IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 13, 2020, accepted January 21, 2020, date of publication January 24, 2020, date of current version January 31, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2969223

Cost-Efficient Outsourced Decryption of Attribute-Based Encryption Schemes for Both Users and Cloud Server in Green Cloud Computing

YONGJIAN LIAO, (Member, IEEE), GANGLIN ZHANG, AND HONGJIE CHEN
School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China

Corresponding author: Yongjian Liao (liaoyj@uestc.edu.cn)

This work was supported in part by the Sichuan Science and Technology Program under Grant 2017GZDZX0002.

ABSTRACT To reduce a user's decryption cost and protect the private information from being leaked, Green et al. proposed an approach oursourcing the decryption of the attribute based encryption (ABE) scheme to the cloud server. Later, almost all ABE schemes with outsourced decryption (ABE-OD) used their model or approach. However, the cloud server needs to repeat the outsourced decryption service of the same ciphertext for distinct users satisfying the same access policy in these schemes. Green computing is the atmosphere conscientious and recyclable utilization of resources. The green cloud networks can reduce their cost or energy requirements by adapting its performance, optimizing resources management and services. The method is not efficient for the cloud server in the green cloud networks. In this article, to take into account recyclable utilization of resources for the cloud server, we put forward a new and secure approach to reduce total overhead of the cloud server when many users satisfying an access policy require the outsourced decryptions for the same ciphertext besides decreasing the decryption computation cost for users. Compared with the existing ABE-OD schemes, our total overhead of the cloud server is independent of the number of the users who satisfy an access policy and request the outsourcing decryption service. Finally, we extend our approach to a RCCA-secure ABE-OD scheme.

INDEX TERMS Green cloud computing, attribute-based encryption, outsourced decryption, cloud server, recyclable utilization, bilinear maps.

I. INTRODUCTION

The use oriented IT services to users is offered by cloud computing. Two of the outstanding advantages are the large storage space and the strong computing power for cloud computing. Nowadays, with the development of cloud computing persons gradually have gotten accustomed to store their pictures, contacts or some other files to the cloud servers. Meanwhile, strong computing power is also utilized by persons or companies. For the convenience of people's daily life, many novel applications are proposed in cloud computing.

On the one hand, cloud users/terminals can save their cost via outsourcing their data storage or computation to the cloud servers, while the cloud users/terminals are only viewed as "devices" of input and output. On the other hand, a user's data are out of control by herself/himself, how to protect the users' privacy is a key issue in academia and industry. Thus a sequence of security issues is considered, such as remotely auditing [1], [2], outsourcing computation [3], outsourcing verification [4] and keyword searching [5]. Although various cryptographic skills and methods were put forward, attribute-based encryption (ABE) [6], a fine-grained and flexible scheme for access structure becomes one of the most hot notions to be researched in cloud computing.

ABE put forward by Sahai and Waters [6], was taken into account an extended version of notion of identity-based encryption (IBE). It is efficient to perform one-to-many encryption model but other than broadcast encryption. Lately, according to the access control policy deploying, ABE schemes were categorized two distinct types, key-policy ABE

The associate editor coordinating the review of this manuscript and approving it for publication was Chien-Ming Chen.

(KP-ABE) [7] and ciphertext-policy ABE (CP-ABE) [8]. However, one main barrier is that the decryption cost of these ABE schemes is very high. Because the user's decryption cost and the ciphertext's length linearly grow with the access policy's complexity. It has become a critical obstacle for various applications in cloud computing, such as the applications on wireless sensor, smart phone.

In order to reduce the computation cost of ABE's decryption algorithm and use powerful computational capability of the cloud server or some proxies, Green *et al.* [9] presented the notion of ABE with outsourced decryption (ABE-OD), which was called GHW method. A user need spend a small cost decrypting a ciphertext via a cloud server completes a large number of computation in their scheme. That is to say, the cloud server first outputs a transformed ciphertext by computing a delegated transformation key and the original ciphertext, and next the user can obtain the corresponding plaintext via computing "decryption" algorithm. While for security consideration, the ABE-OD scheme at the outsourcing process couldn't leak any information about the plaintext. However, there are two other issues they didn't solve in their ABE-OD scheme. One is that there is no mechanism to ensure the transformed ciphertext's correctness. The other is that the authority needs to firstly generate user's private key and then generate user's transformation key (the private key cannot be used any longer), which causes the scheme isn't fine-grained for user and increments the overhead of the authority.

To solve these problems, Lai *et al.* [10] used the GHW method to construct an ABE-OD scheme called verifiable outsourced decryption of ABE (ABE-VOD), which can verify the transformed ciphertext's correctness by a proxy or the cloud server. A random message and a plaintext are encrypted and meanwhile generated a commitment by the data owner in their ABE-VOD scheme. While the data receiver can make use of her/his private key to create a retrieving key and a transformation key which is utilized to produce a transformed ciphertext. In their decryption algorithm or outsourced decryption algorithm, the commitment is to make use of checking the generated transformed ciphertext's correctness. When the attributes set meets the ciphertext's access structure, the user is able to verify the transformed ciphertext's correctness. The model of their ABE-OD is described in FIGURE 1. Subsequently, relying on distinct scenarios or different correctness-checking methods, several ABE-VOD schemes were presented [11]–[29], while all these schemes used the GHW skill to design the outsourced decryption. Although Qin *et al.* [30] and Zhao and Wang [21] also put forward ABE-VOD schemes, they required the authority to produce the transformation key. Xu *et al.* [31] constructed an ABE-VOD scheme from multilinear map [32] that is secure based on k-multilinear Decisional Diffie-Hellman problem. However, Hu and Jia [33] showed that construction of the multilinear map [32] is not secure.

Arbitrary usage of cloud computing can lead to uneconomical energy consumption in data storage, processing and communication. Hence, green cloud computing solutions aim not only to save energy but also reduce operational costs and carbon footprints on the environment [34].

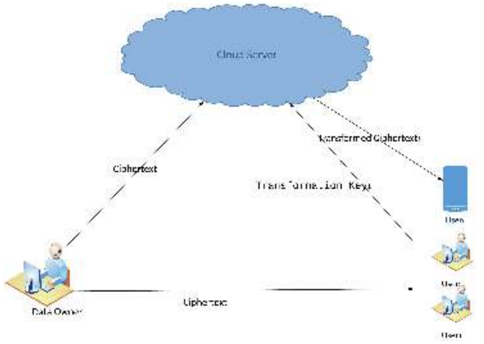

FIGURE 1. Architecture for ABE-OD.

While green computing is the atmosphere conscientious and recyclable utility of resources [35]. The green cloud networks can reduce their energy requirements by adapting their performance when it deploys, manages and provides the services [36].

A. MOTIVATION

It is the key technique for implementing the green cloud computing to reuse the resource and reduce the total energy consumption on the condition of guaranteeing the quality of service for performing the same task.

We analyze the outsourced decryption model proposed by Lai *et al.* [10] in the FIGURE 1. Every user generates her/his transformation key and retrieving key. When a user $User_1$ starts to outsource decryption of a ciphertext C, she/he sends the transformation key TK_1 to the cloud server and gets a corresponding transformed ciphertext TCT_1 from the cloud server. The value of the transformed ciphertext TCT_1 relies on C and TK_1. Another user $User_2$ also requires to outsource decryption for the same ciphertext C, so she/he and the cloud server will interactively repeat the previous process to produce the transformed ciphertext TCT_2. Since their transformation keys are different, we have

$$TCT_1 \neq TCT_2.$$

But the two users obtain the same plaintext after they decrypt the ciphertext C. The ABE is a one-to-many encryption concept. That is to say, there are many users who belong to the same set of attributes satisfying the access policy and can decrypt the ciphertext to get the same plaintext but their transformation keys and retrieving keys are different. Accordingly, to consider the cloud server's computation cost, it needs to compute n transformed ciphertexts for the ciphertext C of which the corresponding plaintext is the same if n users ask for the cloud server's outsourced decryption service, where n is the number of users whose attributes satisfy the access policy.

Although we suppose that the cloud server has much strongly computation power, it seems that the existing outsourced decryption manner wastes the computing resource of the cloud server which is required to help the users to repeatedly convert the same ciphertext into the transformed ciphertexts which correspond the same plaintext respectively. *Thus, an ideal solution is that the cloud server can repeatedly use the transformed ciphertext of a ciphertex C for ABE-OD scheme in green cloud networks.* That is to say, the cloud server computes a transformed ciphertext for the same ciphertext once which can be used by all users who satisfy the access policy.

To our best knowledge, all existing ABE-OD schemes solved the problem of reducing the computation cost of the users and preventing the cloud server from cheating for every outsourced decryption. No scheme solves the problem of efficiently reduceing the total computation cost of the cloud server besides the users at the same time.

Thus, the goal of our work is to find an efficient outsourced decryption method of ABE scheme to reduce the cloud server's computation cost besides reducing the computation cost of users.

B. CONTRIBUTION

In this article, we propose a new approach to outsource the decryption of the ABE scheme. Compared with the GHW method, our method is much more efficient for the cloud server besides reducing the computation cost of decryption of users if many users require the outsourced decryption services for the same ciphertext. Because the cloud server only computes the transformed ciphertext once for each ciphertext and many users satisfying the common access policy. To the best of our knowledge, there is no other scheme researching efficiency of the computation cost of the cloud server and users at the same time. Our approach has some advantages as follows.

- For the cloud server, our method only needs to compute the transformed ciphertext once for any ciphertext and an access policy. That is to say, when the cloud server find that checks the transformation keys are used and the transformed ciphertext of the ciphertext has been generated (from the record having done), it returns the same transformed ciphertext.
- For any user, our method doesn't require the additional computation cost to produce the transformation key and additional storage space to store the user's transformation key, besides storing the user's private key.

Finally, we extend our approach to a RCCA-secure ABE-OD scheme.

C. ORGANIZATION

The rest sections are organized below. Some basic notions are recalled in section 2. Then in section 3 we propose our construction. We analyze our construction's performance and security in section 4 and 5, respectively. In section 6 we utilize our method to design a RCCA-secure construction and analyze its security. Finally, we concludes our paper in last section.

TABLE 1. The abbreviations and notations.

Notations	Description
ABE	Attribute-based encryption
CP-ABE	Ciphertext policy attribute-based encryption
KP-ABE	Key policy attribute-based encryption
IBE	Identity-based encryption
ABE-OD	attribute-based encryption with outsourced decryption
GHW	M. Green, S. Hohenberger and B. Waters
ABE-VOD	verifiable outsourced decryption of attribute-based encryption
TK	Transformation key
TCT	Transformed ciphertext
TCT	Public Parameters
LSSS	Linear secret sharing schemes
PK	Public key
SK	Secret key
RK	Retrieving key
BDH	Bilinear Diffie-Hellman
PK	Private key
CS	The cloud server

II. PRELIMINARIES

Firstly we recall some notions, the ABE-OD model and its security model.

A. NOTATION

In order to clearly understand our paper, the abbreviations used in the paper are given in TABLE 1.

B. BILINEAR MAP

The order of two multiplicative groups $\mathbb{G}_1$, $\mathbb{G}_2$ is p which is prime. If a map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ [37] fulfills the properties below, then it is called admissible bilinear map. Let $g \in \mathbb{G}_1$ be a random element which is not the identity.

- Bilinear: For any elements $u, v \in \mathbb{Z}_p^*$, $e(g^u, g^v) = e(g, g)^{uv}$ holds.
- Non-degenerate: $e(g, g) \neq 1_{\mathbb{G}_2}$, where $1_{\mathbb{G}_2}$, $1_{\mathbb{G}_1}$ are the identity elements of $\mathbb{G}_2$ and $\mathbb{G}_1$, respectively.
- Computable: For every element $g'', g' \in \mathbb{G}_1$, $e(g'', g')$ is able to be computed efficiently.

Defintion 1. $\mathbb{G}_1$, $\mathbb{G}_2$ and e are defined above. The bilinear Diffie-Hellman (BDH) problem is:

$e(g, g)^{\eta\theta\vartheta}$ can be calculated by a probabilistic polynomial-time algorithm (PPT algorithm) for $g, g^\eta, g^\theta, g^\vartheta \in \mathbb{G}_1$. Where η, θ, $\vartheta \in \mathbb{Z}_p^*$ is randomly selected.

For any PPT algorithm $\mathcal{A}$, advantage of outputing $e(g, g)^{\eta\theta\vartheta} \in \mathbb{G}_2$ is $Adv_{\mathcal{A}}^{\mathrm{BDH}}(\tau) =$

$$\Pr[\mathcal{A}(g, g^\eta, g^\theta, g^\vartheta) = e(g, g)^{\eta\theta\vartheta} : g, g^\eta, g^\theta, g^\vartheta \in \mathbb{G}_1].$$

For any $\mathcal{A}$ above, if the advantage $Adv_{\mathcal{A}}^{\mathrm{BDH}}(\tau)$ is negligible in security parameter τ, the BDH assumption holds.

C. LINEAR SECRET SHARING SCHEMES

The linear secret sharing scheme (LSSS) Π [38] is described below. Set $\mathcal{P}$ to be a parties set. Π satisfies the following conditions.

- Each party's secret shares form a vector in $\mathbb{Z}_p$.
- A is a matrix which is of n columns and l rows. Set a map $\rho : i \rightarrow \rho(i)$, where $\rho(i)$ that is the i-th row A_i

of the matrix A denotes the party labeling row i. For a column vector $\vec{v}_i = (s, v_2, \ldots, v_n)^T$, $s, v_2, \ldots, v_n \in \mathbb{Z}_p$ are randomly picked, where the secret $s \in \mathbb{Z}_p$ is shared. Let $(A\vec{v})_i$ belong to $\rho(i)$. $A\vec{v}$ is the l shares vector for s w.r.t. Π.

- Let $\mathbb{A}$ be an access policy of an LSSS Π, any authorized set S be an element of $\mathbb{A}$. Set $I = \{i : \rho(i) \in S\}$ to be a subset of $[l] = \{1, 2, \ldots, l\}$. For the set $\{(A\vec{v})_i\}_{i\in I}$, if it is a valid share set of any secret s w.r.t. Π, then we can calculate the set $\{o_i \in \mathbb{Z}_p\}_{i\in I}$ which satisfies the condition $\sum\limits_{i\in I} o_i(A\vec{v})_i = s$.

Note: The "target" vector of any LSSS is $(1, 0, \ldots, 0)$. For any unauthorized set of I, $(1, 0, \ldots, 0)$ isn't an element of the span of the rows of I. Otherwise, it is an element of the span of I for any authorized set of rows of I in A [40].

D. SYSTEM MODEL

Since the model of ABE-OD Scheme [9] required that the authority generates the transformation key, it doesn't include Decrypt algorithm and GenTK$_{out}$ algorithm, which isn't flexible for the user. We use the model defined by Lai et al. [10], which includes seven algorithms: Setup, Encrypt, KeyGen, GenTK$_{out}$, Transform$_{out}$, Decrypt and Decrypt$_{out}$. The detailed is the following description.

- Setup$(1^\tau, U)$. This algorithm produces a master secret key msk and public parameters PK on input 1^τ and U.
- KeyGen(msk, PK, S). This algorithm produces users' private decryption key SK of any attribute set S.
- Encrypt$(PK, M, \mathbb{A})$. It produces a ciphertext CT according to $\mathbb{A}$.
- Decrypt(SK, CT). It utilizes SK to decrypt the ciphertext, if S meets $\mathbb{A}$.
- GenTK$_{out}(PK, SK)$. This algorithm produces a transformation key TK used to outsourced decryption computation and a corresponding retrieving key RK.
- Transform$_{out}(TK, CT)$. This algorithm produces the transformed ciphertext TCT.
- Decrypt$_{out}(CT, TCT, RK)$. When S meets $\mathbb{A}$, it utilizes CT, TCT and RK to decrypt the ciphertext; Otherwise, it outputs $'\perp'$.

E. SECURITY MODEL

Firstly, we recall the definition of chosen plaintext attack (CPA) for ABE-OD in [15]. We take into account the selectively CPA security model for ABE-OD via the interactive game between $\mathcal{C}$ (a challenger) and $\mathcal{A}$ (an adversary) below.

- **Init**. $\mathcal{A}$ sets an access policy $\mathbb{A}^*$ as a challenge.
- **Setup**. $\mathcal{C}$ produces the public parameters PK and sends them to $\mathcal{A}$; $\mathcal{C}$ produces the master secret key msk and keeps it secret.
- **Phase 1**. The challenger $\mathcal{C}$ sets an empty table T and an empty set D initially. $\mathcal{A}$ can enquire as follows.
  - *Private key* query. $\mathcal{A}$ inquiries private key for any attribute set S and records it in the set D. $\mathcal{C}$ answers all private key queries on the attribute sets S which cannot fulfill $\mathbb{A}^*$.
  - *Transformation key* query. The adversary $\mathcal{A}$ inquiries the transformation key for any attribute sets S, $\mathcal{C}$ answers all queries on them and stores them in the table T.
- **Challenge**. In this phase, $\mathcal{C}$ randomly picks $c \in \{0, 1\}$ from two same length messages M_0 and M_1 which are produced by $\mathcal{A}$, and calculates

  $$CT^* = Encrypt(PK, M_c, \mathbb{A}^*).$$

  At last, $\mathcal{C}$ sends CT^* to the adversary.
- **Phase 2**. $\mathcal{A}$ repeats making some queries as in Phase 1.
- **Guess**. $\mathcal{A}$ guesses a bit $c' \in \{0, 1\}$ of c.

The advantage of $\mathcal{A}$ winning above game is

$$|\frac{1}{2} - \Pr[c = c']|,$$

here, the probability $\Pr[c = c']$ is taken over random coins by $\mathcal{C}$ and $\mathcal{A}$.

Definition 3: An ABE-OD scheme is selectively CPA-secure if the above advantage is negligible in τ for any PPT adversary $\mathcal{A}$.

Next, we recall the definition of the RCCA-secure of the ABE-OD. Compared with the CPA-secure, the RCCA adversary $\mathcal{A}$ can also take the decryption queries besides the above game. The decryption queries are described as follows:

- **Phase 1**. *Decrypt* queries. $\mathcal{A}$ inquiries the decryption on S and CT. The adversary $\mathcal{C}$ produces and returns the corresponding plaintext M.
- **Phase 2**. *Decrypt* queries. $\mathcal{A}$ repeats making the decryption queries as in Phase 1 except that $\mathcal{C}$ aborts if the corresponding plaintext is M_0 or M_1.

The advantage of $\mathcal{A}$ winning above game is

$$|\frac{1}{2} - \Pr[c = c']|,$$

here, the probability $\Pr[c = c']$ is taken over random coins by $\mathcal{C}$ and $\mathcal{A}$.

Definition 4. An ABE-OD scheme is selectively RCCA-secure if the above advantage is negligible in τ for any PPT adversary $\mathcal{A}$.

III. OUR CPA-SECURE CONSTRUCTION

We design an ABE-OD scheme which is based on the construction of Waters [40] below.

- Setup $(1^\tau, U)$. On input 1^τ and $U = \{at_1, \cdots, at_l\}$. To produce a bilinear map e with groups $\mathbb{G}_1, \mathbb{G}_2$ as section II, where the order of $\mathbb{G}_1$, $\mathbb{G}_2$ is prime p. Select a random element $g \neq 1_{\mathbb{G}_1}$ of $\mathbb{G}_1$, elements $a, \alpha \in \mathbb{Z}_p^*$ and l random elements $T_1, \cdots, T_l$ of $\mathbb{G}_1$, to calculate

  $$y = g^a, Y = e(g, g)^\alpha.$$

  Set the public parameters $PK = (\mathbb{G}_1, g, y, Y, \mathbb{G}_2, T_1, \cdots, T_l)$, the master key $msk = \alpha$.

- KeyGen(msk, PK, S). It selects an element $\lambda \in \mathbb{Z}_p^*$ randomly, calculates
$$\{K_i = T_i^{\lambda}\}_{at_i \in S}, K = y^{\lambda} g^{\alpha}, K_0 = g^{\lambda}.$$
Finally it sets $SK_S =$
$$(S, \{K_i : at_i \in S\}, K, K_0)$$
as the user's private key.
- Encrypt$(M, \mathbb{A})$. It uses a message $M \in \mathbb{G}_2$ and $\mathbb{A} = (A, \rho)$ to calculate a ciphertext, where $l \times n$ matrix A and a map ρ defined above. It randomly picks $\lambda_i \in \mathbb{Z}_p^*$ and $\vec{v} =$
$$(v, v_2, \cdots, v_n) \in \mathbb{Z}_p^n,$$
where v is to be shared. Then it calculates:
$$C_0 = g^{v}, C_M = Y^{v} M,$$
$$(D_1 = g^{\lambda_1}, C_1 = T_{\rho(1)}^{-\lambda_1} g^{aA_1 \cdot \vec{v}}),$$
$$\cdots, (D_l = g^{\lambda_l}, C_l = T_{\rho(l)}^{-\lambda_l} g^{aA_l \cdot \vec{v}}).$$
Set $CT = (C_0, C_M, (D_1, C_1), \cdots, (D_l, C_l))$ to be the ciphertext.
- Decrypt (SK, S, CT). This algorithm decrypts the ciphertext CT for input SK, S and $(\mathbb{A}, CT)$.
When S meets $\mathbb{A}$, this algorithm calculates $o_i \in \mathbb{Z}_p^*$ $(i \in I)$ which satisfies $\Sigma_{i \in I} o_i A_i = (1, 0, \cdots, 0)$ firstly, and then calculates the corresponding plaintext
$$M' = \frac{C_M \prod_{i \in I} (e(D_i, K_{\rho(i)}) e(C_i, K_0))^{o_i}}{e(C_0, K)}.$$
- GenTK$_{out}(SK)$. This algorithm uses $SK = (K, K_0, \{K_i : at_i \in S\})$ to set $(K_0, \{K_i : at_i \in S\})$ as the transformation key TK, and K as the retrieving key RK, respectively.
- Transform$_{out}(TK, CT)$. This algorithm uses CT and TK to calculate
$$TCT = \prod_{i \in I} (e(D_i, K_{\rho(i)}) e(C_i, K_0))^{o_i},$$
which equals $e(g, g)^{av\lambda}$.
- Decrypt$_{out}(CT, TCT, RK)$. This algorithm uses CT, TCT and RK to calculate the corresponding plaintext
$$M' = \frac{C_M TCT}{e(C_0, K)}.$$

Note: Compared the GHW method to generate the transformation key, our approach does neither store the transformation key (if compute the transformation key offline), nor increase the overhead of outsourcing decryption (if compute the transformation key online).

It is obvious for the correctness of our scheme, we omit it.

TABLE 2. **Comparison of two approaches.**

Approaches	[9]	[15]	Ours
Overhead (CS)	$\xi(2\lvert I\rvert+1)t_{BP}$	$\xi(2\lvert I\rvert+1)t_{BP}$	$2\lvert I\rvert t_{BP}$
Overhead (User)	$t_{\mathbb{G}_2} + t_{EXP_{\mathbb{G}_2}}$	$t_{\mathbb{G}_2} + t_{EXP_{\mathbb{G}_2}}$	$2t_{\mathbb{G}_2} + t_{BP}$
PK	N/A	store	store
Length or Overhead of TK	$(\lvert I\rvert+2)\lvert\mathbb{G}_1\rvert$	$(\lvert I\rvert+2)\lvert\mathbb{G}_1\rvert$ or $(\lvert I\rvert+2)t_{EXP_{\mathbb{G}_1}}$	N/A

Where $|I|$, $|\mathbb{G}_1|$ and ξ represent the order of set I, the average length of the element of $\mathbb{G}_1$ and the number of users satisfying the access policy, respectively. $t_{EXP_{\mathbb{G}_2}}$, t_{BP}, $t_{\mathbb{G}_2}$ represent the average time of an exponentiation operation on $\mathbb{G}_2$, a bilinear pairing and the multiplicative operation on $\mathbb{G}_2$, respectively. The CS, PK, TK represent the cloud server, the private key and the transformation key, respectively.

IV. PERFORMANCE ANALYSIS

The main goal of our method is to reduce the total overhead of the cloud server for outsourced decryption of ABE scheme besides decreasing the user's overhead of the decryption of ABE scheme when many users submit the outsourced decryption service to the cloud server. Although Green *et al.* [9] and Li *et al.* [15] used the same outsourcing method to outsource the decryption, the entities to generate the transformation key were different and subsequent works were used these two methods, respectively. We compare their methods with ours in TABLE 2.

For every ciphertext, the cloud server for the GHW method [9], [15] needs to compute $(2|I| + 1)$ bilinear pairings for every outsourced decryption. Thus, the cloud server needs to compute at most $\xi(2|I| + 1)$ bilinear pairings when ξ users require the outsourced decryption services. While the cloud server merely needs to compute $2|I|$ bilinear pairings for the total overhead of all outsourced decryption for our method via simply comparing their transformation keys. Because the cloud server only could check whether the transformation keys are the same or not. For example, the cloud server could setup a list L which includes elements as a form l_{CT} = (CT, TK, TCT)[1] for every ciphertext CT. When a user submits a decryption service of the ciphertext CT, the cloud server returns TCT if l_{CT} exists in L; Otherwise, it computes TCT and returns TCT to the user and records (CT, TK, TCT) in the list L. Thus the total overhead of all outsourced decryption for the GHW method grows with the number of users ξ, but our method is independent of it. However, for the overhead of every user, their method is somewhat more efficient than our method. Because our method requires the user to compute 1 bilinear pairing and 2 multiplications over $\mathbb{G}_2$, but the GHW method only requires the user to compute 1 multiplication and exponentiation over $\mathbb{G}_2$. This overhead can be performed by the vast majority of devices, even some resource-constrained devices. We implement the our approach by using pairing- based cryptography library [39] on an Intel Core i5-3210M (2.50 GHz) machine with the Windows XP operating system and 4G RAM. The relative

[1] In order to reduce the storage, we could store a hash value of TK but not TK.

TABLE 3. Time of cryptographic operations(in milliseconds).

t_{BP}	$t_{\mathbb{G}_2}$	$t_{EXP_{\mathbb{G}_2}}$
16.064ms	0.013ms	1.882ms

Implementation of pairing-based cryptography library [39] on an Intel Core i5-3210M (2.50 GHz) machine with 4G RAM and the Windows XP operating system.

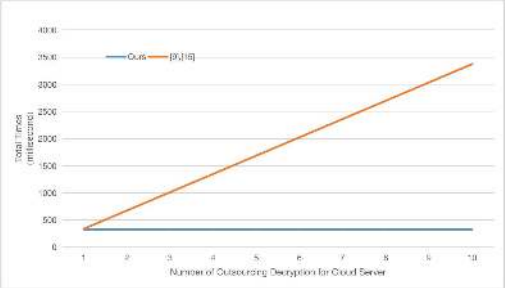

FIGURE 2. Total time for the cloud server.

times of considered cryptographic operations are given in TABLE 3. We set $|I| = 10$. The detailed comparison of operating time for the cloud server is in FIGURE 2.

Then, we analyze the difference between these two methods for users further in TABLE 2. For our method, every user needs to store the private key of an attributes set satisfying the access policy, eg. $(|I| + 2)|\mathbb{G}_1|$ bits. This is the same as [15], but Green et al.'s scheme [9] discarded the private key and didn't design the decryption algorithm. Our method does not need to additional computation cost and storage for the transformation key and retrieving key, while every user needs to additionally store $(|I|+2)|\mathbb{G}_1|$ bits transformation key and retrieving key or compute $(|I|+2)$ exponentiation operations on the group $\mathbb{G}_1$ to generate the transformation key TK once in [15].

The last column represents the entity who generates the transformation key. For flexibility, we take into account the model proposed by Li *et al.* [15]. For the GHW model, our approach is also useful to reduce the total overhead of the cloud server being outsourced decryption for all users and the cost generating the transformation key.

Thus, compared with the existing schemes, our approach can reduce the total overhead for the cloud server when ξ ($\xi > 1$) users require the outsourced decryption service, and it also can reduce decryption cost for every user and generation of the transformation key.

V. SECURITY ANALYSIS

Intuitively, our scheme is as secure as the scheme proposed by Waters [40]. For Waters's scheme, any adversary can select an element $\lambda' \in \mathbb{Z}_p^*$ randomly and then calculate

$$TK'_S = (K'_0 = g^{\lambda'}, \{K'_i = T_i^{\lambda'}\}_{at_i \in S}).$$

Obviously, we have the following facts.

- *Fact* 1. The distribution of TK_S for our scheme is identical to that of TK'_S.
- *Fact* 2. Given TK'_S, there exists $K' = g^{\alpha} y^{\lambda'}$ but the adversary cannot compute K' even if it knows the λ', and (K', TK'_S) is another valid private key of the attribute set S.

From above two facts, we can know if the Waters's ABE scheme [40] is CPA-secure, it is also CPA-secure even if TK_S is leaked. Thus, we have a lemma below.

Lemma: Let the Waters's CP-ABE scheme [40] be selectively CPA-secure. The scheme is also selectively CPA-secure even if TK_S is leaked.

For our scheme, any untrusted cloud server is able to obtain the transformation key from $GenTK_{out}$ algorithm before it will output the transformed ciphertext, which is equivalent to that the adversary can make transformation key queries obtain $TK_S = (K_0, \{K_i = T_i^{\lambda}\}_{at_i \in S})$ in the game of *definition* 3, where S satisfies the access policy $\mathbb{A}$. In the light of **Lemma**, we have a theorem below.

Theorem 1: Our construction is selectively CPA-secure if the Waters's CP-ABE scheme [40] is selectively CPA-secure.

VI. OUR RCCA-SECURE CONSTRUCTION

Here, we extend our scheme to the stronger selectively RCCA-secure construction. We get this result by also using the technique proposed by Fujisaki and Okamoto [41]. The construction similar to [9] is described below.

- Setup $(1^{\tau}, U)$. The same as it in the aforementioned ABE-OD scheme besides selecting secure hash functions

$$H_2 : \{0, 1\}^* \rightarrow \{0, 1\}^k,$$
$$H_1 : \{0, 1\}^* \rightarrow \mathbb{Z}_p^*.$$

- KeyGen(msk, PK, S). The same as it in the aforementioned ABE-OD scheme.
- Encrypt$(M, \mathbb{A})$. This algorithm uses $M \in \{0, 1\}^k$ and $\mathbb{A}$ = (A, ρ) to calculate a ciphertext. It picks an element R of $\mathbb{G}_2$ randomly and calculates

$$\nu = H_1(R, M) \in \mathbb{Z}_p^* \text{ and } r = H_2(R) \in \{0, 1\}^k.$$

Then it uses ν to calculate $(D_1, C_1), \ldots, (D_l, C_l)$ as it in the aforementioned construction, where ν is as part of the vector $\vec{v}$. Finally it calculates:

$$C_0 = g^{\nu}, C_R = Y^{\nu} R, C_M = r \oplus M,$$

Set $CT = (C_0, C_M, C_R, (D_1, C_1), \ldots, (D_l, C_l))$ as the ciphertext. Decrypt(SK, S, CT). This algorithm uses SK, a user's attribute set S and a ciphertext $(\mathbb{A}, CT)$ to calculate

$$R' = \frac{C_R \prod_{i \in I} (e(D_i, K_{\rho(i)}) e(C_i, K_0))^{o_i}}{e(C_0, K)}.$$

Then it calculates $M' = H_2(R') \oplus C_M$, and checks the following equation

$$g^{H_1(R', M')} \overset{?}{=} C_0.$$

The algorithm outputs M' if the equation above holds; Otherwise, it outputs symbol $\perp$.

- GenTK$_{out}(SK)$. The same as it in the aforementioned CP-ABE-OD scheme.
- Transform$_{out}(TK, CT)$. The same as it in the aforementioned CP-ABE-OD scheme.
- Decrypt$_{out}(CT, TCT, RK)$. This algorithm uses CT, TCT and RK to calculate

$$R' = \frac{C_R \cdot TCT}{e(C_0, K)}.$$

Then it calculates

$$M' = H_2(R') \oplus C_M,$$

and checks the equation

$$g^{H_1(R',M')} \overset{?}{=} C_0.$$

The algorithm outputs M' if the equation above holds; Otherwise, it outputs symbol $\perp$.

We have that our scheme is selectively RCCA-secure but not selectively CCA-secure. The security proof of the above scheme is similar to the *Theorem* 3.2 in [40].

Theorem 2: In the random oracle model our construction above is selectively RCCA-secure if the Waters's CP-ABE scheme [40] is selectively CPA-secure.

Obviously, the efficiency of the RCCA-secure construction put forward by Green *et al.* [9] and it of our RCCA-secure construction are similar to them in section IV.

Our method cannot only be used to construct the KP-ABE-OD scheme which is analogous to describe the GHW KP-ABE-OD scheme [9], but also can be used to construct the ABE-VOD schemes [10]–[13]. Here, we omit these concrete constructions.

VII. CONCLUSION

The resource is reused and on the condition of guaranteeing the quality of service the total energy consumption is reduced for performing the same task are key features in the green cloud computing. In this paper, we considered the outsourced decryption of ABE scheme in the green cloud computing. In order to reduce the total overhead of the cloud server when many users satisfying the access policy require their outsourced decryptions for the same ciphertext, we put forward a new and secure method used in the ABE-OD schemes. Our approach can reduce the overhead of both users and the cloud server. That is to say, the cloud server's overhead only needs constant computation cost for all outsourced decryptions of the same ciphertext besides reducing the user's computation cost. Finally, we extended our approach to a RCCA-secure ABE-OD scheme.

REFERENCES

[1] J. Yu, K. Ren, and C. Wang, "Enabling cloud storage auditing with verifiable outsourcing of key updates," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 6, pp. 1362–1375, Jun. 2016.

[2] Y. Fan, Y. Liao, F. Li, S. Zhou, and G. Zhang, "Identity-based auditing for shared cloud data with efficient and secure sensitive information hiding," *IEEE Access*, vol. 7, pp. 114246–114260, 2019, doi: 10.1109/access.2019.2932430.

[3] Q. Su, J. Yu, C. Tian, H. Zhang, and R. Hao, "How to securely outsource the inversion modulo a large composite number," *J. Syst. Softw.*, vol. 129, pp. 26–34, Jul. 2017.

[4] Y. Liao, Y. He, F. Li, and S. Zhou, "Analysis of a mobile payment protocol with outsourced verification in cloud server and the improvement," *Comput. Standards Inter.*, vol. 56, pp. 101–106, Feb. 2018.

[5] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.*, 2004, pp. 506–522.

[6] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Proc. 24th Int. Conf. Adv. Cryptol.*, 2005, pp. 457–473.

[7] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proc. 13th ACM Conf. Comput. Commun. Secur.-CCS*, Alexandria, VA, USA, Oct./Nov. 2006, pp. 89–98.

[8] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. IEEE Symp. Secur. Privacy (SP)*, Berkeley, CA, USA, May 2007, pp. 321–334.

[9] M. Green, S. Hohenberger, and B. Waters, "Outsourcing the decryption of ABE ciphertexts," in *Proc. 20th USENIX Conf. Secur.*, San Francisco, CA, USA, Aug. 2011, p. 34.

[10] J. Lai, R. H. Deng, C. Guan, and J. Weng, "Attribute-based encryption with verifiable outsourced decryption," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 8, pp. 1343–1354, Aug. 2013.

[11] X. Mao, J. Lai, Q. Mei, K. Chen, and J. Weng, "Generic and efficient constructions of attribute-based encryption with verifiable outsourced decryption," *IEEE Trans. Depend. Sec. Comput.*, vol. 13, no. 5, pp. 533–546, Sep. 2016, doi: 10.1109/tdsc.2015.2423669.

[12] S. Lin, R. Zhang, H. Ma, and M. Wang, "Revisiting attribute-based encryption with verifiable outsourced decryption," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 10, pp. 2119–2130, Oct. 2015.

[13] J. Li, X. Huang, J. Li, X. Chen, and Y. Xiang, "Securely outsourcing attribute-based encryption with checkability," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 8, pp. 2201–2210, Aug. 2014.

[14] J. Li, F. Sha, Y. Zhang, X. Huang, and J. Shen, "Verifiable outsourced decryption of attribute-based encryption with constant ciphertext length," *Secur. Commun. Netw.*, vol. 2017, no. 2, pp. 1–11, 2017, Art. no. 3596205, doi: 10.1155/2017/3596205.

[15] J. Li, Y. Wang, Y. Zhang, and J. Han, "Full verifiability for outsourced decryption in attribute based encryption," *IEEE Trans. Serv. Comput.*, to be published, doi: 10.1109/tsc.2017.2710190.

[16] H. Chen and Y. Liao, "Improvement of an outsourced attribute-based encryption scheme," *Soft Comput.*, vol. 23, no. 22, pp. 11409–11417, Nov. 2019, doi: 10.1007/s00500-019-04088-y.

[17] Y. Liao, Y. He, F. Li, S. Jiang, and S. Zhou, "Analysis of an ABE scheme with verifiable outsourced decryption," *Sensors*, vol. 18, no. 2, p. 176, Jan. 2018, doi: 10.3390/s18010176.

[18] R. Mohammed, Y. Liao, F. Li, S. Zhou, and H. Abdalla, "IBEET-RSA: Identity-based encryption with equality test over RSA for wireless body area networks," in *Proc. Mobile Netw. Appl.*, 2019, pp. 1–11, doi: 10.1007/s11036-019-01215-9.

[19] Y. Miao, Q. Tong, K.-K.-R. Choo, X. Liu, R. H. Deng, and H. Li, "Secure online/offline data sharing framework for cloud-assisted industrial Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8681–8691, Oct. 2019, doi: 10.1109/jiot.2019.2923068.

[20] K. Fan, J. Wang, X. Wang, H. Li, and Y. Yang, "A secure and verifiable outsourced access control scheme in fog-cloud computing," *Sensors*, vol. 17, no. 7, p. 1695, Jul. 2017.

[21] Z. Zhao and J. Wang, "Verifiable outsourced ciphertext-policy attribute-based encryption for mobile cloud computing," *KSII Trans. Internet Inf. Syst.*, vol. 11, no. 6, pp. 3254–3272, 2017.

[22] J. Ning, Z. Cao, X. Dong, K. Liang, H. Ma, and L. Wei, "Auditable σ-time outsourced attribute-based encryption for access control in cloud computing," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 1, pp. 94–105, Jan. 2018, doi: 10.1109/tifs.2017.2738601.

[23] P. K. Premkamal, S. K. Pasupuleti, and P. J. A. Alphonse, "A new verifiable outsourced ciphertext-policy attribute based encryption for big data privacy and access control in cloud," *J. Ambient Intell. Humanized Comput.*, vol. 10, no. 7, pp. 2693–2707, Jul. 2019.

[24] C. Zuo, J. Shao, G. Wei, M. Xie, and M. Ji, "CCA-secure ABE with outsourced decryption for fog computing," *Future Gener. Comput. Syst.*, vol. 78, pp. 730–738, Jan. 2018, doi: 10.1016/j.future.2016.10.028.

[25] V. K. Arthur Sandor, Y. Lin, X. Li, F. Lin, and S. Zhang, "Efficient decentralized multi-authority attribute based encryption for mobile cloud data storage," *J. Netw. Comput. Appl.*, vol. 129, pp. 25–36, Mar. 2019.

[26] J. Li, N. Chen, and Y. Zhang, "Extended file hierarchy access control scheme with attribute based encryption in cloud computing," *IEEE Trans. Emerg. Topics Comput.*, to be published, doi: 10.1109/tetc.2019.2904637.

[27] J. Li, X. Lin, Y. Zhang, and J. Han, "KSF-OABE: Outsourced attribute-based encryption with keyword search function for cloud storage," *IEEE Trans. Serv. Comput.*, vol. 10, no. 5, pp. 715–725, Sep. 2017.

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

[29] J. Li, Q. Yu, and Y. Zhang, "Hierarchical attribute based encryption with continuous leakage-resilience," *Inf. Sci.*, vol. 484, pp. 113–134, May 2019.

[30] B. Qin, R. H. Deng, S. Liu, and S. Ma, "Attribute-based encryption with efficient verifiable outsourced decryption," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 7, pp. 1384–1393, Jul. 2015.

[31] J. Xu, Q. Wen, W. Li, and Z. Jin, "Circuit ciphertext-policy attribute-based hybrid encryption with verifiable delegation in cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 1, pp. 119–129, Jan. 2016.

[32] S. Garg, C. Gentry, and S. Halevi, "Candidate multilinear maps from ideal lattices," in *Proc. EUROCRYPT*, in Lecture Notes in Computer Science, vol. 7881, T. Johansson and P. Q. Nguyen, Eds. Berlin, Germany: Springer, 2013, pp. 1–17.

[33] Y. Hu and H. Jia, "Cryptanalysis of GGH map," in *Proc. EUROCRYPT*, in Lecture Notes in Computer Science, vol. 9665. Berlin, Germany: Springer, 2016, pp. 537–565, doi: 10.1007/978-3-662-49890-3_21.

[34] J. Baliga, R. W. A. Ayre, K. Hinton, and R. S. Tucker, "Green cloud computing: Balancing energy in processing, storage, and transport," *Proc. IEEE*, vol. 99, no. 1, pp. 149–167, Jan. 2011.

[35] S. T. Selvi and C. Valliyammai, "Dynamic resource allocation with efficient power utilization in cloud," in *Proc. 6th Int. Conf. Adv. Comput. (ICoAC)*, Dec. 2014, pp. 302–307, doi: 10.1109/icoac.2014.7229730.

[36] A. T. Saraswathi, Y. R. A. Kalaashri, and S. Padmavathi, "Dynamic resource allocation scheme in cloud computing," *Procedia Comput. Sci.* vol. 47, pp. 30–36, 2015, doi: 10.1016/j.procs.2015.03.180.

[37] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *Proc. 21st Annu. Int. Cryptol. Conf. (CRYPTO)*, Santa Barbara, CA, USA, vol. 2139, Aug. 2001, pp. 213–229.

[38] A. Beimel, "Secure schemes for secret sharing and key distribution," Ph.D. dissertation, Israel Inst. Technol., Haifa, Israel, 1996.

[39] PBC Library. [Online]. Available: http://crypto.stanford.edu/pbc

[40] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in *Proc. PKC*, 2011, pp. 53–70.

[41] E. Fujisaki and T. Okamoto, "Secure integration of asymmetric and symmetric encryption schemes," in *Proc. CRYPTO*, vol. 1666, 1999, pp. 537–554.