_x = \prod_{y \in S_x} F_y^{\Delta_{i,S'_x}(0)} = \prod_{y \in S_x} (e(g,g)^{r \cdot q_y(0)})^{\Delta_{i,S'_x}(0)} = \prod_{y \in S_x} (e(g,g)^{r \cdot q_{parent(y)}(index(y))})^{\Delta_{i,S'_x}(0)}$$
$$= \prod_{y \in S_x} (e(g,g)^{r \cdot q_x(i)})^{\Delta_{i,S'_x}(0)} = e(g,g)^{r \cdot q_x(0)}$$

where $i = index(y)$ and $S'_x = \{index(y) : y \in S_x\}$. Thus, we can obtain $e(g,g)^{rs_2}$ and $e(g,g)^{rq_{R_{E-CSP}}(0)} = e(g,g)^{rs_1}$ and calculate $\varphi = e(g,g)^{rs}$. Then, $\psi = e(C', D') = e(h^s, g^{\frac{\delta(\alpha+r)}{\beta}}) = e(g,g)^{\delta s(\alpha+r)}$ is calculated. Finally, D-CSP sends the intermediate decrypted ciphertext $IDC = \{\varphi, \psi\}$.

Decrypt*$_{DU}$*(IDC, RK, C): After receiving IDC, DU calculates $\psi' = \psi^{1/\delta} = e(g,g)^{s(\alpha+r)}$ and computes the message: $M = \frac{\varphi C}{\psi'} = \frac{e(g,g)^{sr} Me(g,g)^{s\alpha}}{e(g,g)^{s(\alpha+r)}}$.

4.2 Improving the OC-CP-ABE Scheme in Verifiability

Next, we formally describe our verifiable outsourced CP-ABE scheme. Our generic construction uses the following components.

- The outsourced computation CP-ABE system: OC-CP-ABE $= (Setup^O, KenGen^O, Encrypt^O, KeyBlind^O, Decrypt^O)$ with message space $\mathcal{M}$.
- Two collision-resistant hash functions: $H_0 : \mathcal{M} \to \{0,1\}^{l_{H_0}}$ and $H_1 : \mathcal{M} \to \{0,1\}^{l_{H1}}$.
- A symmetric encryption (SE) scheme $SE = (SE.Enc, SE.Dec)$ with key space $\{0,1\}^{l_{SE}}$.
- A family of pairwise independent hash functions $\mathcal{H}$ from $\mathcal{M}$ to $\{0,1\}^{l_{SE}}$.

The above parameters satisfy the following condition: $0 < l_{SE} \le (\log |k| - l_{H_0}) - 2\log(1/\varepsilon_H)$, where ε_H is a negligible value in k.

This VOC-CP-ABE scheme is presented in detail in the following:

***Setup*(1^λ)**: The setup algorithm is executed by the authority. It runs $Setup^O(1^\lambda)$ to obtain (MSK^O, PK^O). Then, choose an extractor[1] $h \in \mathcal{H}$, two hash functions, H_0 and H_1, and a symmetric encryption scheme SE. Finally, output the public key $PK = \langle PK^O, H_0, H_1, h, SE \rangle$ and the master key $MSK = MSK^O$, which is only known by the authority.

***KeyGen*(MSK, S)**: The authority runs the key generation algorithm. It is the same as $KeyGen^O$. The private key is $SK = SK^O$.

***Encrypt*(PK, M, T)**: The encryption algorithm requires communication between DO and E-CSP. DO chooses a random message $R \in G_T$ and runs $Encrypt^O(PK^O, R, T)$ to obtain a ciphertext CT^O. At the same time, DO sets $Mark' = H_0(R)$ and computes a symmetric key

[1] A randomness extractor, often simply called an "extractor," is a function that, when applied to the output from a weakly random entropy source, together with a short, uniformly random seed, generates a highly random output that appears to be independent of the source and uniformly distributed [21].

$K_{SE} = h(R)$. Next, calculate $C_{SE} = SE.Enc(K_{SE}, M)$ and $Mark = H_1(Mark' \| C_{SE})$. Finally, DO sends C_{SE} and $Mark$ to E-CSP. After that, E-CSP generates the following ciphertext $CT = \{CT^O, C_{SE}\}$ and the verification mark $VM = Mark$.

***KeyBlind*(*SK*)**: It is the same as $KeyBlind^O$. DU holds the unique retrieval key $RK = RK^O$. The blinded private key is $SK' = SK'^O$.

***Decrypt*$_{D\text{-}CSP}$(*SK'*,*CT'*)**: Suppose that the private key's attribute set $S \cup \{Att(\xi)\}$ satisfies the hybrid tree T inserted into the ciphertext CT^O. Then, DU sends SK' to D-CSP and requests S-CSP to send the ciphertext to D-CSP. After receiving the request, S-CSP sends $CT' = CT'^O$ to D-CSP, where $CT' \subset CT^O \subset CT$. It runs $Decrypt^O_{D-CSP}(CT', SK')$ to obtain $\varphi = \varphi^O$ and $\psi = \psi^O$. Finally, D-CSP sends the intermediate decrypted ciphertext $IDC = \{\varphi, \psi\}$ to DU.

***Decrypt*$_{DU}$(*IDC*, *RK*, *CT''*, *VM*)**: S-CSP sends $CT'' = \langle C, C_{SE} \rangle$ to DU, where $CT'' \subset CT$. It runs $Decrypt^O_{DU}(IDC, RK, C)$ to obtain the random message R. Then, it calculates $Mark' = H_0(R)$. If $H(Mark' \| C_{SE}) \neq VM$, it returns $\perp$ and halts immediately. Otherwise, it calculates $K_{SE} = h(R)$ and returns $M = SE.Dec(K_{SE}, C_{SE})$.

We depict the framework of our scheme as shown in **Fig. 3**.

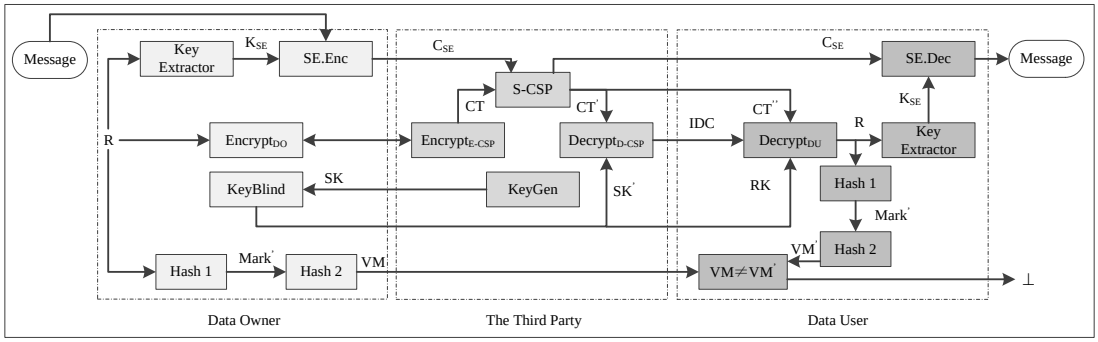


Fig. 3. The Framework of Verifiable Outsourced CP-ABE

4.3 Security Analysis

Theorem 1. *The OC-CP-ABE scheme provided in Section 4.1 is selectively CPA secure. The proof is similar to that in* [15], *and due to the space limits, the proof is omitted.*

Theorem 2: *Suppose that the OC-CP-ABE scheme is (selectively) CPA secure,* $\mathcal{H}$ *is a family of pairwise independent hash functions,* SE *is a semantically secure one-time symmetric encryption scheme, and the parameters satisfy* $0 < l_{SE} < (\log|k| - l_{H_0}) - 2\log(1/\varepsilon_H)$. *Then, the VOC-CP-ABE is (selectively) CPA secure.*

Proof: First, we consider the following three games:

- **Game0**: The original CPA-secure game. Let $(CT^*, VM^*) = ((CT^{O*}, C^*_{SE}), Mark^*)$ denote the challenge ciphertext and verification mark for a challenge value T^* selected by the adversary $\mathcal{A}$. We also denote by $R^* \in \mathcal{M}$ the key encrypted in ciphertext CT^{O*} and by $K^*_{SE} = h^*(R^*)$ the symmetric key used in ciphertext C^*_{SE}.

- **Game1**: In *Game0*, we run $Encrypt^O(PK^{O*}, R^*, T^*)$ to obtain the ciphertext CT^{O*}, which is the ciphertext of $R^* \in \mathcal{M}$. Then, we set $Mark'^* = H^*_0(R^*)$ and compute the symmetric key

$K_{SE}^{*} = h^{*}(R^{*})$. In *Game1*, CT^{O*} is still the ciphertext of $R^{*} \in \mathcal{M}$. However, we compute $Mark'^{*} = H_0^{*}(K^{*})$ and $K_{SE}^{*} = h^{*}(K^{*})$ using another random key $K^{*} \in \mathcal{M}$, which is independent of R^{*}. In addition, the two games are exactly the same in other respects.

- **Game2**: Same as *Game1* except that K_{SE}^{*} is replaced by a random string $Ra_{SE}^{*} \in \{0,1\}^{l_{SE}}$.

Then, we can prove the indistinguishability between the pairs (*Game0* and *Game1*) and (*Game1* and *Game2*). Finally, we can prove that the advantage of adversary $\mathcal{A}$ in *Game2* is negligible. Thus, the advantage of adversary $\mathcal{A}$ in the real game is negligible. We prove this theorem via the following lemma:

Lemma 1: *Suppose that the OC-CP-ABE scheme is (selectively) CPA secure; then, the adversary's views in Game0 and Game1 are computationally indistinguishable.*

Proof of Lemma 1: Suppose that there exits an adversary $\mathcal{A}$ who has a non-negligible difference ε between its advantage in *Game0* and its advantage in *Game1*. We can build a PPT simulator $\mathcal{B}$ to break the CPA security of the OC-CP-ABE scheme with a non-negligible advantage ε. $\mathcal{C}$ is the challenger of the OC-CP-ABE scheme. $\mathcal{B}$ simulates $\mathcal{A}$'s views in *Game0* or in *Game1* depending on its challenge ciphertext. The simulator $\mathcal{B}$ plays the role of the adversary in the OC-CP-ABE scheme and interacts with $\mathcal{A}$ as follows:

Init: $\mathcal{B}$ gives the challenge access policy T^{*} from $\mathcal{A}$ to the challenger $\mathcal{C}$.

Setup: $\mathcal{B}$ first runs $\mathcal{C}$ to obtain a challenge public parameter PK^{*} and chooses two collision-resistant hash functions H_0^{*} and H_1^{*}, a random extractor $h^{*} \in \mathcal{H}$ and a semantically secure one-time encryption scheme SE^{*}. Finally, it sends $(PK^{*}, H_0^{*}, H_1^{*}, h^{*}, SE^{*})$ to $\mathcal{A}$ as a challenge public key PK.

Phase 1: $\mathcal{B}$ forwards any of $\mathcal{A}$'s queries, including *Creat*(*S*), *Corrupt.SK*(*i*), *Corrupt.SK'*(*i*) and *Corrupt.CT*(*i*), to $\mathcal{C}$ and returns the replies to $\mathcal{A}$.

Challenge: $\mathcal{A}$ submits two equal-length messages M_0 and M_1, as well as a value T^{*}; $\mathcal{B}$ first chooses two independent random keys $R^{*}, K^{*} \in \mathcal{M}$. It then queries $\mathcal{C}$ with $((R^{*}, K^{*}), T^{*})$. $\mathcal{C}$ will return a challenge ciphertext CT^{O*} to $\mathcal{B}$. Next, $\mathcal{B}$ sets $K_{SE}^{*} = h^{*}(K^{*})$ and $Mark'^{*} = H_0^{*}(K^{*})$. It also computes $C_{SE}^{*} = SE^{*}.Enc(K_{SE}^{*}, M_b)$ for a random $b \in \{0,1\}$ and sets $Mark^{*} = H_1^{*}(Mark'^{*} \| C_{SE}^{*})$. Finally, it sends $CT^{*} = \{CT^{O*}, C_{SE}^{*}\}$ and $VM^{*} = Mark^{*}$ to $\mathcal{A}$. CT^{*} is a challenger ciphertext, as in *Game0*, if CT^{O*} is the ciphertext of K^{*}. CT^{*} is a challenger ciphertext, as in *Game1*, if CT^{O*} is the ciphertext of R^{*}.

Phase 2: $\mathcal{B}$ proceeds as in Phase 1.

Guess: $\mathcal{A}$ outputs its guess $b' \in \{0,1\}$. $\mathcal{B}$ outputs 0 if $b' = b$ or outputs 1 if $b' \neq b$.

From the above analysis, $\mathcal{B}$ perfectly simulates $\mathcal{A}$'s views in *Game0* or *Game1*. By our assumption, the probability that $\mathcal{A}$ guesses b correctly in *Game0* has a non-negligible ε difference from that of it guessing b correctly in *Game1*. When CT^{O*} is the ciphertext of K^{*}, $\mathcal{A}$ is in *Game0*, and when CT^{O*} is the ciphertext of R^{*}, $\mathcal{A}$ is in *Game1*. Therefore, $\mathcal{B}$ has advantage ε in the OC-CP-ABE scheme.

Lemma 2: Suppose that $\mathcal{H}$ is a family of pairwise independent hash functions; then, the adversary's views in *Game1* and *Game2* are statistically indistinguishable.

Proof of Lemma 2: In both *Game1* and *Game2*, K^* is completely independent of CT^{O*}, h^* and PK^*. In addition, $Mark'^* = H_0^*(K^*)$ has at most $2^{l_{H_0}}$ possible values. Hence, we have $\tilde{H}_\infty(K^* \mid (PK, CT^{O*}, h^*, Mark'^*)) \geq \tilde{H}_\infty(K^* \mid (PK, CT^{O*}, h^*)) - l_{H_0} = \log|\mathcal{M}| - l_{H_0}$ [2].

Because $0 < l_{SE} \leq (\log|\mathcal{M}| - l_{H_0}) - 2\log(1/\varepsilon_H)$, from the adversary's point of view (except the variable C_{SE}^*), the symmetric key $K_{SE}^* = h^*(K^*)$ of *Game1* is ε_H-statistically indistinguishable from a truly random symmetric key $Ra_{SE}^* \in \{0,1\}^{l_{SE}}$ of *Game2*. That is, K_{SE}^* and Ra_{SE}^* are not distinguishable. Observe that C_{SE}^* is a function of K_{SE}^* and that $Mark^*$ is a function of $Mark'^*$ and C_{SE}^*. Thus, they do not increase the distance between the above two distributions. That is, a series of changes in *Game1* and *Game2* do not alter the distance between the above two distributions. Thus, the statistical distinguishability between *Game1* and *Game2* is the same as that between K_{SE}^* and Ra_{SE}^*. Thus, *Game1* and *Game2* are indistinguishable in the adversary's view.

Lemma 3: Suppose that the symmetric encryption scheme SE^* is semantically secure; then, the adversary $\mathcal{A}$ in *Game2* has a negligible advantage.

Proof of Lemma 3: In *Game2*, the symmetric key Ra_{SE}^* is a random string. Thus, adversary $\mathcal{A}$ has no information of the random value. Thus, we can directly construct a simulator $\mathcal{B}$ from $\mathcal{A}$ to attack the semantic security of SE^*. We can obtain $|\Pr[\text{Game2}] - 1/2| \leq Adv_{\mathcal{B}}^{SE^*}(\lambda)$, where $\mathcal{B}$ is an adversary attacking the semantic security of SE^*. Because the symmetric encryption scheme SE^* is semantically secure, adversary $\mathcal{A}$ in *Game2* has a negligible advantage.

Taking lemmas 1, 2 and 3, the VOC-CP-ABE scheme is (selectively) CPA secure.

Theorem 3: Suppose that H_0 and H_1 are collision-resistant hash functions. Then, the VOC-CP-ABE scheme is privately verifiable.

Proof: Given an adversary $\mathcal{A}$ against the verifiability, we construct an efficient algorithm $\mathcal{B}$ to break the collision resistance of the underlying hash function H_0 or H_1. Given two challenge hash functions (H_0^*, H_1^*), $\mathcal{B}$ simulates the experiment as follows.

Setup: $\mathcal{B}$ runs the *Setup* algorithm to obtain the public parameter PK and master secret key MSK, except for the hash functions H_0^* and H_1^*. Note that $\mathcal{B}$ knows the MSK.

Phase 1: $\mathcal{B}$ simulates $\mathcal{A}$'s queries in Phase 1 above.

Challenge: $\mathcal{A}$ submits a challenge message M^* and a value T^*, and $\mathcal{B}$ first invokes $Encrypt^O(PK, R^*, T^*)$ to obtain a ciphertext CT^{O*} of a random key $R^* \in \mathcal{M}$. It then sets $Mark'^* = H_0^*(R^*)$ and $K_{SE}^* = h^*(R^*)$. $\mathcal{B}$ also computes $C_{SE}^* = SE.Enc(K_{SE}^*, M^*)$ and $Mark^* = H_1^*(Mark'^* \| C_{SE}^*)$. After that, it sends $CT^* = \{CT^{O*}, C_{SE}^*\}$ and $VM^* = Mark^*$ to $\mathcal{A}$. $\mathcal{B}$ holds VM^* and (R^*, C_{SE}^*).

Phase 2: $\mathcal{B}$ simulates $\mathcal{A}$'s queries in Phase 2 above.

Guess: $\mathcal{A}$ outputs a value S^* (such that $f(T^*, S^*) = 1$), an intermediate decrypted ciphertext $IDC = \{\varphi, \psi\}$ and C_{SE}.

---

[2] Let X, Y and Z be random variables. If Y has at most 2^r possible values, then $\tilde{H}_\infty(X \mid (Y, Z)) \geq \tilde{H}_\infty(X \mid Z) - r$ [22].

If $\mathcal{A}$ breaks the verifiability, $\mathcal{B}$ will recover a message $M \notin \{M^*, \perp\}$ via $Decrypt_{DU}(IDC, RK, CT'', VM^*)$. We now discuss $\mathcal{A}$'s success probability. Observe that the decryption algorithm outputs $\perp$ if $H(Mark' \| C_{SE}) \neq Mark^*$, where $Mark' = H_0^*(R)$ and $R = Decrypt'_{DU}(IDC, RK, CT'')$. Thus, we only need to consider the following two cases:

Case 1: $(Mark', C_{SE}) \neq (Mark'^*, C_{SE}^*)$. Because $\mathcal{B}$ knows $(Mark'^*, C_{SE}^*)$, if this case occurs, $\mathcal{B}$ immediately obtains a collision of the hash function H_1^*.

Case 2: $(Mark', C_{SE}) = (Mark'^*, C_{SE}^*)$, but $R \neq R^*$. Observe that $H_0^*(R) = Mark' = Mark'^* = H_0^*(R^*)$. Thus, it breaks the collision-resistance of H_0^*.

Through the above two cases, the proof of Theorem 3 is completed.

5. Performance

In this section, we give both an efficiency analysis and an experimental comparison of the VOC-CP-ABE scheme.

5.1 Efficiency Analysis

To evaluate the performance of the presented VOC-CP-ABE scheme, we evaluate the computation overhead of service providers and users based on theoretical analysis. We compare the performance of our scheme with those of other outsourced ABE schemes in [10,11,16,18,19]. The reasons for choosing these five schemes are that references [11,18] belong to the classical class of the outsourced scheme and references [10,16,19] represent the recent research status of the outsourced scheme. In the course of comparison, s, l and y indicate the set that satisfies the decryption requirements, the number of rows of the matrix $\boldsymbol{M}$ for LSSS (or the amounts of the attributes related to ciphertext) and the number of leaf nodes, respectively. E_G and E_{G_T} denote modular exponentiation computations in G and G_T, respectively. E_{G_1} denotes a modular exponentiation computation in G_1 (in reference [10], G_1 is a subgroup of G; G in reference [10] is different from G in other studies). P denotes a paring computation. We first give a comparison based on the theoretical aspects in **Table 2**.

Table 2. Efficiency Comparison

Schemes	Group order	Encryption		Decryption		Verify
		E-CSP	DO	D-CSP	DU	
Green[11]	Prime	None	$(3l+1)E_G+1E_{G_T}$	$2\|s\|E_{G_T}+(\|s\|+2)P$	$1E_{G_T}$	None
Lai[18]	Prime	None	$(6l+4)E_G+2E_{G_T}$	$2\|s\|E_{G_T}+(4\|s\|+2)P$	$2E_G+2E_{G_T}$	Has
Li[16]	Prime	None	$(l+2)E_G+1E_{G_T}+1P$	$2\|s\|E_{G_T}+(2\|s\|+2)P$	$1E_{G_T}$	Has
Wang[10]	Composite	$3lE_{G_1}$	$3E_{G_1}+1E_{G_T}$	$\|s\|E_{G_1}+(3\|s\|+2)P$	$1P$	Has
Li[19]	Prime	None	$(2l+6)E_G+4E_{G_T}$	$2E_{G_T}+4P$	$4E_{G_T}$	Has
Ours	Prime	$2yE_G$	$3E_G+1E_{G_T}$	$(2+\|s\|)E_{G_T}+(2\|s\|+3)P$	$1E_{G_T}$	Has

As shown in **Table 2**, Wang's [10] and our schemes achieve both verifiable outsourced encryption and decryption and leave little computation to the user with limited resources. The schemes in [11,16,18,19] only implement the decryption phase of the outsourcing calculation.

In addition, the scheme in [11] does not implement a verifiable function, and validation is essential for obtaining correct results. During the encryption phase, the computational cost for the user is only 4 exponentiations in the scheme in [10] and our scheme. This is less than that for the schemes in [11,16,18,19]. During the decryption phase, the user's computational cost of all schemes is constant.

In the outsourced decryption phase, the calculation of the scheme in [19] is a constant value $2E_{G_T}+4P$. However, a large amount of calculation in the decryption phase is outsourced to a cloud with powerful computing power, so the advantage of the scheme in [19] is not obvious to the user. Additionally, the scheme does not achieve outsourced encryption.

The scheme in [10] is based on the composite order bilinear group, while the other schemes are based on the prime order bilinear group. The computational efficiency of the composite order bilinear group is much lower than that of the prime order bilinear group. This is clearly reflected in the experimental comparison below.

Compared with other schemes, our schemes have a considerable advantage in terms of functionality and efficiency. In addition, our scheme requires hash and symmetric encryption algorithms in the encryption and verification processes, which increase the terminals' calculations. In any case, our scheme achieves both verifiable outsourced encryption and decryption in the terminals, outsourcing the complex attribute computation to cloud service providers. At the same time, our scheme supports verifiable outsourcing to ensure the safety and correctness of the system.

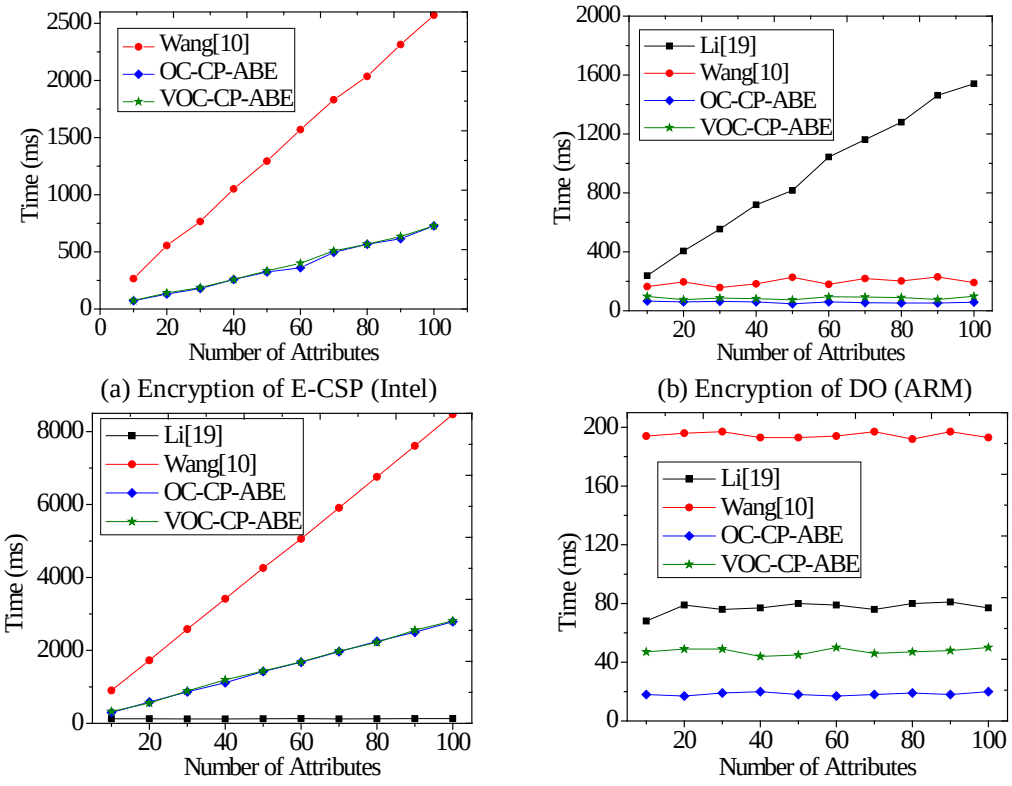
5.2 Experimental Comparison

To evaluate the practical performance of our scheme, we implement the scheme using 224-bit MNT elliptic curves from the Pairing-Based Cryptography library [23], and it is executed on an Intel(R) Core(TM) i5-2450M CPU @ 2.50 GHz with 8 GB of RAM running on a 64-bit Fedora 20 system and a 2.1-GHz ARM-based Samsung Exynos 7420 with 4 GB of RAM running Android OS. We use the Intel platform to simulate the cloud service provider and used the ARM platform to simulate the user terminal. For the $\lambda=80$ bits security parameter, we choose $l_{H_0}=l_{H_1}=160$ and encapsulate a random 128-bit symmetric key l_{SE}. In our scheme, we first hash the element C_T of group G_T to a random "seed" and then apply a pseudorandom number generator (e.g., the AES scheme) to extend it to a 512-bit key R. It sufficiently guarantees that $\log|R|-l_{H_1}\geq l_{SE}+2\log 2^{\lambda}$.

Our scheme has the advantage in terms of functionality and efficiency from the theoretical analysis. To evaluate the practical performance of our schemes, we implement the scheme provided in Section 4.1 and the scheme provided in Section 4.2. At the same time, we compare them with the schemes in [10,19]. To illustrate the efficiency of the verification mechanism, we also compare the verifiable scheme (VOC-CP-ABE) with the unverifiable scheme (OC-CP-ABE).

Experiment Setting: In a CP-ABE scheme, the complexity of the ciphertext policy impacts both the encryption and decryption time. To illustrate this, we generate ciphertext policies in the form of (S_1 and S_2 and … and S_n) to simulate the worst situation, where each S_i is an attribute. This approach ensures that all the ciphertext components are involved in the decryption computation. We generate 10 distinct policies in this form, increasing from 10 to 100. For each ciphertext policy, we repeat our experiment 20 times on the PC and 20 times on the ARM device, and we take the average values as the experimental results. We keep all the

instances completely independent of each other. The time is given in milliseconds. The *Encrypt.E-CSP Time*, *Encrypt.DO Time*, *Decrypt.D-CSP Time* and *Decrypt.DU Time* of our schemes are shown in **Fig. 4**.



(a) Encryption of E-CSP (Intel) (b) Encryption of DO (ARM)

(c) Decryption of D-CSP (Intel) (d) Decryption of DU (ARM)

Fig. 4. Experimental Comparison

We know that the computational power of the Intel platform is much greater than that of the ARM platform. **Fig. 4(a)** and **Fig. 4(c)** are implemented on the Intel platform simulating the cloud service provider, and **Fig. 4(b)** and **Fig. 4(d)** are implemented on the ARM platform simulating the user terminal. **Fig. 4(a)** and **Fig. 4(b)** imply that E-CSP undertakes most of the encryption work in the scheme of [10] and our scheme. A user should perform the entire encryption by himself in the scheme of [19]. Due to the limited computational power of the mobile terminal, it needs to spend a large amount of time to complete the encryption work. **Fig. 4(c)** and **Fig. 4(d)** imply that D-CSP undertakes most of the decryption work in all the schemes.

Because the scheme of [19] does not have an encrypted outsourcing function, there is no outsourced encryption curve for the scheme of [19] in **Fig. 4 (a)**. Therefore, in **Fig. 4 (b)**, the scheme of [19] requires huge computational time at the mobile terminal to complete the encryption operation, while the encryption time of the other schemes is constant. As shown in **Fig. 4 (c)**, the outsourced decryption time of the scheme of [19] is constant, which is superior to other schemes. However, a large amount of calculation in the decryption phase is outsourced to a cloud with powerful computing power, so the advantage of the scheme of [19] is not obvious to the user.

The scheme of [10] is based on the composite order bilinear group. The computational time of the composite order bilinear group is much larger than that of the prime order bilinear group. It can be seen from **Fig. 4** that although the scheme of [10] implements the outsourced encryption and decryption function, the time required for each process is higher than that for our scheme. The experimental data obtained from the above experimental process is also consistent with the theoretical analysis of Section 5.1.

Moreover, the encryption and decryption time of VOC-CP-ABE is approximately 30 ms longer than the time of OC-CP-ABE. This is because VOC-CP-ABE needs to calculate the hash function and symmetric encryption. Compared with other schemes, our schemes have a considerable advantage in terms of functionality and efficiency. Our scheme outsources the majority of encryption and decryption to cloud service providers, and the user can complete the encryption and decryption calculations in a shorter amount of time. Therefore, it is very important to outsource encryption and decryption to the cloud service provider for mobile terminals.

6. Conclusion

Cloud computing is an emerging computing paradigm in which IT resources and capacities are provided as services over the Internet. With the development of wireless access technologies and the popularity of mobile intelligent terminals, cloud computing is expected to expand to mobile environments. Attribute-based encryption, widely applied in cloud computing, incurs massive computational cost during the encryption and decryption phases. Additionally, the computational cost grows with the complexity of the access policy. This disadvantage becomes more serious for mobile devices because they have limited resources.

In this paper, we present an efficient verifiable outsourced ABE scheme based on the bilinear group of prime order to address this problem. Called the verifiable outsourced computation ciphertext-policy attribute based encryption scheme (VOC-CP-ABE), it provides a way to outsource intensive computing tasks during encryption and decryption to CSP without revealing the private information and leaves only marginal computation to the user.

In detail, we first construct a CP-ABE scheme with outsourced encryption and decryption based on the scheme in [15], in which the root node of the access policy must be an AND gate. However, our scheme is able to delegate encryption for any policy. Then, we use the key-encapsulated mechanism to improve OC-CP-ABE in Verifiability, that is, VOC-CP-ABE, where the CP-ABE scheme encrypts a symmetric session key, and the message is encrypted by the symmetric session key. At the same time, we use two hash functions to obtain two hash values. To verify the integrity of the symmetric encrypted ciphertext, we use a hash value on the concatenation of the ciphertext. The second hash value is then used to verify the correctness of the outsourced decryption. Then, we provide a formal security proof of its (selective) CPA security and verifiability. Finally, we discuss the performance of the proposed scheme with comparisons to several related works. In the case of the same function, our scheme based on the prime order bilinear group has a smaller computational cost than the scheme of [10] based on the composite order bilinear group.

References

[1] R. Buyya, C. S. Yeo, S. Venugopal and et al, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Generation computer systems*, vol. 25, no. 6, pp. 599-616, June, 2009. Article (CrossRef Link).

[2] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Proc. of Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 457-473, May 22-26, 2005. Article (CrossRef Link).

[3] V. Goyal, O. Pandey, A. Sahai and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. of the 13th ACM conference on Computer and communications security*, pp. 89-98, October 30-November 3, 2006. Article (CrossRef Link).

[4] J. Bethencourt, A. Sahai and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. of IEEE symposium on security and privacy*, pp. 321-334, May 20-23, 2007. Article (CrossRef Link).

[5] Y. Rouselakis and B. Waters, "Practical constructions and new proof methods for large universe attribute-based encryption," in *Proc. of the 2013 ACM SIGSAC conference on Computer & communications security*, pp. 463-474, November 4-8, 2013. Article (CrossRef Link).

[6] K. Emura, A. Miyaji, A. Nomura and et al, "A ciphertext-policy attribute-based encryption scheme with constant ciphertext length," in *Proc. of International Conference on Information Security Practice and Experience*, pp. 13-23, April 13-15, 2009. Article (CrossRef Link).

[7] A. Lewko, T. Okamoto, A. Sahai and et al, "Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption," in *Proc. of Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 62-91, May 30-June 3, 2010. Article (CrossRef Link).

[8] Y. Fang, Z. Wen, Q. Shen and et al, "POSTER: Ciphertext-policy attribute-based encryption method with secure decryption key generation and outsourcing decryption of ABE ciphertexts," in *Proc. of International Conference on Security and Privacy in Communication Systems*, pp. 585-589, October 26-29, 2015. Article (CrossRef Link).

[9] K. Zhang, J. Ma, J. Liu and H. Li, "Adaptively secure multi-authority attribute-based encryption with verifiable outsourced decryption," *Science China Information Sciences*, vol. 59, no. 9, pp. 99-105, September, 2016. Article (CrossRef Link).

[10] H. Wang, D. He, J. Shen and et al, "Verifiable outsourced ciphertext-policy attribute-based encryption in cloud computing," *Soft Computing*, pp. 1-11, July, 2016. Article (CrossRef Link).

[11] M. Green, S. Hohenberger and B. Waters, "Outsourcing the decryption of ABE ciphertexts," in *Proc. of USENIX Security Symposium*, pp.34-34, August 8-12, 2011. Article (CrossRef Link).

[12] D. Benjamin and M. J. Atallah, "Private and cheating-free outsourcing of algebraic computations," in *Proc. of Sixth Annual Conference on Privacy, Security and Trust*, pp. 240-245, October 1-3, 2008. Article (CrossRef Link).

[13] J. Li, C. Jia, J. Li and X. Chen, "Outsourcing encryption of attribute-based encryption with mapreduce," in *Proc. of International Conference on Information and Communications Security*, pp. 191-201, Otober 29-31, 2012. Article (CrossRef Link).

[14] J. Li, X. Chen, J. Li and et al, "Fine-grained access control system based on outsourced attribute-based encryption," in *Proc. of European Symposium on Research in Computer Security*, pp. 592-609, September 9-13, 2013. Article (CrossRef Link).

[15] Z. Zhou and D. Huang, "Efficient and secure data storage operations for mobile cloud computing," in *Proc. of the 8th International Conference on Network and Service Management*, pp. 37-45, October 22-26, 2012. Article (CrossRef Link).

[16] J. Li, X. Huang, J. Li and et al, "Securely outsourcing attribute-based encryption with checkability," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 8, pp. 2201-2210, August, 2014. Article (CrossRef Link).

[17] F. Armknecht, J. M. Bohli, G. O. Karame and et al, "Outsourced proofs of retrievability," in *Proc. of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pp. 831-843, November 3-7, 2014. Article (CrossRef Link).

[18] J. Lai, R. H. Deng, C. Guan and J. Wang, "Attribute-based encryption with verifiable outsourced decryption," *IEEE Transactions on information forensics and security*, vol. 8, no. 8, pp. 1343-1354, August, 2013. Article (CrossRef Link).
[19] J. Li, F. Sha, Y. Zhang and et al, "Verifiable outsourced decryption of attribute-based encryption with constant ciphertext length," *Security and Communication Networks*, vol. 2017, no. 2017, January, 2017. Article (CrossRef Link).
[20] Q. Chai and G. Gong, "Verifiable symmetric searchable encryption for semi-honest-but-curious cloud servers," in *Proc. of IEEE International Conference on Communications*, pp. 917-922, June 10-15, 2012. Article (CrossRef Link).
[21] L. Trevisan and S. Vadhan, "Extracting randomness from samplable distributions," in *Proc. of 41st Annual Symposium on Foundations of Computer Science*, pp. 32-42, November 12-14, 2000. Article (CrossRef Link).
[22] Y. Dodis, R. Ostrovsky, L. Reyzin and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," *SIAM journal on computing*, vol. 38, no. 1, pp. 97- 139, March, 2008. Article (CrossRef Link).
[23] B. Lynn, The pairing-based cryptography (PBC) library[OL]. http://crypto.stanford.edu/pbc. 2006.

Zhiyuan Zhao received an M.S. degree from Zhengzhou Information Science and Technology Institute, Zhengzhou, China, in 2015. He is currently pursuing a Ph.D. degree at the Institute of Information Science and Technology, Zhengzhou, China. His research interests include cryptography theory, especially attribute-based encryption.

Jianhua Wang received a Ph.D. degree from the Institute of Information Science and Technology, Zhengzhou, China, in 2008. He became a Full Professor in 2006. His research interests mainly focus on information security. He has authored and co-authored more than 100 journal and conference papers.