Hindawi
Security and Communication Networks
Volume 2021, Article ID 6617669, 11 pages
https://doi.org/10.1155/2021/6617669

Research Article

User Privacy Protection Scheme Based on Verifiable Outsourcing Attribute-Based Encryption

Luo Sheng

School of Information and Statistics, Guangxi University of Finance and Economics, Nanning 530003, China

Correspondence should be addressed to Luo Sheng; 315412323@qq.com

Received 2 December 2020; Revised 18 July 2021; Accepted 7 August 2021; Published 17 August 2021

Academic Editor: Vincenzo Conti

Given the rapid development of cloud computing and cloud storage technology, a growing number of enterprises and individuals use cloud storage services to save data or back up data. In the cloud storage services, although attribute-based encryption can protect users' data security, the computational cost of key generation, data encryption, and data decryption linearly increases with the complexity of access strategies, which becomes more critical for resource-constrained users. Therefore, this paper proposes a verifiable attribute-based encryption scheme for a fully outsourced ciphertext policy. The scheme can simultaneously realize the functions of key generation, data encryption, and data decryption outsourcing and verify the correctness of the outsourcing calculation results. This scheme can effectively reduce the computational burden of a cloud storage system. The security and verifiability of the scheme are indicated and proved with the random oracle model. The experimental results show that the scheme has the advantages of function and efficiency compared with other schemes. The research results have theoretical and practical significance.

1. Introduction

In the era of big data, large amounts of data are rapidly generated from different sources (e.g., smartphones, sensors, and social networks). Traditional computer systems have been unable to store and process these data. The emerging cloud storage technology has been extensively utilized due to its advantages of low cost, large capacity, transparent access, and ability to provide services at any time [1, 2]. With the continuous development of cloud storage technology and mobile terminals, users can store their data in the cloud and rely on cloud servers to share data with other users. However, when data are outsourced to a cloud server, users lose physical control over their data, and the cloud service provider cannot be fully trusted. To solve the data security problem of cloud server users, Lai et al. [3] proposed a verifiable outsourcing decryption scheme, which can effectively verify the correctness of cloud computing results by establishing validation labels. As access policies are appended in plaintext to ciphertext, access policies may reveal user privacy. For example, if Alice encrypts her data to an ophthalmologist, then the attributes that an access policy may contain are ophthalmologists and doctors. Although the data cannot be decrypted, a person that views the data may speculate that Alice has an eye disease, which violates Alice's privacy. To enable end users to effectively control their data access, sensitive data privacy protection mechanisms must be established for cloud storage. According to the literature, the existing cloud storage model has some shortcomings regarding encrypting and decrypting attributes based on outsourcing. The model has low computational efficiency and reliability and cannot adapt to large-scale user access. In this context, this paper proposes an efficient and verifiable fully outsourced CP-ABE scheme. This scheme can simultaneously realize the outsourcing of key generation, encryption, and decryption, verify the correctness of outsourcing calculation results, and build a secure and efficient cloud storage access control model.

2. Related Work

Traditional access control is based on trusted servers. To satisfy the requirements of cloud storage services, a cryptographic mechanism that is based on traditional access control is necessary. Attribute-based encryption (ABE) is a promising technology that cannot only guarantee the

security of data storage but also realize flexible access permission settings [4, 5]. Currently, two classical algorithms exist for the attribute-based cryptosystem. The first algorithm is an attribute encryption algorithm that is based on ciphertext policy ABE (CP-ABE), which embeds access strategy into ciphertext, while the user's private key binds a set of attributes to represent the user's identity [6, 7]. Wang et al. [8] adopted the method [9] and extended most attribute encryption schemes based on a bilinear pairing operation to outsourced attribute encryption schemes. Lai et al. [10] considered the security verification of outsourcing decryption based on [9] and proposed an ABE scheme that supports verifiable outsourcing decryption and proves the security of the scheme with the standard model; however, the efficiency of the scheme is low. Subsequently, Lin et al. [11] systematically solved the security verification problem of outsourcing decryption; the efficiency of the proposed scheme was nearly half higher than that of [10]. In addition, Li et al. [12] used the MapReduce method to implement ABE-encrypted outsourcing computing. This scheme supports a tree access strategy and has a rich expressive ability but does not consider outsourcing decrypted computing. Fan et al. [13] proposed an ABE scheme that supports private key generation and decryption outsourcing, which employs two key generation service providers to help attribute authorization agencies complete the work of private key generation. Li et al. [14] proposed a verifiable outsourced multiauthorization access control scheme, which outsources most of the encryption and decryption tasks to fog nodes to reduce the user's computing burden. The scheme can verify the correctness of the outsourced computing results. Li et al. [15] proposed a new verifiable outsourcing decryption ABE scheme. In this scheme, the length of ciphertext is not related to the complexity of the access policy. However, the scheme only supports the outsourcing of decryption computing, and the expression ability of the access policy is limited. None of these schemes can be completely outsourced, that is, private key generation, encryption, and decryption computing are simultaneously outsourced to third parties. Although Zhang et al. [15] proposed a completely outsourced CP-ABE scheme, which outsources key generation, encryption, and decryption to cloud service providers and completes the security certification of the scheme, the scheme could not verify the correctness of the outsourced calculation results. Ma et al. proposed a new construction of attribute-based encryption (ABE) which can outsource the complicated encryption task to a Encryption Service Provider (ESP) in a verifiable manner [16]. Hu et al. showed that Ma et al.'s proposal fails to provide the verifiability property for outsourced encryption [17]. However, verifiability is very important for cloud storage system applications.

Due to the shortcomings of existing research, this paper proposes a verifiable complete outsourcing CP-ABE scheme. The scheme can realize the outsourcing of key generation, encryption, and decryption and verify the correctness of the outsourcing calculation results. Specifically, attributes authorization agencies employ two collude cloud service providers to generate an intermediate private key ISK_x, such as $x = \{1, 2\}$. The attribute authorization authority can complete the work of private key generation by simple calculation according to ISK_x. This paper introduces the default attribute ξ to reconstruct the access policy to complete the encryption outsourcing work. Using the private key SK, reconstruct the transform key TK and retrieve the key RK. The cloud service providers complete part of the decryption of ciphertext via TK. In addition, two hash functions are used to verify the correctness of the outsourcing calculation results. This scheme can effectively reduce the computational burden of attribute authorization agencies and users. Based on the decision-making q-Bilinear Diffie–Hellman Exponent (q-BDHE) hypothesis, this paper proves the indistinguishable security of the proposed scheme in selecting plaintext attacks with the random oracle model and provides the verifiability proof of the proposed scheme. The theoretical analysis and experimental verification show that the proposed scheme has advantages of functionality and efficiency and is more suitable for practical application.

3. Research Model

The system model of the scheme proposed in this paper is shown in Figure 1.

The core components of the system model include the Key Generation-Cloud Service Provider (KG-CSP), Encryption-Cloud Service Provider (E-CSP), Decryption-Cloud Service Provider (D-CSP), and Storage-Cloud Service Provider (S-CSP). These components provide private key generation services, data encryption services, data decryption services, and data storage services. During the process of service, however, the components cannot know the user's private key and data plaintext.

In this scheme, the data owner (DO) can use a mobile computing terminal to encrypt plaintext information and store it in the cloud. The data user (DU) can use mobile computing terminals to download ciphertext information from the cloud and decrypt it; mobile computing terminals can withstand this computing load.

This paper assumes that the Attribute Authority (AA) is a fully trusted key distribution agency. Cloud service providers are honest but curious, that is, cloud service providers honestly follow the proper steps but due to curiosity will pry into the privacy of data during the course of their work [15]. Two KG − CSP cannot collude with each other to share data; thus, *ISK* obtained is information that is hidden relative to two KG − CSP.

4. Research Design

4.1. Theoretical Basis

4.1.1. Bilinear Group. A bilinear group is an important key technology in the cryptosystem. Let ψ be a group generation algorithm that considers the safety parameter λ as the input and output (p, G, G_T, and e), where p is a prime determined by the safety parameter λ and G and G_T are cyclic groups of the order prime p. Bilinear mapping e: $G \times G \longrightarrow G_T$ satisfies the following properties:

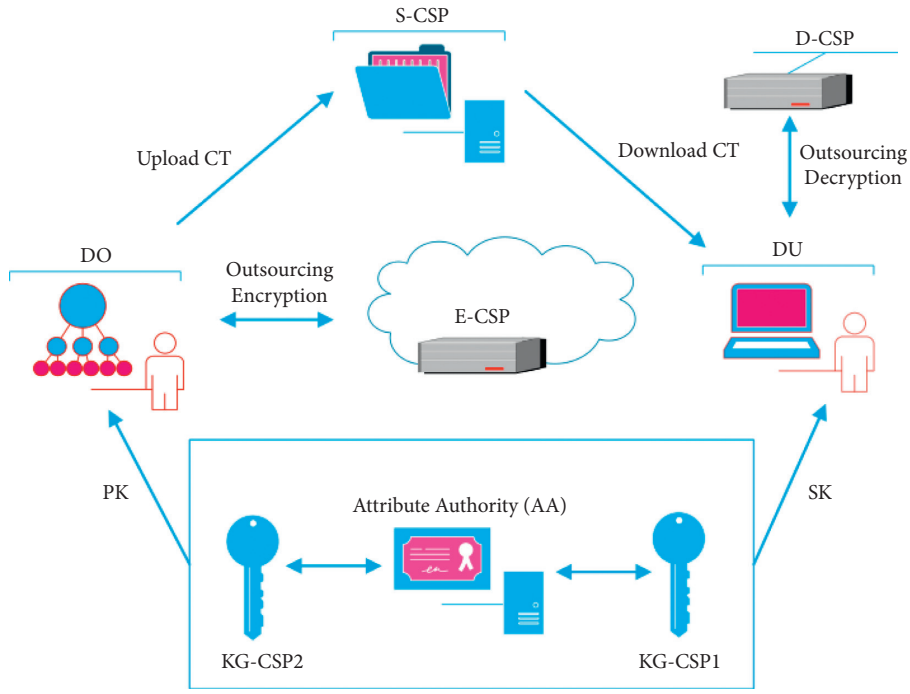

Figure 1: A system model of fully outsourced attribute-based encryption.

(1) Bilinear: $\forall u$ and $v \in G, a$ and $b \in Z_P$, and $e(u^a, v^b) = e(u, v)^{ab}$

(2) Non-degeneracy: g $\in G$ exists such that the order of $e(g, g)$ in G_T is p, $s \in Z_p$

(3) Computability: for $\forall u$ and $v \in G$, $e(u, v)$ can be effectively calculated

4.1.2. Linear Secret Sharing Scheme. The linear secret sharing scheme (LSSS) is defined as follows: if one of the key sharing schemes Π in the participant set p satisfies the following two conditions, then the scheme is referred to as a linear secret sharing scheme on Z_p.

(1) The secret share of each entity constitutes a vector on Z_p.

(2) For each secret sharing scheme Π, a generating matrix $M(l \times n)$ exists. For each row i $(i = 1, 2, \ldots, l)$ in matrix M and mapping ρ: $\{1, 2, \ldots, l\} \longrightarrow p$, map each row in M to participant $\rho(i)$, where ρ is a single projective function. Consider vector $v = (s, y_2, \ldots, y_n)$, where $s \in Z_p$ is a shared key, $y_2, \ldots, y_n \in Z_p$ is a random selection that is used to hide s, and M_v is a vector that consists of l secret shares. $\lambda_i = (Mv_i)$ represents the secret share of participant $\rho(i)$.

The LSSS scheme has a linear reconstruction property. Assume that Π is a linear secret sharing of access policy A. Let $S \in A$ be an access authorization set, which is defined as $I = \{i: \rho(i) \in S\}$. If $\{\lambda_i\}$ is the effective share of secret s, then a set of constants $\{w_i \in Z_p\}_{i \in I}$ can be established in polynomial time, and equation $\sum_{i \in I} w_i \lambda_i = S$ holds.

4.1.3. Decision-Making q-BDHE Hypothesis. Let G denote a bilinear group whose order is prime p, and let g and h be two independent generators of group G. Choose the random value $a \in Z_P^*$, and define $y_{g,a,l} = (g_1, g_2, \ldots, g_l, g_{l+2}, \ldots, g_{2l}) \in G^{2l-1}$ among $g_i = g^{(\alpha^i)}$. The algorithm guesses by the output value $z \in \{0, 1\}$.

If $|p_r[B(g, h, y_{g,\alpha,l}, e(g_{l+1}, h)) = 0] - p_r[B(g, h, y_{g,\alpha,l}, Z) = 0]| \geq \varepsilon$, then the dominant ε is defined to solve the decision-making q-BDHE problem for group G and G_T. If there are no polynomial time algorithm existing, which cannot be disregarded to solve the decision-making q-BDHE problem, then we conclude that the decision-making q-BDHE hypothesis is valid in groups G and G_T.

4.2. Hybrid Access Policy. This paper proposes a verifiable full outsourcing CP-ABE scheme based on Waters' CP-ABE scheme [18]. In this scheme, the user's private key is associated with the attribute set S, and the ciphertext is associated with the access policy (M, ρ). To ensure the confidentiality of data in the process of encrypting outsourcing, this paper establishes the hybrid access policy $Str = (M, \rho) \wedge \{\xi\}$, where "$\wedge$" represents the "AND" gate, (M, ρ) represents the original access policy, and ξ represents the default attribute. For any given access policy T, the hybrid access policy $Str = (M, \rho) \wedge \{\xi\}$ in this paper is constructed by introducing the default attribute ξ into the original access policy (M, ρ) using the "AND" gate. By this ingenious construction, the original access policy can be arbitrary. In the process of encryption, the data owner completes ξ encryption, and E-CSP completes (M, ρ) encryption without leaking plaintext information.

4.3. Encryption Scheme. The data security access control scheme in this article is as follows:

(1) Setup (1^{λ}): the algorithm chooses the bilinear group G, whose order is prime p, g is the generator of group G, and $h_{\xi}, h_1, \ldots, h_U \in G$ is a random group element. In addition, the exponent α and $\beta \in Z_p$ is randomly selected and $g_1 = g^{\beta}$. Select the hash function H_0: $\{0,1\}^{2\lambda} \longrightarrow \{0,1\}^*$, H_1: $\{0,1\}^* \longrightarrow Z_p$, and H_2: $\{0,1\}^* \longrightarrow \{0,1\}^{\lambda}$. The system master-private key $MSK = \langle g \rangle^a$ and the system public key $PK = \langle G, g, g_1, e(g,g)^{\alpha}, h_{\xi}, h_1, \ldots, h_U, H_0, H_1, H_2 \rangle$ are output.

(2) Key $\mathrm{Gen}_{init}(PK, N)$: the algorithm randomly chooses exponent r' and calculates $D' = g^{\beta r'}$ and $L' = g^{r'}$. For $j = 1$ to N, calculate $D'_j = h_j^{r'}$. For $j = \xi$, calculate $D'_{\xi} = h_{\xi}^{r'}$. The output intermediate key is $ISK_x = (D', L', \{D'_j\}_{j \in [1,N] \cup \{\xi\}})$, such as $x = \{1, 2\}$.

(3) Key $\mathrm{Gen}_{\mathrm{package}}(MSK, S, ISK_1, ISK_2)$: the algorithm uses the master-private key MSK, attribute set $S = \{\mathrm{att}_1, \mathrm{att}_2, \ldots, \mathrm{att}_k\}$, and two independent sets $ISK_1 = (D', L', \{D'_j\}_{j \in [1,k] \cup \{\xi\}})$ and $ISK_1 = (D', L', \{D'_j\}_{j \in [1,k] \cup \{\xi\}})$ as input. The algorithm calculates $L = L' \cdot L'' = g^{(r'+r'')} = g^r$, $\overline{D} = D' \cdot D'' = g^{\beta(r'+r'')} = g^{\beta r}$, and $D_j = D'_j \cdot D''_j = h_j^{(r'+r'')} = h_j^r$, which implies that $r = r' + r''$. Obtain $ISK = (\overline{D}, L, \{D_j\}_{j \in [1,k] \cup \{\xi\}})$. $D = \overline{D} \cdot g^{\alpha} = g^{\alpha} g^{\beta r}$ is computed, and the private key $SK = \langle D, L, \{D_j\}_{j \in s \cup \{\xi\}} \rangle$ associated with attribute set S is output.

(4) Key $\mathrm{Blind}(SK)$: the algorithm chooses a random value $\delta \in Z_p$ and calculates $\hat{D} = D^{\delta}$ and $\hat{L} = L^{\delta}$. For $j \in S \cup \{\xi\}$, calculate $\hat{D}_j = D_j^{\delta}$. The retrieve key $RK = \delta$ and the conversion key $TK = \langle \hat{D}, \hat{L}, \{\hat{D}_j\}_{j \in s \cup \{\xi\}} \rangle$ are output.

(5) $\mathrm{Encrypt}_{\mathrm{init}}(m)$: the algorithm randomly chooses $R \in G_T$, calculates $s = H_1(R, m)$, and randomly specifies a first-order polynomial $q(\cdot)$, such as $q(0) = s$. The algorithm further sets $s_1 = q(1)$ and $s_1 = q(2)$. The encryption keys $EK_{E-CSP} = \{s_1\}$ and $EK_{DO} = \{s, s_2, R\}$ are output.

(6) $\mathrm{Encrypt}_{E-CSP}(PK, (M, \rho), EK_{E-CSP})$: The algorithm takes PK, (M, ρ), EK_{E-CSP} as input parameters. M is a $l \times n$ matrix. Function ρ is a projective function that maps every row of M to an attribute. The algorithm randomly chooses vector $v = (s_1, y_2, \ldots, y_n) \in Z_p$, which is used to encrypt index s_1. For $i = 1$ to l, calculate $\lambda_i = (vM)_i$. M_i is the i line of the matrix M. The output middle ciphertext is $CT_{E-CSP} = \langle C' = g^{s_1}, \{C_i = g^{\beta\lambda_i} h_{\rho(i)}^{-s_1}\}_{1 \le i \le l} \rangle$.

(7) $\mathrm{Encrypt}_{DO}(PK, (M, \rho), EK_{DO}, CT_{E-CSP}, m)$: the algorithm obtains $CT_{DO} = \langle C, C'', C'_{\xi}, C_{\xi} \rangle$ and verification flag $VK_m = H_0(t \| C'')$ by calculating $t = H_2(R)$, $C = R \cdot e(g,g)^{\alpha s}$, $C'' = m \oplus t$, $VK_m = H_0(t \| C'')$, $C'_{\xi} = g^{s_2}$, and $C_{\xi} = g^{\beta s_2} h_{\xi}^{-s_2}$. DO outputs the ciphertext $CT = \langle CT_{DO}, CT_{E-CSP} \rangle$ and then sends VK_m and CT to $S - CSP$.

(8) $\mathrm{Decrypt}_{D-CSP}(TK, CT)$: assume that the set of attributes $S \cup \{\xi\}$ that is associated with the user's private key satisfies the mixed access policy $Str = (M, \rho) \wedge \{\xi\}$, which is associated with the ciphertext CT. The subscript set $I \subseteq \{1, 2, \ldots, l\}$ of participants is defined as $I = \{i: \rho(i) \in S\}$. If $\{\lambda_i\}$ is an effective share of secret S_1, then the set of constants $\{w_i \in Z_p\}_{i \in I}$ exists in polynomial time; thus, $\sum_{i \in I} w_i \lambda_i = S_1$. We note that several different ways to choose w_i may exist to satisfy this formula. In addition, the decryption algorithm only needs to know M and I to determine these constants. DU sends TK to $D - CSP$. $D - CSP$ is calculated according to the following formula:

$$T\prime = \frac{e\left(C', \hat{D}\right)}{\prod_{i \in I}\left(e\left(C_i, \hat{L}\right) e\left(\left(C', \hat{D}_{\rho(i)}\right)\right)\right)^{w_i}} = \frac{e\left(g^{s_1}, \left(g^{\alpha} g^{\beta r}\right)^{\delta}\right)}{\prod_{i \in I}\left(e\left(g^{\beta\lambda_i} h_{\rho(i)}^{-s_1}, g^{r\delta}\right) e\left(g^{s_1}, h\right)_{\rho(i)}^{r\delta}\right)^{w_i}} = \frac{e(g,g)^{\alpha s_1 \delta} e(g,g)^{\beta r s_1 \delta}}{\prod_{i \in I} e(g,g)^{\beta r \delta \lambda_i w_i}},$$

$$T\prime\prime = \frac{e\left(C'_{\xi}, \hat{D}\right)}{e\left(\left(C_{\xi}, \hat{L}\right)\right) e\left(C'_{\xi}, \hat{D}_{\xi}\right)} = \frac{e\left(g^{s_2}, \left(g^{\alpha} g^{\beta r}\right)^{\delta}\right)}{e\left(g^{\beta s_2} h_{\xi}^{-s_2}, g\right)^{r\delta} e\left(g^{s_2}, h_{\xi}^{r\delta}\right)} = e(g,g)^{\alpha s_2 \delta}. \tag{1}$$

We can calculate and obtain $T = e(g,g)^{\alpha s \delta}$. The output converted the ciphertext $TC = \langle C, C'', T \rangle$. $D - CSP$ sends the converted ciphertext TC to DU.

(9) $\mathrm{Decrypt}_{DU}(TC, VK_m, RK)$: after DU receives TC, it calculates $R = C/(T)1/\delta$ and $t = H_2(R)$. If $H_0(t \| C'') \neq VK_m$, output terminator $\perp$. Otherwise,

calculate $m = C'' \oplus t$ and $s = H_1(R, m)$. If $C = R \cdot e(g, g)^{\alpha s}$, then $T = e(g, g)^{\alpha s \delta}$ outputs m; otherwise, it outputs terminator $\perp$.

4.4. Safety Detection

4.4.1. Scenario Hypothesis. This paper considers an adversary: adversary A is a malicious user that can conspire with $KG - CSP_x$ (x can only be 1 or 2), $E - CSP$, $D - CSP$, and $S - CSP$. It can obtain user information, including ISK_x (x can only be 1 or 2) of $KG - CSP_x$, $E - CSP$ encryption key EK_{E-CSP}, intermediate ciphertext CT_{E-CSP}, the conversion key TK of $D - CSP$, and the ciphertext CT of $S - CSP$. In this paper, the author assumes that $x = 1$, and then, A attempts to decrypt the ciphertext of other normal users.

Choose the plaintext attack security game. The scheme proposed in this paper describes the Chosen Plaintext Attack (CPA) security game, which is described as follows:

(1) System initialization: the access strategy (M^*, ρ^*) that adversary A will challenge is transmitted to emulator B

(2) System establishment: B executes the Setup algorithm and sends PK to rival A

Query phase 1: simulator B initializes empty table T_0, empty set E, and integer $j = 0$. Rival A can repeat any of the following queries on attribute set S.

(i) Create(S) : simulator B sets j: $= j + 1$, runs the $\text{Ken Gen}_{\text{init}}$ algorithm twice to obtain the intermediate private keys ISK_1 and ISK_2, runs $\text{Ken Gen}_{\text{package}}$ to obtain the private key SK of the associated attribute set S, and runs the Key Blind algorithm to obtain the conversion key TK and retrieve the key RK. $(j, S, SK, TK, RK, ISK_1)$ is stored in table T_0.

Note: an adversary can repeatedly ask for the same set of attributes, such as $f((M^*, \rho^*), S) \neq 1$. However, A can submit the set of attributes $S\prime$, which satisfies (M^*, ρ^*) for the Corrupt ISK_1 query.

(ii) Corrupt $SK(i)$: B verifies whether the number i entity (i, S, SK) exists in table T_0. If this entity exists, set E: $= E \cup (S)$ and return SK; otherwise, return terminator $\perp$.

(iii) Corrupt $ISK_1(i)$: simulator B verifies whether the number i entity (i, S, ISK_1) existing in table T_0. If this entity exists, return ISK_1; otherwise, return terminator $\perp$.

(iv) Corrupt $TK(i)$: B verifies whether the number i entity (i, S, TK) exists in table T_0. If this entity exists, return TK; otherwise, return terminator $\perp$.

Challenge phase: rival A submits two equal-length messages m_0 and m_1, and then, emulator B randomly chooses $b \in \{0, 1\}$. Based on the challenge access policy (M^*, ρ^*) and plaintext message m_b, after running $\text{Encrypt}_{\text{init}}$, the encrypted key pair $(EK^*_{E-CSP}, EK^*_{DO})$ is obtained. Run Encrypt_{E-CSP} to obtain the intermediate ciphertext CT^*_{E-CSP}. Run Encrypt_{DO} to obtain the ciphertext CT^* and verification flag VK^*_m of the plaintext message m_b. B sends EK^*_{E-CSP}, CT^*, and VK^*_m to rival A.

Query phase 2: similar to query phase 1, rival A continues to submit a list of attributes to simulator B.

Guessing stage: rival A outputs the value $b' \in \{0, 1\}$ as a guess of b. If $b' = b$, we consider that rival A won the game. The advantage of rival A in the game is defined as follows: $Adv^{CPA}_A(\lambda)$: $= |[Pr[b' = b] - 1/2]|$.

Definition 1. If no polynomial time exists, the rival uses the advantages that cannot be disregarded to attack the security model. We note that the objective of the proposed scheme is to choose plaintext security.

Verifiability game: verifiability ensures correct execution during the transformation phase. The verifiability of the proposed scheme is described by the game between simulator B and rival A. The specific process is detailed as follows:

System establishment: simulator B executes the Set up algorithm, sends PK to rival A, and retains the master-private key MSK.

Query phase 1: B responds to the query of rival A according to the way of private key generation. Because B knows MSK, it can answer all private key queries.

Challenge phase: rival A submits the plaintext m^* and access policy (M^*, ρ^*). Simulator B runs $\text{Encrypt}_{\text{init}}$ to obtain the encryption key EK^*_{E-CSP}. Run Encrypt_{E-CSP} to obtain the intermediate ciphertext CT^*_{E-CSP}. Run Encrypt_{DO} to obtain (CT^*, VK^*_m). B sends them to rival A.

Query phase 2: B responds to the query from rival A by the way of query phase 1. However, rival A cannot ask for the set S of attributes that satisfy the access policy (M^*, ρ^*).

Guessing stage: the output of rival A satisfies S^* and TC^* of $f((M^*, \rho^*), S^*) = 1$. If $\text{Decrypt}_{DU}(TC^*, RK_{s^*}, VK^*_m *) \notin \{m^*, \perp\}$, then A wins the game. A's advantage in the game is defined as follows: $\text{Adv}^{\text{Ver}}_A(\lambda)$: $= \Pr[A \text{ Wins}]$.

Definition 2. If no polynomial time exists, the rival uses the advantages that cannot be disregarded to attack the security model. We note that the scheme proposed in this paper is verifiable.

4.4.2. Proof of Safety

Theorem 1. *Assuming that the decision-making q-BDHE hypothesis is valid for groups G and G_T, then the scheme proposed in this paper is selective CPA security with the random oracle model.*

Prove: assuming that a polynomial time exists, rival A can have the nonnegligible advantage ε and break through the scheme using the selective CPA security model. We can construct the simulator B, which has a nonnegligible advantage for solving the decision-making q-BDHE problem. Rival A is a malicious user and can conspire with $KG - CSP_x$ (x can only be 1 or 2), $E - CSP$, $D - CSP$, and $S - CSP$. This paper assumes that two $KG - CSP$ cannot

collude with each other to share data, while ISK is calculated by ISK_1 and ISK_2. ISK information is hidden from the perspective of all rivals A. This paper assumes that $x = 1$ and that rival A attempts to decrypt the ciphertext of other normal users. Therefore, rival A can submit to satisfy attribute set $S\prime$ of (M^*, ρ^*) for the ISK_1 query but does not obtain any useful information about it.

B inputs the decision q-BDHE challenge tuple $(g, h, y_{g,\alpha,l}$, and $Z)$, where Z is a random element or $e(g_{l+1}, h)$ and $y_{g,\alpha,l} = (g_1, g_2, \ldots, g_l, g_{l+2}, \ldots, g_{2l}) \in G^{2l-1}$ in group G_T.

System initialization: rival A chooses the access strategy $T^* = (M^*, \rho)^*$, which needs to be challenged, and sends it to the emulator B.

Establish system: B calculates $PK = \langle G, g, g_1 = g^a, e(g, g)^a, h_1, \ldots, h_U, h_\xi \rangle$ by the way of challenger C in [18], and then, emulator B sends the public key PK to rival A.

Query phase 1: B initializes empty tables T_0, T_1, and T_2, empty set E, and integer $j = 0$. Rival A can repeat any of the following queries on the set of attributes.

(1) Random Oracle Hash $H_1(R, m)$. If the entity (R, m, s) exists in table T_1, then it returns s. Otherwise, select the random value $s \in Z_p$, record (R, m, s) in table T_1, and return s.

(2) Random Oracle Hash $H_2(R)$. If the entity (R, t) exists in table T_2, then it returns t. Otherwise, select the random value $t \in \{0, 1\}^\lambda$, record (R, t) in table T_2, and return t.

(3) Creat(S). After B receives the private key query of attribute set s from rival A, the default attribute ξ is added to the attribute set, that is, the private key query set is $S \cup \{\xi\}$. B sets $j{:}\ = j + 1$ and calculates and obtains $SK = \langle D, L, \{D_j\}_{j \in S \cup \{\xi\}} \rangle$ according to challenger C in [18]. B runs Ken Gen_{init} to obtain ISK_1. Run Key Blind to obtain the conversion key TK and retrieve the key RK. $(j, S, SK, TK, RK, ISK_1)$ is stored in table T_0. Note: A can repeatedly ask the same set of attributes S, where S can submit to satisfy $f((M^*, \rho^*), S) \neq 1$. However, A can submit attribute set $S\prime$ to satisfy (M^*, ρ^*) for the ISK_1 query.

(4) Corrupt $SK(i)$. Simulator B verifies whether the ith entity $(i, S,$ and $TK)$ that exists in table T_0. If it exists, set $E{:}\ = E \cup \{S\}$ and return SK. Otherwise, return terminator $\perp$.

(5) Corrupt $ISK_1(i)$. Simulator B verifies whether the ith entity $(i, S,$ and $ISK_1)$ exists in table T_0. If it exists, return ISK_1. Otherwise, return terminator $\perp$.

(6) Corrupt $TK(i)$. Simulator B verifies whether the ith entity $(i, S,$ and $TK)$ exists in table T_0. If it exists, return TK. Otherwise, return terminator $\perp$.

Challenge phase: rival A submits two plaintext messages of equal length m_0 and m_1. B randomly selects "message" $(R_0, R_1) \in G_T$ and $b \in \{0, 1\}$. According to the method of C in [18], the ciphertext $CT_w = \langle \overline{C}, C', \{C_i\}_{i \in [1,j]} \rangle$ (replace s with s_1 in [18]; $\overline{C}$ is equivalent to C) of the plaintext R_b correlation (M^*, ρ^*) is obtained. B calculates $s = H_1(R_b, m_b)$ and $t = H_2(R_b)$, sets $s_2 = s - s_1$, and calculates $C'_\xi = g^{s_2}$, $C_\xi = g^{\beta s_2} h_\xi^{-s_2}$, and $e(g^{s_2}, g^{\alpha'})$. Simulator B calculates $C'' = m_b \oplus t$, $VK^*_m = H_0(t \| C'')$, and $C = \overline{C} \cdot e(g^{s_2}, g^{\alpha'})$. Simulator B sends $CT^* = \langle C, C'', C'_\xi, C_\xi, C', \{C_i\}_{i \in [1,l]} \rangle$, $EK^*_{E-CSP} = \{s_1\}$, and $VK^*_m = H_0(t \| C'')$ to rival A.

Query phase 2: similar to query phase 1, rival A continues to submit a list of attributes to B.

Guessing stage: rival A outputs the value $b' \in \{0, 1\}$ as a guess of b. If $b' = b$, B output 0 indicates the guess $Z = h(g_{n+1}, h)$. Otherwise, output 1 indicates the guess that Z is a random element in group G_T. When $Z = e(g_{n+1}, h)$, simulator B can provide an effective simulation. Accordingly, we conclude that $Pr[B(g, h, y_{g,\alpha,l}, e(g_{l+1}, h)) = 0] = 1/2 + Adv_A$. When Z is a random element in G_T, m_b is completely random for A. Accordingly, we conclude that $P[B(g, h, y_{g,\alpha,l}, Z) = 0] = 1/2$. Therefore, B has the advantage that it cannot be disregarded to attack the decision-making q-BDHE hypothesis.

The proof is complete.

4.4.3. Proof of Verifiability

Theorem 2. *Assume that H_0 and H_2 are hash functions against a collusion attack. The proposed scheme is verifiable.*

Prove: assuming that rival A can break through the verifiability, simulator B can be constructed to break down the anticollusion ability of the underlying hash functions H_0 and H_2. Rival A submits two challenge hash functions (H^*_0, H^*_2). The simulation process of B is described as follows:

Establish system: B executes the Setup algorithm to obtain the public key PK and primary private key MSK. H^*_0 and H^*_2 are used to replace the hash functions in the public key PK. Note that B knows the primary private key MSK.

Query phase 1: B answers the query of rival A according to the adaptability of the scheme algorithm.

Challenge phase: rival A submits the challenge plaintext m^* and access strategy (M^*, ρ^*). Simulator B calculates the ciphertext $CT^{R*} = \langle C, C', C_i, C'_\xi, C_\xi \rangle$ of the random value $R^* \in M$ and then calculates $t^* = H^*_2(R^*)$, $C'^* = m^* \oplus t^*$, and $VK^*_m = H^*_0(t^* \| C''^*)$. B sends $CT^* = \langle CT^{R*}, C''^* \rangle$ and VK^*_m to rival A. Retain VK^*_m and $(R^*, C\prime\prime^*)$.

Query phase 2: B responds to the query of rival A in the way of query phase 1 but adversary A cannot query the attribute set S that satisfies the access policy (M^*, ρ^*).

Guessing stage: A outputs the set of attributes S^* $((f(M^*, \rho^*), S) = 1)$ and the transformation ciphertext $TC = \langle C, C'', T \rangle$.

If rival A breaks through the verifiability, then simulator B will restore plaintext $m \notin \{m^*, \perp\}$ by $Decrypt_{DU}(TC^*, RK_{s^*}, VK^*_m *)$. Analyse the potential success of rival A. If $H^*_0(t \| C'') \neq VK^*_m$, the decryption algorithm outputs the terminator $\perp$, such as $t = H^*_2(R)$ and $R = C/T^{1/RK_{S^*}}$. Therefore, we only need to consider the following two situations:

Situation 1: $(t, C'') \neq (t^*, C''^*)$. As simulator B knows $(t^*, C\prime\prime^*)$, B immediately obtains the collision of hash function H^*_0.

Situation 2: $(t, C'') = (t^*, C''^*)$ *but* $R \neq R^*$. $H_2^*(R) = t = t^* = H_2^*(R^*)$ *will break the ability of* H_2^* *to resist a collusion attack.*

Based on this analysis, the security proof of Theorem 2 is completed. The proof is complete.

5. Experiments and Results

5.1. Theoretical Analysis. To evaluate the computational efficiency of the scheme proposed in this paper, the computational overhead in the stages of private key generation, encryption, and decryption is theoretically analysed. The computational efficiency of this scheme is compared with that of the ABE scheme in [8, 13–15, 18]. In the process of the comparison, $|U|$ represents the number of all attributes in the system, $|S|$ denotes the number of attributes of DU, s represents a set of attributes that satisfies the decryption requirements, and l represents the number of rows in matrix M in $LSSS$. In addition, E_G and E_{GT} represent modular exponential operations in G and G_T, respectively. P represents bilinear pairing operations. To compare the fairness, [13] is assumed to have only one AA. Table 1 shows the efficiency comparison of each scheme. Wang et al. [8] used offline or online technology in the encryption phase. To facilitate the comparison, other outsourcing technologies are compared.

The comparison of calculation efficiency of each scheme is shown in Table 1. The work by Waters [18] is a CP-ABE-based scheme, which is also a solution without outsourcing. In [18], attribute authorization agencies, data users, and data owners need to compute a large number of pairwise and exponential operations. And, this paper proposes the author's scheme based on [18]. The scheme achieves a verifiable and complete outsourcing function. This scheme can reduce the computational burden of AA, DO, and DU and considerably ease the computational burden of computing resource-constrained terminals. Li et al. [14] only support outsourced decryption calculation and can verify the correctness of the calculation results. Although the work [14] does not support the outsourcing computing function of key generation and encryption, it achieves the constant length of ciphertext and requires less computation in the key generation and encryption stage. The disadvantage is that [14] only supports the "AND" gate access strategy and has limited expressive ability. Wang et al. [8] support offline/online encryption and decryption outsourcing but the scheme does not support validation of the correctness of outsourcing decryption, and AA requires a large number of exponential operations. Kai et al. [13] support encryption and decryption outsourcing and verify the correctness of the calculation results but their AA requires numerous exponential operations. Rui et al. [15] and the author's scheme realize the outsourcing functions of key generation, encryption, and decryption. However, Rui et al. [15] does not support verifiability and cannot guarantee the correctness of the calculation results.

The comprehensive analysis shows that only the author's scheme achieves the outsourcing computing functions of key generation, encryption, and decryption, reduces the computing load of the terminal, and supports verifiability. Outsourcing computing is important for mobile devices with limited electricity and computing resources. Therefore, the author's scheme is effective and practical.

5.2. Experimental Analysis. Via theoretical analysis, the author's scheme has advantages of function and efficiency. To further evaluate the actual performance of the author's scheme, the author tested the computational time of [15] and the author's scheme in terms of private key generation, data encryption, and data decryption by the following experimental environments. The experimental environment configuration is shown in Table 2.

In the CP-ABE scheme, access complexity affects the time of encryption and decryption. To illustrate this point, this paper uses $(S_1 \text{ AND } S_2 \text{ AND} \ldots S_n)$'s access strategy to simulate the most complex situation, where each S_i is an attribute. This method ensures that all ciphertext components are involved in decryption computation. In this form, each time increment is 10, from 10 to 100, which produces 10 different access strategies. For each access strategy, 20 experiments are repeated, each experiment is completely independent, and the average is considered the experimental result.

Attribute-based encryption usually cooperates with symmetric encryption to encrypt plaintext data, that is, encrypt plaintext with the symmetric key, and then encapsulates the symmetric key with attribute-based encryption. Therefore, to obtain benchmark results, a 128 bit symmetric key is encapsulated based on the previously mentioned access strategy. The experimental results are shown in Figure 2.

Figure 2 has three subgraphs. Each subgraph compares the execution time of the author's scheme and Rui et al.'s [15] scheme.

Figure 2(a) shows that $KG - CSP$ undertakes most of the key generation, and the time of key generation linearly increases with the number of attributes. Attribute authorization agencies only need to undertake a small amount of computation to complete the key generation. In the foregoing section, the author analyses that the calculation amount of AA is 0. The author disregards multiplication and hash operations as they are secondary factors.

Figure 2(b) shows that $E - CSP$ undertakes most of the encryption work, and the encryption time linearly increases with the complexity of the access policy. The data owner only requires a constant amount of computation to complete the encryption work.

Figure 2(c) shows that $D - CSP$ undertakes most of the decryption work, and the ciphertext conversion time linearly increases with the complexity of the access strategy. User decryption only requires constant computation to complete the decryption work, which is unrelated to the complexity of the access policy.

Figure 2 shows that the key generation time, encryption time, and decryption time increase with the complexity of the attribute sets or access policies. By comparing the two schemes, the computing cost of the author's scheme in the

Table 1: Comparative analysis of computational efficiency.

Schemes		Reference [18]	Reference [8]	Reference [14]	Reference [15]	Reference [13]	Author's scheme
Key generation	AA	$(2+\|S\|)E_G$	$(4+\|S\|)E_G$	$3E_G$	0	$(5+2\|S\|)E_G$	0
	KG-CSP				$(4\|S\|+3)E_G$		$(2\|S\|+6)E_G$
Encryption	DO	$(2l+1)E_G + 1E_{G_T}$	$2E_G + E_{G_T}$	$6E_G + 2E_{G_T}$	$1E_{G_T}$	$1E_{G_T}$	$3E_G + 1E_{G_T}$
	E-CSP		$3\|U\|E_G$		$(5l+1)E_G$	$(3l+1)E_G$	$(2l+1)E_G$
Decryption	DU	$sE_{G_T} + (2s+1)p$	$1E_{G_T}$	$2E_G + 2E_{G_T}$	$1E_{G_T}$	$1E_{G_T}$	$3E_{G_T}$
			$sE_G + sE_{G_T} + (3s+2)p$	$4p$	$(2s+2)E_G + sE_{G_T} + (3s+2)p$	$(3s+1)E_G + sE_{G_T} + (2s+1)p$	$sE_{G_T} + (2s+4)p$
Verifiability	D-CSP	No	No	Yes	No	Yes	Yes
Outsource		No	Yes	Yes	Yes	Yes	Yes

Table 2: Configuration of the experimental environment.

Name		Parameter	Remarks
Hardware	CPU	Intel(R) CoreTM i5-2400 @ 3.10 GHZ	8 M cache
	Memory	DDR3 1333/1600 MHz, 4G	
I/O reading speed 2 M	Hard disk	256 GB PCIe × 4 NVMe SSD	
	Network	Guangzhou Telecom	
Software	Operating system	Ubuntu 12.04	Nothing
	PBC Library	PBC-0.5.14	
	Encryption mode	OpenSSL-1.0.0c, 128Bit, AES	

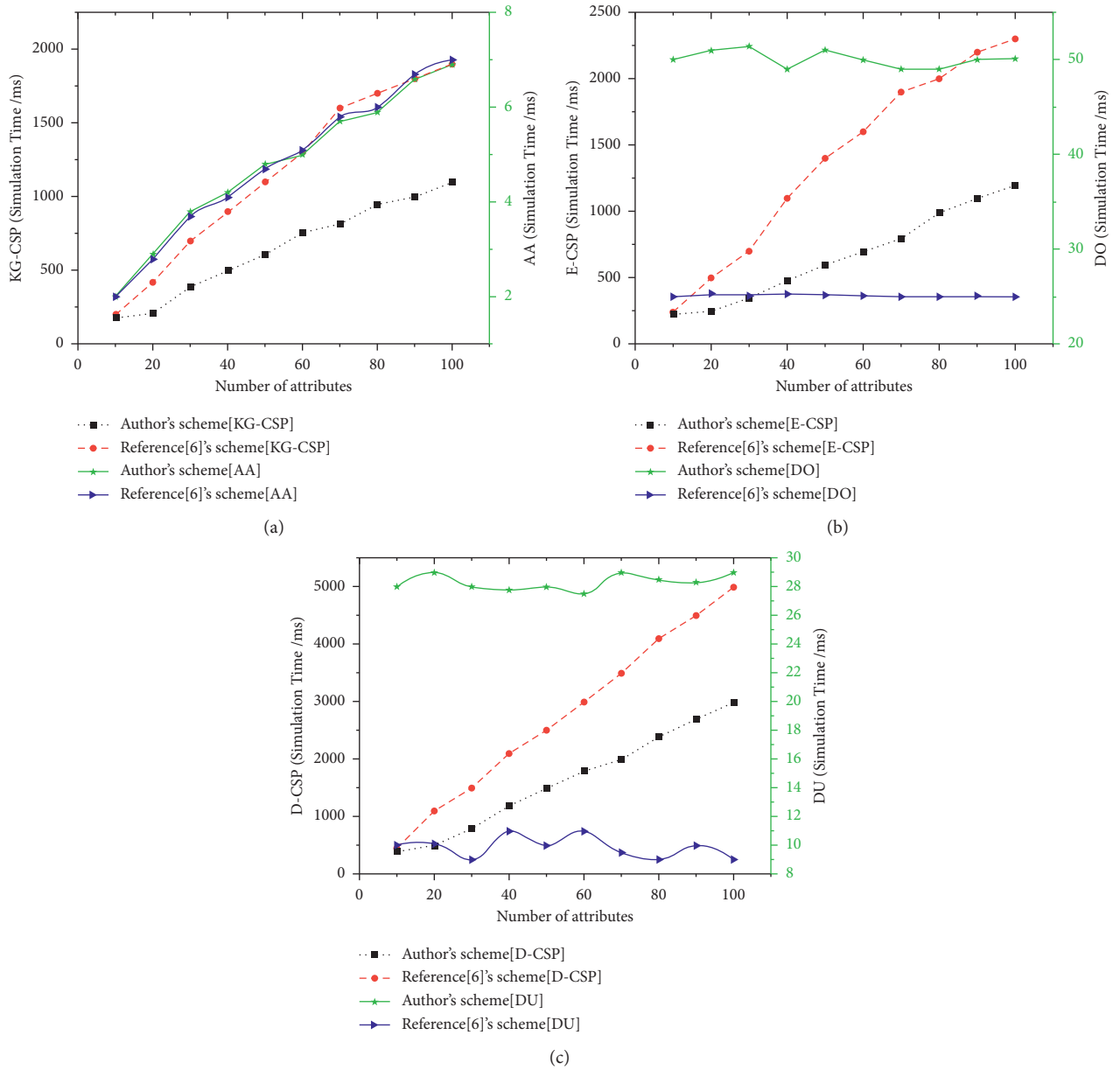

Figure 2: Comparison of simulation time. (a) Key generation of KG-CSP and AA. (b) Encryption of E-CSP and DO. (c) Decryption of D-CSP and DU.

cloud is less than that of [15]. This advantage becomes more distinct with an increase of the number of attributes or the complexity of the access strategy. Figure 2(a) illustrates that both [15] and *AA* of the author's scheme require less computation to complete the key generation, and the efficiency is comparable. Figures 2(b) and 2(c) show that *DO*

and *DU* in [15] require less computation than the author's scheme. However, this gap is very small and does not change with the number of attributes or the complexity of the access policies.

The computation of the author's scheme in the cloud is less than that in [15] and becomes more distinct with an increase in the number of attributes or the complexity of the access strategy, which helps *AA* and enables users to rent less cloud computing resources and reduce costs. The local computational complexity of the author's scheme is slightly higher than that of [15]. However, this gap is very small and does not change with an increase in the number of attributes or the complexity of the access strategy. In addition, the author's scheme supports the verifiability of decryption outsourcing, which is not available in [15].

6. Concluding Remarks

To improve the efficiency of the CP-ABE scheme, the author proposes a verifiable full outsourcing scheme. The scheme can simultaneously realize the outsourcing function of key generation, encryption, and decryption calculation and verify the correctness of the outsourcing calculation results. This scheme can effectively alleviate the computational burden of attribute authorization agencies, data owners, and data users, especially for cloud storage systems with a large number of users and users with limited resources, and the advantages are more distinct. With the random oracle model, the indistinguishable security of the author's scheme against a plaintext attack and the verifiability of the author's scheme are proved. The theoretical analysis and experimental verification show that the author's scheme has advantages of functionality and efficiency and is more suitable for practical application [19–24].

Data Availability

The data used to support the findings of the study are included within the article.

Conflicts of Interest

The author declares that there are no conflicts of interest regarding the publication.

Acknowledgments

This research work was supported by the Middle-Age and Young Teachers' Basic Ability Promotion Project of Guangxi (2019KY0659) and Central Government Guides the Local Science and Technology Development Fund Project Subsidization (Guike AD20238072).

References

[1] C. Feng, Z. Qin, and D. Yuan, "Techniques of secure storage for cloud data," *Chinese Journal of Computers*, vol. 38, no. 1, pp. 150–163, 2015.

[2] D.-G. Feng, M. Zhang, Y. Zhang, and Z. Xu, "Study on cloud computing security," *Journal of Software*, vol. 22, no. 1, pp. 71–83, 2011.

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

[4] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Proceedings of the 24th Annual International Conference on Theory and Applications of Cryptographic Techniques*, pp. 457–473, Springer, Aarhus, Denmark, May 2005.

[5] J. S. Su, D. Cao, X. F. Wang et al., "Attribute-based encryption schemes," *Journal of Software*, vol. 22, no. 6, pp. 1299–1315, 2011.

[6] in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 321–334, IEEE Computer Society, Oakland, CA, USA, May 2007.

[7] X. X. Yan and H. Meng, "Ciphertext policy attribute-based encryption scheme supporting direct revocation," *Journal on Communications*, vol. 37, no. 5, pp. 44–50, 2016.

[8] H. Wang, Z. Zheng, L. Wu, and Y. Wang, "Adaptively secure outsourcing ciphertext-policy attribute-based encryption," *Journal of Computer Research and Development*, vol. 52, no. 10, pp. 2270–2280, 2015.

[9] M. Green, S. Hohenberger, and B. Waters, "Outsourcing the decryption of ABE ciphertexts," in *Proceedings of the 20th USENIX Conference on Security*, USENIX Association, Berkeley, CA, USA, August 2011.

[10] G. Li, J. Lai, L. Jun, and Y. Li, "Ciphertext-policy attribute-based encryption with partially hidden access structure and its application to privacy-preserving electronic medical record system in cloud environment," *Security & Communication Networks*, vol. 9, 2016.

[11] S. Lin, R. Zhang, H. Ma, and Y. Wang, "Revisiting attribute-based encryption with verifiable outsourced decryption," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 10, pp. 2119–2130, 2015.

[12] J. Li, C. Jia, J. Li, and X. Chen, "Outsourcing Encryption of Attribute-Based Encryption with mapreduce," in *Proceedings of the The 14th Information and Commu-Nications Security*, pp. 191–201, Hongkong, China, October 2012.

[13] F. Kai, 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, 2017.

[14] J. Li, F. Sha, Y. Zhang, X. Huang, and J. Shen, "Verifiable outsourced decryption of attribute-based encryption with constant ciphertextlLength," *Information Technology*, vol. 2017, Article ID 3596205, 11 pages, 2017.

[15] Z. A. Rui, M. Hui, and L. C. Yao, "Fine-grained access control system based on fully outsourced attribute-based encryption," *Journal of Systems and Software*, vol. 125, pp. 344–353, 2017.

[16] M. Hui, Z. Rui, Z. Wan, Y. Lu, and S. Lin, "Verifiable and exculpable outsourced attribute-based encryption for access control in cloud computing," *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 6, pp. 679–692, 2015.

[17] X. Hu and J. Sun, "Comments on "verifiable and exculpable outsourced attribute-based encryption for access control in cloud computing"" *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 4, pp. 461-462, 2017.

[18] B. Waters, "Ciphertext-policy attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization," in *Proceedings of the International Workshop on Public Key*

Cryptography, Springer Berlin Heidelberg, Barcelona, Spain, March 2008.

[19] J. Bethencourt, A. Sahai, and B. Waters, *Ciphertext-Policy Attribute-Based Encryption*, Springer, New York, NY, USA.

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

[21] A. B. Lewko, T. Okamoto, K. Takashima, A. Sahai, and B. Waters, "Fully Secure Functional Encryption: Attribute-Based Encryption and (Hierarchical) Inner Product encryption," in *Proceedings of the International Conference on Theory & Applications of Cryptographic Techniques*, Orlando, Canada, June 2010.

[22] B. Waters, "Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization," *Lecture Notes in Computer Science*, vol. 2008, pp. 321–334, 2008.

[23] K. Zhang and J. F. Ma, "Adaptively secure multi-authority attribute-based encryption with verifiable outsourced decryption," *Science China Information Sciences*, vol. 59, no. 9, pp. 99–105, 2016.

[24] M. Q. Zhang, W. D. Du, X. Y. Yang, and Y. Hang, "A fully secure KP-ABE scheme in the standard model," *Journal of Computer Research and Development*, vol. 52, no. 8, pp. 1893–1901, 2015.