Efficient Attribute-Based Encryption Schemes for Mobile Cloud Environments (Postprint)

Authors: Fu Yumeng, Sun Lei, Li Zuohui

Date: 2018-06-19T00:00:00+00:00

Abstract

With the proliferation of cloud computing, mobile devices can store and retrieve personal data anytime and anywhere. Attribute-based encryption (ABE) can be utilized to address security issues in mobile cloud data. Current research on ABE for mobile clouds has primarily concentrated on single-authority scenarios, which fails to meet real-world attribute authorization requirements. To address these issues, this paper proposes a novel multi-authority attribute-based encryption scheme. This scheme features no central authority, allows authorization authorities to operate independently without mutual interference, and supports the addition of distributed attributes. It leverages pre-computation and outsourced decryption methods to reduce computational overhead on the user side. The scheme is proven to be statically secure in the random oracle model. Experimental results indicate that when mobile terminals perform data sharing in cloud environments, this scheme can reduce computational overhead on mobile devices by 20%, making it more aligned with practical application scenarios in mobile cloud environments.

Full Text

Preamble

Efficient Decentralized Attribute-Based Encryption for Mobile Cloud Computing

Fu Yumeng, Sun Lei, Li Zuohui (Information Engineering University, Zhengzhou 450001, China)

Abstract: With the proliferation of cloud computing, mobile devices can store and retrieve personal data anytime and anywhere. Attribute-based encryption (ABE) can be employed to address mobile cloud data security concerns. Current research on ABE schemes for mobile cloud environments primarily focuses

on single-authority scenarios, which do not reflect real-world attribute authorization situations. To address these limitations, this paper proposes a novel multi-authority attribute-based encryption scheme that eliminates the need for a central authority, allows attribute authorities to operate independently without mutual interference, and supports dynamic attribute addition. The scheme leverages precomputation and outsourced decryption to reduce computational overhead on the user side. Furthermore, the scheme is proven statically secure under the random oracle model. Experimental results demonstrate that when mobile devices perform data sharing in cloud environments, the proposed scheme can reduce computational overhead on the mobile device side by 20%, making it more suitable for practical mobile cloud application scenarios.

Keywords: mobile cloud computing; attribute-based encryption; outsourcing; multi-authority

0 Introduction

Mobile cloud computing (MCC) [1~3] has dramatically expanded the application scope of cloud computing, enabling users to access powerful computational, storage, and software services from the cloud without temporal or spatial constraints. Major domestic and international enterprises have successively launched mobile cloud computing services, such as Alibaba' s Apsara Mobile released in 2017, Apple' s "MobileMe" service, and Microsoft' s "LiveMesh" . With the proliferation of mobile devices and the rapid development of 5G networks, mobile cloud has become a focal point of attention in recent years. Concurrently, cloud computing security issues have gradually gained prominence. Gartner' s 2017 cloud security maturity curve [4] indicates that protection for data, applications, and workloads in cloud environments is uneven, with data protection for mobile devices being particularly immature. Existing cloud security solutions cannot be directly applied to mobile cloud environments.

Attribute-based encryption effectively addresses the limitations of traditional encryption methods in flexibly sharing data with multiple users and demonstrates excellent performance in cloud data sharing security. However, most proposed schemes are built upon bilinear mapping technology, and the substantial computational cost of bilinear pairings and group exponentiation operations has long been criticized by researchers. In 2011, Green et al. [9] introduced outsourcing techniques into ABE, leveraging the cloud's powerful computational and storage capabilities to address ABE performance bottlenecks by securely outsourcing complex decryption computations, thereby providing a new research direction. Subsequent research has primarily focused on resource-constrained terminal devices, utilizing outsourcing techniques to enable mobile users to share cloud resources similarly to PC users. In 2013, Li et al. [10] proposed a low-complexity multi-authority ABE scheme for mobile users that relied on a semi-trusted intermediary between mobile users and attribute authorities to facilitate information

exchange. In 2014, Hohenberger et al. [11] proposed an online/offline outsourcing ABE scheme that reduces encryption computational overhead through precomputation, cleverly dividing encryption/decryption workload into online and offline phases to enhance user experience.

These studies provided a solid foundation for discussions on mobile cloud environment security. However, it was not until recent years that secure data sharing for mobile cloud environments has been extensively explored. In 2015, Indian scholars Vijay et al. [12] proposed a CP-ABE scheme for mobile cloud environments addressing attribute revocation, supporting multiple attribute authorities working simultaneously. In 2016, Li et al. [13] proposed a lightweight data sharing scheme for mobile cloud environments that delegated most computations to external proxy servers by modifying the access control tree structure while also achieving revocation functionality. Both schemes require a central authority, whose complete trustworthiness cannot be guaranteed. To mitigate security threats from central authority corruption, Lyu et al. [14] proposed a decentralized mobile cloud ABE scheme in 2017 that employed an anonymous key issuance protocol for privacy protection and utilized online/offline techniques and verifiable outsourced decryption to reduce computational overhead while also achieving revocation. Zhao et al. [15] proposed a verifiable outsourced computation mobile cloud CP-ABE scheme that verifies outsourcing results through two hash functions, but the scheme targets single-authority scenarios and cannot address real-world applications requiring multiple authorities to manage different user attributes. De et al. [16] proposed an ABE scheme for mobile cloud environments enabling fast encryption and decryption with attribute decentralization. Li et al. [17] employed a two-factor authentication mechanism to achieve user anonymity and proposed a multi-authority CP-ABE scheme for mobile cloud environments without requiring a CA. Although these schemes address data sharing issues in mobile cloud environments to some extent, they all require determining the number of attributes before system deployment and cannot flexibly add attributes without resetting system parameters.

Additionally, regarding multi-authority scenarios in cloud environments, various solutions have been proposed in literature [19~22]. Yang et al. [18] presented a decentralized revocable scheme supporting arbitrary monotone access structures. Huang et al. [19] proposed a scheme supporting large attribute domains that can arbitrarily add attributes without affecting system parameters. Cui et al. [20] constructed a revocable scheme based on composite-order bilinear groups. Zhang et al. [21] proposed a decentralized scheme requiring only one exponentiation operation for decryption and supporting large attribute domains. However, these schemes all have high performance requirements and overhead for user-side devices, making them unsuitable for mobile cloud environments.

From a security perspective, designing adaptively secure schemes (both composite-order and prime-order) requires sacrificing performance to some extent, which is not ideal for data sharing in mobile cloud environments. While selecting secure schemes cannot meet practical security requirements,

using dual system encryption techniques to prove adaptive security is also not suitable. In 2015, Rouselakis et al. from Waters' team [22] proposed a static security model adapted for multi-authority settings, which better aligns with data sharing scenarios involving multiple authorities and has gained significant recognition in the field. Compared to selectively secure schemes and adaptively secure schemes proven using dual system encryption techniques, statically secure schemes offer satisfactory security and efficiency.

To the best of our knowledge, there is currently no satisfactory solution for fine-grained access control that is both efficient and secure for multi-authority attribute-based encryption in mobile cloud environments: (a) In practical scenarios, different authorities authorize and manage different user attributes, making multi-authority ABE schemes more computationally intensive and complex to manage than single-authority schemes; (b) Existing schemes require presetting attributes and authorities during system initialization, with subsequent additions necessitating global parameter resets, which is not ideal for practical efficiency; (c) Computational overhead in both encryption and decryption phases is proportional to the size of attribute sets or access structures; (d) There remains significant research space regarding the trade-off between security and efficiency. This paper proposes a decentralized CP-ABE scheme that effectively addresses computational issues arising from multi-authority settings, offers improved security performance compared to existing schemes, and is more suitable for solving practical data sharing problems in mobile cloud environments.

This paper proposes a decentralized attribute-based encryption scheme for mobile cloud environments with the following main advantages:

a) Utilizing online/offline and decryption outsourcing techniques to migrate partial computations to the cloud, enabling mobile users to perform only one exponentiation operation each during encryption and decryption, thereby imposing low performance requirements on mobile devices and better aligning with mobile cloud application scenarios;

b) Constructing the scheme based on LSSS (linear secret sharing schemes) access structures to achieve stronger expressiveness and higher efficiency;

c) Introducing a mapping concept during scheme construction to establish a surjection between attributes and authorities, aligning with management realities and reducing key computation overhead for attributes authorized by the same authority; establishing a one-to-one mapping between attributes and group elements to satisfy the need for subsequent attribute addition without affecting overall scheme operation;

d) Proving the scheme statically secure under the random oracle model.

---

1 Related Work

Mobile cloud computing can generally be summarized as a usage and delivery model where mobile terminals obtain required infrastructure, platform, software, and other resources or information services from the cloud on-demand and in an easily scalable manner through wireless networks [5]. Since its proposal in 2010, mobile cloud security has attracted widespread attention from scholars [6~8]: literature [6] proposed a mobile cloud storage keyword search encryption scheme, literature [7] proposed an outsourced attribute-based encryption scheme for mobile cloud medical applications with hidden access structures, and literature [8] focused on the latest work ensuring mobile cloud computing infrastructure security, summarizing the importance of security issues for mobile cloud computing development.

Currently, attribute-based encryption technology provides a better solution for cloud computing security, offering both data security and flexible access control. However, typical ABE schemes cannot be directly deployed in practice because they require bilinear mapping technology, with complex functionalities relying on modular exponentiation or bilinear pairing operations, making efficiency difficult to improve. In 2011, Green et al. [9] introduced outsourcing concepts into ABE, leveraging the cloud' s powerful computational and storage capabilities to address ABE performance bottlenecks by securely outsourcing complex decryption computations, providing new research directions. Subsequent research has primarily focused on terminal resource-constrained devices, utilizing outsourcing techniques to enable mobile users to share cloud resources like PC users. In 2013, Li et al. [10] proposed a low-complexity multi-authority ABE for mobile users that relied on a semi-trusted intermediary between mobile users and attribute authorities to complete information exchange. In 2014, Hohenberger et al. [11] proposed a method to reduce encryption computational overhead in ABE schemes through precomputation—online/offline outsourcing ABE schemes that divide encryption/decryption workload into online and offline phases, cleverly improving user application experience.

The above research provided a good transition for discussions on mobile cloud environment security. It was not until the past two years that secure data sharing for mobile cloud environments has been widely explored. In 2015, Indian scholar Vijay et al. [12] proposed a CP-ABE scheme for mobile cloud environments addressing attribute revocation, supporting multiple attribute authorities working simultaneously. In 2016, Li et al. [13] proposed a lightweight data sharing scheme for mobile cloud environments that delegated most computations to external proxy servers by modifying the access control tree structure while also achieving revocation functionality. Both schemes require a central authority, whose complete trustworthiness cannot be guaranteed. To mitigate security threats from central authority corruption, Lyu et al. [14] proposed a decentralized mobile cloud ABE scheme in 2017 that employed an anonymous key issuance protocol for privacy protection and utilized online/offline techniques and verifiable outsourced decryption to reduce computational overhead while

also achieving revocation. Zhao et al. [15] proposed a verifiable outsourced computation mobile cloud CP-ABE scheme that verifies outsourcing results through two hash functions, but the scheme targets single-authority scenarios and cannot address real-world applications requiring multiple authorities to manage different user attributes. De et al. [16] proposed an ABE scheme for mobile cloud environments enabling fast encryption and decryption with attribute decentralization. Li et al. [17] employed a two-factor authentication mechanism to achieve user anonymity and proposed a multi-authority CP-ABE scheme for mobile cloud environments without requiring a CA. Although these schemes address data sharing issues in mobile cloud environments to some extent, they all require determining the number of attributes before system deployment and cannot flexibly add attributes without resetting system parameters.

Additionally, regarding multi-authority scenarios in cloud environments, various solutions have been proposed in literature [19~22]. Yang et al. [18] presented a decentralized revocable scheme supporting arbitrary monotone access structures. Huang et al. [19] proposed a scheme supporting large attribute domains that can arbitrarily add attributes without affecting system parameters. Cui et al. [20] constructed a revocable scheme based on composite-order bilinear groups. Zhang et al. [21] proposed a decentralized scheme requiring only one exponentiation operation for decryption and supporting large attribute domains. However, these schemes all have high performance requirements and overhead for user-side devices, making them unsuitable for mobile cloud environments.

From a security perspective, designing adaptively secure schemes (both composite-order and prime-order) requires sacrificing performance to some extent, which is not ideal for data sharing in mobile cloud environments. While selecting secure schemes cannot meet practical security requirements, using dual system encryption techniques to prove adaptive security is also not suitable. In 2015, Rouselakis et al. from Waters' team [22] proposed a static security model adapted for multi-authority settings, which better aligns with data sharing scenarios involving multiple authorities and has gained significant recognition in the field. Compared to selectively secure schemes and adaptively secure schemes proven using dual system encryption techniques, statically secure schemes offer satisfactory security and efficiency.

To the best of our knowledge, there is currently no satisfactory solution for fine-grained access control that is both efficient and secure for multi-authority attribute-based encryption in mobile cloud environments: (a) In practical scenarios, different authorities authorize and manage different user attributes, making multi-authority ABE schemes more computationally intensive and complex to manage than single-authority schemes; (b) Existing schemes require presetting attributes and authorities during system initialization, with subsequent additions necessitating global parameter resets, which is not ideal for practical efficiency; (c) Computational overhead in both encryption and decryption phases is proportional to the size of attribute sets or access structures; (d) There remains significant research space regarding the trade-off between security and

efficiency. This paper proposes a decentralized CP-ABE scheme that effectively addresses computational issues arising from multi-authority settings, offers improved security performance compared to existing schemes, and is more suitable for solving practical data sharing problems in mobile cloud environments.

---

2 Preliminaries

This chapter introduces relevant background knowledge.

2.1 Bilinear Groups

Let p be a prime, G and G_T be two cyclic groups of order p, and g be a generator of group G. Let $e : G \times G \to G_T$ be a mapping satisfying:

a) **Bilinearity.** For all $a, b \in \mathbb{Z}_p$ and $f, g \in G$, we have $e(f^a, g^b) = e(f, g)^{ab}$.

b) **Non-degeneracy.** There exists $g \in G$ such that $e(g, g)$ has order p in G_T.

Assume that group operations in G and G_T and the mapping e are all computable in polynomial time, and that the descriptions of groups G and G_T include a generator for each group.

2.2 Access Structure

Let $\mathcal{P} = \{P_1, P_2, ..., P_n\}$ denote a set of n parties. An access structure $\mathbb{A} \subseteq 2^{\mathcal{P}} \setminus \{\emptyset\}$ is a non-empty subset of the power set of $\mathcal{P}$. If the access structure $\mathbb{A}$ is monotone, then for all $B, C \in \mathbb{A}$, if $B \subseteq C$ then $C \in \mathbb{A}$. Sets in $\mathbb{A}$ are called authorized sets, while sets not in $\mathbb{A}$ are called unauthorized sets.

2.3 Linear Secret Sharing Scheme

A secret sharing scheme Π over a set of participants $\mathcal{P}$ is linear over $\mathbb{Z}_p$ if it satisfies the following two properties:

a) The shares of all participants form a vector over $\mathbb{Z}_p$.

b) There exists a matrix M with n rows and l columns called the share-generating matrix for Π. For $i = 1, 2, ..., n$, we let the function ρ map the i-th row of M to a participant. Let column vector $\vec{v} = (s, y_2, ..., y_l)$, where $s \in \mathbb{Z}_p$ is the secret to be shared and $y_2, ..., y_l \in \mathbb{Z}_p$ are randomly chosen. Then $M\vec{v}$ represents the n shares of the secret s, where $(M\vec{v})_i$ is the i-th share belonging to participant $\rho(i)$.

According to the definition of LSSS in literature [23], LSSS has linear reconstruction property. That is, if Π is a linear secret sharing scheme for access structure $\mathbb{A}$, then for any authorized set $S \in \mathbb{A}$, there exists constants $\{\omega_i \in \mathbb{Z}_p\}_{i \in I}$ such

that if $\{\lambda_i\}$ are valid shares of secret s, then the equation $s = \sum_{i \in I} \omega_i \lambda_i$ holds, while no such constants exist for unauthorized sets.

2.4 Complexity Assumptions

q-DBPBDHE2 Assumption [22]. The q-DBPBDHE2 problem in group G is described as follows: Randomly select $a, s, b_1, ..., b_q \in \mathbb{Z}_p$ and a random element $g \in G$. When given $D = (g, g^s, g^a, ..., g^{a^q}, g^{a^{q+2}}, ..., g^{a^{2q}}, \{g^{a^i b_j}, g^{b_j}, g^{a^i/b_j^2}\}_{i \in [2q], i \neq q+1, j \in [1,q]}, \{g^{a^i/(b_i b_j)}\}_{i,j \in [1,q], i \neq j})$, and a random element $T \in G_T$, an algorithm $\mathcal{A}$ solves the q-DBPBDHE2 problem with advantage ϵ if:

$$\Pr[\mathcal{A}(D, e(g,g)^{a^{q+1}s}) = 0] - \Pr[\mathcal{A}(D, T) = 0] \geq \epsilon$$

If any polynomial-time algorithm has only negligible advantage in breaking the q-DBPBDHE2 problem, then the q-DBPBDHE2 assumption holds in group G.

---

3 Scheme and Security Model

To address data security sharing issues in mobile cloud computing, this paper designs an efficient data security sharing architecture for mobile cloud environments based on attribute-based encryption methods. The proposed architecture is more aligned with practical application scenarios, satisfies fine-grained user access control, ensures better user experience during data sharing, and achieves secure, controlled, flexible, and efficient data sharing in mobile cloud environments.

3.1 Sharing Framework

The scheme mainly involves four entities: Data Owner (DO), Data User (DU), multiple Attribute Authorities (AA), and Cloud Service Provider (CSP). The sharing framework is shown in Figure 1 [Figure 1: see original paper].

Figure 1. Sharing Framework

1) **Data Owner (DO):** The DO primarily formulates access structures according to security policies, then completes data encryption through precomputation based on the access structure, and finally uploads the encryption result—ciphertext associated with the access policy—to the cloud.

2) **End User (DU):** Any user can freely access and retrieve ciphertext files from the cloud server. Decryption is possible only when the attributes possessed by the DU satisfy the ciphertext's access policy. User attributes are distributed by multiple attribute authorities according to user permissions, thereby enabling cross-domain access to ciphertexts.

3) **Multiple Attribute Authorities (AA):** AAs authorize users and generate public/private key pairs, publishing their public keys while secretly retaining their private keys. This paper assumes that each attribute in the scheme is authorized by a specific AA (e.g., ID numbers are authorized by police stations, while master' s degrees are authorized by universities), and each AA can manage multiple attributes (e.g., a graduate school office can authorize both master' s and doctoral degrees). In practice, attributes can be viewed as a combination of an AA' s public key and an attribute, ensuring that when multiple AAs have duplicate attributes, they correspond to different attributes in the system (e.g., master' s students from Tsinghua University and Peking University).

4) **Cloud Service Provider (CSP):** The CSP is generally considered honest-but-curious. Therefore, the CSP is only used for storing ciphertexts and performing partial decryption, and cannot obtain any information about data or keys. It is believed that cloud service providers possess powerful storage and computational capabilities, and DUs need not overly consider their own hardware and software conditions, as they can access authorized resources from the CSP through various configured devices.

The DO and DU mentioned here primarily refer to low-end device users, such as mobile phones and vehicle systems, but can also be high-end device users like PCs, which will not be elaborated further.

3.2 Formal Definition of the Scheme

Define the attribute space U, authority space V, and function $T : U \to V$ as a surjection from the attribute space to the authority space, i.e., for all $i \in U$, there exists $T(i) \in V$. Each user in the scheme (AA can also be viewed as a special user) has a unique global identifier GID and an attribute set S authorized by multiple AAs.

A multi-authority ciphertext-policy attribute-based encryption scheme suitable for mobile cloud environments is described by the following eight algorithms:

a) **Global Setup Algorithm** $GlobalSetup(\lambda) \to GP$: Takes security parameter λ as input and outputs global public parameters GP, which serve as public input parameters for the remaining seven algorithms. For brevity, these parameters will not be mentioned in subsequent algorithms.

b) **Authority Setup Algorithm** $AuthoritySetup(GID) \to \{PK, SK\}_\beta$: Run independently by each authority β, takes GID as input and outputs the authority' s public/private key pair (PK_β, SK_β).

c) **User Registration Algorithm** $userKeyGen(GID) \to \{userPK, key\}$: The user portion of the classic key generation algorithm, performed by the user. Takes GID as input and outputs the user' s public/private key pair $(userPK, key)$.

d) **Outsourced Key Generation Algorithm** $outKeyGen(GID, userPK, S, \{SK\}_\beta) \rightarrow SK_{out}$: The outsourced portion of the classic key generation algorithm, run by the user. For user i with identifier GID_i, user attribute set S_i, and private key set $\{SK_\beta\}$ of relevant authorities (i.e., $\forall\beta \in T(S_i)$), outputs the outsourced decryption key SK_{out} for user i, which is secretly uploaded to the cloud storage server.

e) **Precomputation Algorithm** $encPre(\{PK\}_\beta) \rightarrow IC$: Run by the user during device idle time. Takes the public key set $\{PK\}_\beta$ of relevant attribute authorities as input and outputs temporary ciphertext IC, which can be uploaded to the cloud storage server.

f) **Encryption Algorithm** $Encrypt(IC, M, (A, \rho)) \rightarrow CT$: Takes temporary ciphertext IC, plaintext message M to be encrypted, and user-defined access policy (A, ρ) as input, outputs ciphertext CT, and uploads it to the cloud storage server (this algorithm can also skip the precomputation step and encrypt the plaintext directly).

g) **Outsourced Decryption Algorithm** $decOut(SK_{out}, userPK, CT) \rightarrow CT'$: Run by the CSP' s outsourced decryption server. Takes user i' s outsourced decryption key SK_{out} and ciphertext CT as input. When the attribute set S associated with the outsourced decryption key does not satisfy the access policy in ciphertext CT, decryption fails; otherwise, it outputs semi-decrypted ciphertext CT' and sends it to user i.

h) **Terminal Decryption Algorithm** $Decrypt(key, CT') \rightarrow M$: Run by the authorized user device. Takes user private key key and semi-decrypted ciphertext CT' as input and outputs plaintext M.

3.3 Security Model

The static security model defined in this paper is a security game between a challenger and an adversary. The difference from the adaptive model is that the adversary must specify attack targets and query content immediately after receiving public parameters, send them to the challenger, and cannot change them after the game ends. Similar to the adaptive model, the static security model also allows the adversary to query user private keys and partially decrypted ciphertexts stored in the cloud multiple times, meaning the adversary can decrypt ciphertexts by querying outsourced decryption keys to obtain partially decrypted ciphertexts. Additionally, it allows the adversary to corrupt some authorities to generate their public keys for encryption. The model is modified from literature [22], primarily adding resistance against collusion attacks from multiple legitimate users through the following game stages:

a) **Setup:** The challenger runs the global initialization algorithm in the scheme and sends the public parameters GP to the adversary.

b) **Query:** The adversary first selects a subset of authorities to corrupt from the authority set V (generates and sends their public keys PK_β to the

challenger), then queries the challenger as follows:

(a) Select m authorized users $\{GID_i\}_{i=1}^{m}$ and query their public/private keys.

(b) Select some uncorrupted authorities $N \subseteq V$ and query their public keys.

(c) Select n users $\{GID_i\}_{i=1}^{n}$ and query their outsourced decryption keys. Here, S_i is the attribute set owned by user i, requiring $S_i \subseteq U$, i.e., all attributes owned by the user are authorized by uncorrupted authorities. Additionally, $n > m$ is required, meaning the adversary can query outsourced decryption keys not only for users in (a) but also for other users.

(d) Select two equal-length messages m_0, m_1 and an access policy (A, ρ) to query the challenge ciphertext. Here, for each user i who has queried private keys, S_i must not satisfy access policy (A, ρ).

c) **Challenge:** The challenger randomly selects $b \in \{0, 1\}$ and sends the challenge ciphertext CT^* to the adversary.

d) **Guess:** The adversary outputs a guess result b'. Define the adversary' s advantage in winning this game as $\epsilon = |\Pr[b' = b] - 1/2|$. If the adversary' s advantage is non-negligible, the scheme is said to be statically secure.

When the above game does not include type (a) queries, this security model reduces to attacks targeting only the cloud server.

---

4 Concrete Scheme

The proposed scheme draws on the mapping concept from literature [21], using function $T : U \to V$ to map to the authorizing authority, i.e., there exists a surjection $\delta : [l] \to V$ that maps rows of matrix A to an authority. Additionally, a precomputation outsourcing operation is introduced before scheme encryption, and mobile cloud attribute-based secure sharing is divided into four aspects: initialization, user registration, data encryption, and data access. The concrete construction is as follows:

a) **Initialization.** Run the *GlobalSetup* algorithm to generate global public parameters GP. Select a prime-order bilinear group G, where p is the order and g is the generator. Then select two hash functions $H : \{0, 1\}^* \to G$ and $F : U \to G$, mapping user identifiers GID to G and attributes to G respectively. Output global public parameters $GP = \{p, G, g, H, F, U, V, T\}$.

Each authority $\beta \in V$ runs the *AuthoritySetup* algorithm, randomly selecting $y_\beta, \alpha_\beta \in \mathbb{Z}_p^*$, computing and publishing its public key $PK_\beta = \{(e(g, g)^{\alpha_\beta}, g^{y_\beta})\}$, and secretly retaining its private key $SK_\beta = \{\alpha_\beta, y_\beta\}$.

b) **User Registration.** When a new user accesses the system, the user needs to request private keys from attribute authorities. Private keys are jointly generated by each authority corresponding to attributes in the user' s attribute set S through algorithm execution. First, run the *userKeyGen* algorithm on the mobile device, randomly selecting $z \in \mathbb{Z}_p^*$, using its own GID to compute and generate user public key $userPK = \{g^z, H(GID)^z\}$ and publish it. Relevant authorities use SK_β to output outsourced decryption keys for user i:

For each attribute i in user attribute set S, if $\beta = T(i)$, then authority β selects random element $t_i \in \mathbb{Z}_p^*$ and computes:

$$K_{i,1} = g^{\alpha_\beta} \cdot H(GID)^{y_\beta t_i}, \quad K_{i,2} = g^{t_i}$$

The user secretly stores private key $key = z$ and computes:

$$K'_{i,1} = g^{z \cdot t_i}, \quad K'_{i,2} = F(i)^{z \cdot t_i}$$

Output the cloud server key: $SK_{out} = \{GID, \{K_{i,1}, K'_{i,1}, K'_{i,2}\}_{i \in S}\}$, and add it to the cloud server key list $Klist$.

c) **Data Encryption.** When the mobile device is idle, run the *encPre* algorithm, which mainly completes precomputation for each attribute $i \in U$ before formal encryption to provide computational results for subsequent encryption. For attribute i, randomly select $r_i, \lambda_i, \omega_i \in \mathbb{Z}_p^*$ and compute:

$$IC_{i,1} = e(g,g)^{\lambda_i \alpha_\beta}, \quad IC_{i,2} = g^{r_i}, \quad IC_{i,3} = g^{y_\beta r_i}, \quad IC_{i,4} = g^{\omega_i r_i}$$

Let $IC_i = \{IC_{i,1}, IC_{i,2}, IC_{i,3}, IC_{i,4}\}$ and $TK_i = \{\lambda_i, \omega_i\}$. The above operations can also be performed by the data owner during formal encryption by precomputing only relevant attributes as needed before completing encryption. This design draws on the online/offline concept, fully utilizing user-side idle time and cloud storage capabilities to provide partial computational results for the formal encryption stage, alleviating encryption pressure to some extent.

When mobile users need to share secret data, run the *Encrypt* algorithm, sequentially inputting message m, access policy (A, ρ), intermediate ciphertext IC, and temporary key TK. Then randomly select $s, y_2, ..., y_n, z_2, ..., z_n \in \mathbb{Z}_p^*$, let vector $\vec{v} = (s, y_2, ..., y_n)$, and for all $x \in [l]$, map to authority $\beta = \delta(\rho(x))$. Compute ciphertext:

$$C_0 = m \cdot e(g,g)^{\alpha_\beta s}, \quad C_{x,5} = g^{\lambda_{\rho(x)} v_x}, \quad C_{x,6} = g^{\omega_{\rho(x)} v_x}$$

Finally output ciphertext $CT = \{(A, \rho), C_0, \{C_{x,5}, C_{x,6}\}_{x \in [l]}\}$ and upload it to the cloud storage server. The encryption can also choose to upload it to the CSP' s outsourced storage server to save device storage resources.

d) **Data Access.** The DU downloads the ciphertext from the CSP. If the ciphertext is legitimate, the mobile device uses its private key to complete decryption. When the cloud server receives an access request, it first looks up the corresponding cloud server decryption key SK_{out} in its key list $Klist$ based on the terminal user public key $userPK$, then runs $decOut$ for partial decryption. When the attribute set S associated with the terminal user' s outsourced decryption server key does not satisfy the access policy in the ciphertext, decryption fails; otherwise, let $I = \{x : \rho(x) \in S\} \subseteq \{1, 2, ..., l\}$, the decryption server computes:

$$CT' = (C_{part1}, C_{part2})$$

where:

$$C_{part1} = \prod_{x \in I} \left(e(C_{x,5}, K'_{i,1}) \cdot e(C_{x,6}, K'_{i,2})\right)^{c_x}, \quad C_{part2} = \prod_{x \in I} e(K_{i,1}, C_{x,5})^{c_x}$$

and sends the partially decrypted ciphertext CT' to the DU. Here $\{c_x\}_{x \in I}$ are constants satisfying $\sum_{x \in I} c_x A_x = (1, 0, ..., 0)$.

After receiving the partially decrypted ciphertext from the cloud server, the terminal user runs the *Decrypt* algorithm, using the retained user private key $key = z$ to complete the remaining decryption operations and finally recover:

$$m = C_0 \cdot \frac{C_{part2}}{(C_{part1})^{1/z}}$$

5 Analysis

5.1 Correctness Analysis

a) **Outsourced Decryption Process.** When attribute set S satisfies access policy (A, ρ), there exist constants $\{c_x \in \mathbb{Z}_p\}_{x \in I}$ such that $\sum_{x \in I} c_x \lambda_{\rho(x)} = s$ and $\sum_{x \in I} c_x \omega_{\rho(x)} = 0$. The following results hold:

$$\begin{aligned}
C_{part1} &= \prod_{x \in I} \left(e(g^{\lambda_{\rho(x)} v_x}, g^{z \cdot t_{\rho(x)}}) \cdot e(g^{\omega_{\rho(x)} v_x}, F(\rho(x))^{z \cdot t_{\rho(x)}})\right)^{c_x} \\
&= \prod_{x \in I} e(g, g)^{\lambda_{\rho(x)} v_x z t_{\rho(x)} c_x} \cdot e(g, F(\rho(x)))^{\omega_{\rho(x)} v_x z t_{\rho(x)} c_x} \\
&= e(g, g)^{z \sum_{x \in I} c_x \lambda_{\rho(x)} v_x t_{\rho(x)}} \cdot e(g, \prod_{x \in I} F(\rho(x))^{c_x \omega_{\rho(x)} v_x t_{\rho(x)}})^z \\
&= e(g, g)^{zs \sum_{x \in I} c_x v_x t_{\rho(x)}} \cdot e(g, \prod_{x \in I} F(\rho(x))^0)^z \\
&= e(g, g)^{zs \sum_{x \in I} c_x v_x t_{\rho(x)}}
\end{aligned}$$

b) **Mobile Device Completes Final Decryption.** The user has private key $key = z$, randomly selects elements in $\mathbb{Z}_p^*$, and since G is a cyclic group, there exists unknown z^{-1} such that:

$$m = C_0 \cdot \frac{C_{part2}}{(C_{part1})^{1/z}} = m \cdot e(g,g)^{\alpha_\beta s} \cdot \frac{e(g,g)^{\alpha_\beta s \sum_{x \in I} c_x v_x t_{\rho(x)}}}{e(g,g)^{s \sum_{x \in I} c_x v_x t_{\rho(x)}}} = m$$

Thus, the outsourced decryption key is properly distributed.

5.2 Security Analysis

Lemma 1. Assuming the Rouselakis-Waters (RW) scheme [22] is statically secure, the decentralized multi-authority CP-ABE scheme proposed in this paper for mobile cloud environments is also statically secure.

Proof. Suppose an adversary can break this scheme with non-negligible advantage in probabilistic polynomial time. Then we can construct a probabilistic polynomial-time algorithm $\mathcal{B}$ to break the RW scheme.

Algorithm $\mathcal{B}$ outputs global public parameters $GP = \{p, G, g, H, F, U, V, T\}$. The adversary $\mathcal{A}$ first selects a subset of authorities $C \subseteq V$ to corrupt (generates and sends their public keys PK_β to $\mathcal{B}$), then queries $\mathcal{B}$ as follows:

1) **Query:** The adversary selects m authorized users $\{GID_i\}_{i=1}^m$ and queries their public/private keys; selects some uncorrupted authorities $N \subseteq V$ and queries their public keys; selects n users $\{GID_i\}_{i=1}^n$ and queries their outsourced decryption keys, where S_i is the attribute set of user i, requiring $S_i \subseteq U$ (all attributes owned by the user are authorized by uncorrupted authorities) and $n > m$; selects two equal-length messages m_0, m_1 and an access policy (A, ρ) to query the challenge ciphertext, where for each user i who has queried private keys, S_i must not satisfy access policy (A, ρ).

2) **Challenge Response:** Simulator $\mathcal{B}$ queries the RW scheme for corresponding public keys of authorities in N, private keys of users $\{GID_i\}_{i=1}^m$, and challenge ciphertext. The challenger returns the corresponding private keys and public keys to $\mathcal{B}$. $\mathcal{B}$ first computes user private keys in this scheme: for $1 \le i \le m$, compute user public key $userPK_i = \{g^{z_i}, H(GID_i)^{z_i}\}$, then compute corresponding outsourced decryption keys as follows:

For each attribute $j \in S_i$, authority $\beta = T(j)$ selects random element $t_j \in \mathbb{Z}_p^*$ and computes:

$$K_{j,1} = g^{\alpha_\beta} \cdot H(GID_i)^{y_\beta t_j}, \quad K_{j,2} = g^{t_j}$$

The user secretly stores private key $key = z_i$ and computes:

$$K'_{j,1} = g^{z_i \cdot t_j}, \quad K'_{j,2} = F(j)^{z_i \cdot t_j}$$

Output outsourced decryption key $SK_{out} = \{GID_i, \{K_{j,1}, K'_{j,1}, K'_{j,2}\}_{j \in S_i}\}$.

3) **Guess:** The adversary and $\mathcal{B}$ simultaneously output guess result b'. The above distribution is indistinguishably real to the adversary. Therefore, if the adversary can break this scheme with non-negligible advantage, then $\mathcal{B}$ can also break the RW scheme with non-negligible advantage.

Lemma 2. Assuming the q-DPBDHE2 assumption holds, the RW scheme is statically secure in the random oracle model.

Proof. Literature [22] provides detailed proof, which is omitted here due to space limitations.

Theorem 1. Assuming the q-DPBDHE2 assumption holds, the proposed scheme is statically secure in the random oracle model.

Proof. Directly follows from Lemma 1 and Lemma 2.

5.3 Performance Analysis

This section compares the functionality and user-side overhead (storage and computational costs) of this scheme with related schemes. Table 1 presents the functional comparison results, while Table 2 presents the overhead comparison results.

Table 1. Functionality Comparison

Scheme	Central Authority	Large Attribute Domain	Prime Order	Precomputation	Outsourced Decryption
[18]	Yes	No	Yes	No	No
[19]	Yes	Yes	Yes	No	No
[20]	No	Yes	No	No	No
[21]	No	Yes	Yes	No	Yes
Ours	No	Yes	Yes	Yes	Yes

This paper primarily considers multi-authority settings that better reflect practical applications. Therefore, four similar multi-authority schemes are selected for comparison. Schemes in literature [18,19] both require a central authority for identity authentication, cannot avoid threats from central authority corruption, and incur non-negligible communication overhead due to required information exchange between the central authority and each authority. Literature [20] addresses these issues but is based on composite-order bilinear groups, which is significantly less efficient than prime-order schemes and does not consider other

overhead reduction methods. In contrast, literature [21] proposes a better solution, but still imposes relatively high encryption phase overhead for mobile cloud users. The scheme proposed in this paper improves upon these limitations by introducing precomputation operations during encryption, allowing users to fully utilize device idle time and cloud storage space to provide partial computational results for formal encryption without leaking any secret information, making it more suitable for mobile cloud environments.

Table 2. Overhead Comparison

Scheme	AA Public Key	User Private Key	User-side Encryption	User-side Decryption
[18]	$2U+2S+1$	$(5S+3)E+2P$	$IE+2P$	$2SE+(2S+2)I+P$
[19]	$2V$	$(5S+1)E+2P$	$IE+2P$	$2SE+(2S+2)I+P$
[20]	$2U+2S+1$	$(5S+1)E+2P$	$IE+2P$	$2SE+(2S+2)I+P$
[21]	$2V$	$2E$	$(6S+1)E+E$	$2VE+E$
Ours	$2V$	$2E$	$IE+2P$	E

Where U represents the number of attributes, V represents the number of authorities, S represents the number of user attributes, I represents the number of rows in the access structure matrix, P represents bilinear pairing operations in the group, and E represents exponentiation operations in the group. Since multiplication operations have negligible overhead compared to bilinear pairings and exponentiation operations, only the above operations are considered.

As shown in the table, literature [19] achieves constant-size AA public keys, but its scheme requires a central authority whose private key is linearly related to U. Since this scheme employs function $T : U \to V$ to map attributes to authorizing authorities, the AA public key in this scheme is only related to the number of attribute authorities rather than the number of attributes they manage. This benefit ensures that adding attributes later does not affect the AA public key. Therefore, this scheme is superior to literature [18] and [20] in terms of user storage overhead.

Secondly, regarding user private keys, literature [18~20] has AA directly generate them based on user attributes, making private key length linearly related to S. In this scheme, users first generate a public/private key pair for themselves during registration, and then AA uses this key pair along with user attributes to generate outsourced decryption keys for the cloud server. Thus, the user-side private key is constant-size and does not grow with the number of user attributes. Therefore, this scheme' s user-side storage overhead is smaller than schemes proposed in literature [18~20].

In terms of computational overhead, since this paper focuses on CP-ABE schemes suitable for data sharing where access policies are associated with ciphertexts, ciphertext length is linearly related to the number of rows I in the access matrix. This scheme considers outsourcing in both encryption and decryption, thus offering better encryption and decryption efficiency than literature [18~20]. While ensuring security, the online/offline concept is adopted for user-side encryption, with partial computations executed during the offline phase when user devices are idle, significantly reducing online encryption workload to one bilinear pairing operation and one exponentiation operation. Therefore, this scheme is superior to literature [21] in terms of encryption.

For user-side decryption, literature [18~20] has users directly decrypt ciphertexts, so both bilinear pairing and exponentiation operations during decryption are linearly related to I. In this scheme, the cloud outsourced decryption server first performs partial decryption, and finally the user only needs one exponentiation operation on the intermediate ciphertext to recover the plaintext, greatly reducing computational overhead on end devices and making it suitable for secret sharing in mobile clouds.

Furthermore, this scheme adopts the mapping concept from literature [21], using $F : U \to G$ to map the attribute space to G, with the benefit that the number of attributes in the system is not limited—any string in U can be added as a new attribute later. Additionally, this scheme uses function $H : \{0,1\}^* \to G$ to map users to G, enabling both users and authorities with unique identifiers to achieve complete decentralization, thereby resisting collusion attacks between users and authorities.

5.4 Experimental Results

This section comprehensively compares the computational overhead of this scheme with related literature through experiments. The experimental environment is Intel Core i7, 2.6 GHz, 8 GB RAM, Linux MINT 18 operating system, based on the Charm framework [24], using Type D elliptic curves from JPBC. Experiments comparing user-side encryption and decryption efficiency show that this scheme has greater advantages in encryption efficiency than literature [21].

The experimental results are shown in Figure 2 [Figure 2: see original paper]. In terms of computational overhead, this scheme is superior and suitable for resource-constrained users in cloud environments to perform secure data sharing.

Figure 2. Encryption and Decryption Time Overhead

6 Conclusion

Based on the classic ABE scheme, this paper' s scheme: (1) adopts precomputation methods during encryption and secure outsourcing during decryption, effectively reducing computational overhead on the user side during encryption and decryption; (2) introduces a mapping concept that establishes a one-to-one correspondence between attributes and group elements during scheme construction to satisfy the need for subsequent attribute addition without affecting overall scheme operation; (3) considers a more practical decentralized multi-authority scenario where authorities and users have unique identifiers, operate independently to distribute and 保管 keys, with attributes authorized by unique authorities but manageable by multiple authorities; (4) finally proves the security of this scheme under the random oracle model. Scheme analysis and experimental results demonstrate that this scheme can effectively reduce overhead on mobile device sides and is suitable for secure data sharing in mobile cloud environments.

References

[1] Mell P, Grance T. The NIST definition of cloud computing [J]. Communications of the ACM, 2011, 53(6): 50-50.

[2] White Paper. Mobile cloud computing solution brief [R]. 2010.

[3] Cui Yong, Song Jian, Miao Congcong, et al. Mobile cloud computing research progress and trends [J]. Chinese Journal of Computers, 2017, 40(2): 273-295.

[4] Gartner. Five trends in cybersecurity for 2017 and 2018 [EB/OL]. https://www.gartner.com/smarterwithgartner/5-trends-in-cybersecurity-for-2017-and-2018/.

[5] Wikipedia. The definition of mobile cloud computing [EB/OL]. https://en.wikipedia.org/wiki/Mobile_cloud_computing.

[6] Su Hang, Zhu Zhiqiang, Sun Lei. Attribute-based encryption with keyword search in mobile cloud storage [J]. Journal of Computer Research and Development, 2017, 54(10): 2369-2377.

[7] Cao Lei. Outsourcing the attribute-based encryption for mobile medical data hiding access structure [D]. Xi' an: Xidian University, 2015.

[8] Khan A N, Kiah M L M, Khan S U, et al. Towards secure mobile cloud computing: a survey [J]. Future Generation Computer Systems, 2013, 29(5): 1278-1299.

[9] Green M, Hohenberger S, Waters B. Outsourcing the decryption of ABE ciphertexts [C]// Proc of Usenix Conference on Security. [S. l.]: USENIX Association, 2011: 34-34.

[10] Li Fei, Rahulamathavan Y, Rajarajan M, et al. Low complexity multi-authority attribute based encryption scheme for mobile cloud computing [C]// Proc of IEEE International Symposium on Service Oriented System Engineering. 2013: 573-577.

[11] Hohenberger S, Waters B. Online//offline attribute-based encryption [M]// Public-Key Cryptography-PKC 2014. Berlin: Springer, 2014: 293–310.

[12] Vijay H, Goyal D, Singla S. An efficient and secure solution for attribute revocation problem utilizing CP-ABE scheme in mobile cloud computing [J]. International Journal of Computer Applications, 2015, 129(1): 1-5.

[13] Li Ruxuan, Shen Chenglin, He Heng, et al. A lightweight secure data sharing scheme for mobile cloud computing [J]. IEEE Trans on Cloud Computing, 2017, PP(99): 1-1.

[14] Lyu Maoxu, Li Xuejun, Li Hui. Efficient, verifiable and privacy preserving decentralized attribute-based encryption for mobile cloud computing [C]// Proc of the 2nd IEEE International Conference on Data Science in Cyberspace. [S. l.]: IEEE Computer Society, 2017: 195-204.

[15] Zhao Zhiyuan, Wang Jianhua. Verifiable outsourced ciphertext-policy attribute-based encryption for mobile cloud computing [J]. Ksii Trans on Internet & Information Systems, 2017, 11(6): 3254-3272.

[16] De S. J, Ruj S. Efficient decentralized attribute based access control for mobile clouds [J]. IEEE Trans on Cloud Computing, 2017, PP(99): 1-1.

[17] Li Xuejun, Lyu Maoxu. Multi-authority attribute-based encryption scheme in mobile cloud environment [J/OL]. Application Research of Computers, 2018, 35(5): 1-9. http://www.arocmag.com/article/02-2018-05-006.html.

[18] Yang Kan, Jia Xiaohua. Expressive, efficient, and revocable data access control for multi-authority cloud storage [J]. IEEE Trans on Parallel & Distributed Systems, 2014, 25(7): 1735-1744.

[19] Huang Xiaofang, Tao Qi, Qin Baodong, et al. Multi-authority attribute based encryption scheme with revocation [C]// Proc of IEEE International Conference on Computer Communication and Networks. 2015: 1-5.

[20] Cui Hui, Deng R H. Revocable and decentralized attribute-based encryption [J]. Computer Journal, 2016, 59(8): bxw007.

[21] Zhang Kai, Ma Jianfeng, Li Hui, et al. Multi-authority attribute-based encryption with efficient revocation [J]. Journal on Communications, 2017, 38(3): 83-91.

[22] Rouselakis Y, Waters B. Efficient statically-secure large-universe multi-authority attribute-based encryption [C]// Proc of International Conference on Financial Cryptography and Data Security. Berlin: Springer, 2015: 315-332.

[23] Beimel A. Secure schemes for secret sharing and key distribution [J]. International Journal of Pure & Applied Mathematics, 1996.

[24] Akinyele J A, Garman C, Miers I, et al. Charm: a framework for rapidly prototyping cryptosystems [J]. Journal of Cryptographic Engineering, 2013, 3(2): 111-128.

Note: Figure translations are in progress. See original paper for figures.