I. J. Computer Network and Information Security, **2024, 3, 37-51**
Published Online on June 8, 2024 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2024.03.04

Traitor Traceable and Revocation-oriented Attribute Based Encryption with Proxy Decryption for Cloud Devices

G. Sravan Kumar*
Department of Computer Science and Engineering, ANU College of Engineering, Acharya Nagarjuna University, Guntur, Andhra Pradesh, 522510, India
E-mail: sravanphdcse@gmail.com
ORCID iD: https://orcid.org/0000-0002-7462-7602
*Corresponding author

A. Sri Krishna
Department of Information Technology, RVR & JC College of Engineering, Acharya Nagarjuna University, Guntur, Andhra Pradesh, 522510, India
E-mail: ask@rvrjc.ac.in
ORCID iD: https://orcid.org/0000-0002-5774-8875

Received: 17 March 2021; Revised: 06 August 2021; Accepted: 23 April 2024; Published: 08 June 2024

Abstract: Cloud storage environment permits the data holders to store their private data on remote cloud computers. Ciphertext Policy Attribute Based Encryption (CP-ABE) is an advanced method that assigns fine-grained access control and provides data confidentiality for accessing the cloud data. CP-ABE methods with small attribute universe limit the practical application of CP-ABE as the public parameter length linearly increases with the number of attributes. Further, it is necessary to provide a way to perform complex calculations during decryption on outsourced devices. In addition, the state-of-art techniques found it difficult to trace the traitor as well as revoke their attribute due to the complexity of ciphertext updation. In this paper, a concrete construction of CP-ABE technique has been provided to address the above limitations. The proposed technique supports large attribute universe, proxy decryption, traitor traceability, attribute revocation and ciphertext updation. The proposed scheme is proven to be secure under random oracle model. Moreover, the experimental outcomes reveal that our scheme is more time efficient than the existing schemes in terms of computation cost.

Index Terms: Attribute Based Encryption, Large Universe, Traitor Traceability, Attribute Revocation, Ciphertext Updation.

1. Introduction

Cloud computing is the developing research area in which huge number of resources are available for services via the internet. Though it is a promising technology, it brings out issues on data confidentiality and data access control as the data publishers outsource their sensitive data to untrusted cloud servers [1]. To provide data confidentiality as well as fine-grained access control recent cloud storage systems have been implemented with cryptographic approaches. It provides data security by supplying the decryption keys only to trusted data entities. Conversely, this technique requires heavy computation due to key generation and data management activities. Thus, still it is a challenging task to develop an efficient data access control technique for cloud computing servers.

To address this challenge, in [2] proposed Fuzzy-Identity Based Encryption (FIBE) scheme that allows only the authorized data users to access the data. According to [3], Identity Based Encryption (IBE) scheme was categorized into: Key-Policy Attribute Based Encryption (KP-ABE) [4] and Ciphertext-Policy Attribute Based Encryption (CP-ABE) [5]. In KP-ABE, the data users are tagged with a decryption key consisting of an access policy. The data users are allowed to decrypt the text only when the access policy satisfies the attribute set embedded in the ciphertext. On the other hand, in CP-ABE, the ciphertext and the secret keys are embedded with access policy and attributes respectively. The authorized user could decrypt the ciphertext if the attributes follow the embedded access policy. A number of KP-ABE [6-9] and CP-ABE [10-14] methodologies have been presented to provide more expressiveness and data confidentiality for authorized data users. Of these two ABE schemes, CP-ABE schemes are found to be more significant for cloud computing environment since it offers highly efficient access control and data confidentiality for encrypted data as well as authorized data users. In addition, recent cryptographic applications rely on two major considerations on ABE system like multiple authorities and large attribute universe.

The multiple authorities of ABE system are responsible for allocating the attributes to registered data users. ABE systems have either small attribute universe or large attribute universe. In case of small attribute universe, the number of attributes to be used is fixed during system initialization phase. Furthermore, the length of the public parameter remains constant for a large attribute universe, and it linearly increases for a small attribute universe. As stated in [15], the limitations of attribute universe are: (a) with very small attribute universe, the system may surpass the attribute bound limit, and the entire system would have to restore and re-encrypt its information, (b) with very large attribute universe, the system becomes incompetent due to the growing public parameter size. Thus, it is efficient to have large attribute universe based ABE system since it is dynamic to adjust the attribute universe at any time for any given set of attributes.

Apart from these necessities, other limitations of cloud computing based data publishing scheme includes heavy computation cost, difficulties on traitor identification and attribute revocation due to ciphertext updation. The computation cost of ABE techniques grows with the amount of attributes or the type of access policy used. For a data user with limited computing resources, the complex calculations on decryption will be performed by outsourced computing devices available in the cloud called as proxy servers [16,17]. The application of proxy servers can reduce the computation cost at decryption side. Also, in certain CP-ABE broadcasting applications, the data users may intentionally leak their private key to third parties for their beneficiary purpose. Since large number of users share common attributes in their private key, it is necessary to take immediate action to protect the accessibility of data. This is achieved by tracing the traitor as well as revoking the corresponding attributes of traitor simultaneously from the broadcasting system. Therefore, traitor tracing based CP-ABE techniques can trace the traitor and provide fine-grained access control to trusted data entities [18,19].

ABE system expresses two types of revocation like: Attribute revocation [20-22] and User revocation [23-28]. Attribute revocation means, the attributes in the system could be revoked by the data provider at any time for any reason. Under such circumstances, it is important to update the ciphertext of authorized users immediately thereby not allowing the revoked attribute users to decrypt the ciphertext. In some cases, the data users may leave the ABE system after their registration time expiration; this is termed as user revocation. In ABE method, a greater number of users follow similar attributes in their secret key, thus attribute and/or user revocation will definitely influence the unrevoked data users who are using the same attribute. Considering the above requirements, it is necessary to design a computation-less large attribute universe CP-ABE system with features such as outsourcing decryption, traitor traceability, attribute revocation and ciphertext updation.

1.1. Contribution

In this paper, we propose a practical construction of CP-ABE to offer fine-grained data access control with proxy decryption, traitor tracing, attribute revocation and ciphertext updation. This technique encrypts the data with an efficient as well as confidential access policy, and the data user’s private key is embedded with authenticating attributes. As a fine-grained access policy based technique, the encrypted data is decrypted by the data user only when the access policy fulfils the attributes embedded in the private key. The large attribute universe adopted in our scheme is found to be exponentially larger, and the length of the public parameter remains smaller or almost constant. For users with limited computing resources, decryption is performed on outsourced devices called as proxy servers. The malicious user who intentionally leaks the private key is identified and the attributes associated with the traitor is revoked. Moreover, the ciphertext associated with the revoked attribute is updated by the cloud server. The security of the proposed technique is proved in random oracle framework model.

1.2. Organization

The remaining part of this paper is designed as follows. Section 2 outlines the previous research works related to the features to be implemented in our technique. The preliminaries of CP-ABE technique such as bilinear mapping, access policy and Linear Secret Sharing Structures (LSSS) were described in Section 3. Section 4 gives the design of system and security model used in our technique. The construction of traitor traceable and revocation-oriented large attribute multi-authority based CP-ABE technique is presented in Section 5. Section 6 analyse the results obtained for the proposed technique and discuss with the existing methodologies. The conclusion and the future work are given in Section 7.

2. Related Works

In this section, the existing research works of ABE have been briefly discussed under five categories: Multi-authority ABE, Large attribute universe, Outsourced decryption, Traitor traceability and Revocation mechanism.

2.1. Multi-authority ABE

Multi-authority based ABE model was first initiated by Chase [29]. In this model, there is a trusted central authority that has control over the multiple attribute authorities. It is possible for the central authority to decrypt the ciphertext of all users and it is untrusted if the entire system is corrupted. A new multi-authority FIBE method without any central authority is provided by [30]. In this technique, during system initialization, each authority would interact with other authority without sharing any sensitive information about data users. A Proxy re-encryption based multi-authority technique implemented by [31] required more time cost for decrypting the symmetric keys. In [32] presented hidden policy based multi-authority ABE method that is verified to be secure under selective security model and it preserve the data access control under random oracle model. Thus, still it is a challenging task to provide a fully secure multi authority based ABE system with reasonable time cost.

2.2. Large Attribute Universe

In the literature, several research works have been presented on ABE schemes with large attribute universe. [18] designed large attribute based system with both the KP-ABE and CP-ABE approaches. However, their technique is certified to be secure only under selective security models. The policy hidden ABE technique proposed by [15] has supported large attribute space but it is implemented in Composite order bilinear groups. A large attribute based Hierarchical ABE (HABE) and CP-ABE methods were presented by [33] and [34] respectively. Both these methods are implemented on prime order groups with expressive monotonic access policies. Hence, it is necessary to design more efficient and expressive access policy based ABE technique with large attribute space on prime order groups.

2.3. Outsourced Decryption

A privacy preservation based outsourced computation technique is suggested by [17]. The communication overhead and the computation complexity of this technique are increased due to the interaction among data provider as well as outsourced server. In [32], the complex computation algorithm on decryption was partly calculated by an outsourced cloud server called as semi-trusted entity. This method is verified to be confidential under selective security model. In [16] founded an outsourcing computation scheme that permits secured data transmission. In this technique, the cloud servers who perform the outsourced computation would convert the ciphertext into El-Gamal type group, which is then easily decrypted by the end user. However, for a system with limited number of users, this technique will increase the length of the transformed ciphertext.

2.4. Traitor Traceability

In [35] proposed the first white box traceability based CP-ABE scheme that identifies the traitor who sell his/her decryption keys to third parties for some benefits. Traitor traceability based practical CP-ABE technique implemented in [18] is proven to be confidential under prime order groups. This scheme directly revokes the malicious users from the system but it cannot achieve fine-grained access control and does not support ciphertext updation after user revocation. [19] offered a traceable and revocable CP-ABE technique that supports ciphertext updation but it cannot update the user's key after attribute revocation. The traceable scheme recommended by [36] can trace the malicious user who leaked the private key to third party but it cannot revoke the user.

2.5. Revocation Mechanism

In [21] suggested an attribute revocation technique that achieved both forward as well as backward privacy. In their method, the storage overhead is greatly increased as large number of encrypted texts is stored in the cloud. An efficient attribute revocation based CP-ABE technique preventing user's collusion was presented in [20]. During attribute revocation, this method eliminates the collusion among revoked and unrevoked users. But the time cost of this technique depends upon the amount of attributes available in the system. [22] recommended an attribute revoking process with fine-grained access policy and provable deletion. Nonetheless, this technique is implemented using KP-ABE technique.

In [37] introduced a new protocol for confidential data sharing over cloud storage environment. This protocol named Sec Cloud Sharing was constructed to support two types of revocation like attribute revocation and policy-level revocation. Nevertheless, this technique is implemented with AND gate access policy, and it does not support any other expressive access policies. In [38] introduced a technique that support outsourced computation and ciphertext updating for the extended cloud computing infrastructure called as fog computing. [39] have given a policy updating technique for big data storage infrastructure. But the decryption cost and the ciphertext length of this method is dependent on the number of attributes.

3. Technique Preliminaries

3.1. Bilinear Mapping

Definition 1

Bilinear Map [34]: Consider two prime orders (p) cyclic multiplicative groups G_1 and G_2. The generator of the group G_1 is g. The mapping between the groups $e: G_1 \times G_1 \rightarrow G_2$ follows the following properties.

- Bilinear.

$$e(X^x, Y^y) = e(X,Y)^{xy},\ \forall\ (X,Y) \in G_1\ \&\ (x,y) \in Z_p \tag{1}$$

- Non-degenerate.

$$e(X,Y)\ \neq 1 \tag{2}$$

- Computable. $e(X,Y)$ is efficiently computable

$$\forall\ (X,Y) \in G_1 \tag{3}$$

3.2. Structure of Access Policy

Definition 2

Access Structure [34]: Consider a set of parties, $P = \{P_1, P_2, \dots\dots, P_n\}$. The group of parties, $\mathbb{A} \subseteq 2^{\{P_1,P_2,\dots\dots,P_n\}}$ is said to be monotone, if and only if: $\forall\ B, C$: if $B \in \mathbb{A}\ \&\ B \subseteq C$, then $C \in \mathbb{A}$. An access structure (or monotonic access structure) comprises a set (or monotonic set) $\mathbb{A}$ of non-void subsets of parties, that is, $\mathbb{A} \subseteq 2^{\{P_1,P_2,\dots\dots,P_n\}} \setminus \{\emptyset\}$. The elements in set $\mathbb{A}$ are termed as certified elements; else the elements are termed as uncertified elements.

In our technique, the role of parties is played by the attributes. The attributes contain only the certified elements and the access policy used is a monotonic access policy.

3.3. Linear Secret Sharing Structure

Definition 3

Linear Secret-Sharing Structure (LSSS) [34]: The secret-sharing structure Π over a group of parties P is said to be linear over Z_p if

- The secret-share $s \in Z_p$ for the parties form a vector on Z_p.
- There occurs a matrix $M \in Z_p^{l \times n}$ called as share-generating matrix for Π. For $i = 1,2, \dots . l$, the i^{th} row of the matrix is categorized by a party $\rho(i)$. If we take the column vector $c = (s, r_2, \dots . , r_n)$, where $(r_2, \dots . , r_n)$ are randomly selected from Z_p, the vector of l shares of the secret s according to Π is M_{cv}. The share $(M_{cv})_i$ belongs to the party $\rho(i)$.

According to [34], the LSSS follows a linear reconstruction property, which is explained as follows: Let us consider that Π be a LSSS for the access policy $\mathbb{A}$ and an authorized set $A_S \in \mathbb{A}$. Consider $I \subset \{1,2, \dots \dots \dots , l\}$ as $I = \{i : \rho(i) \in A_S\}$, for any valid share $\{\delta_i\}$ of the secret s according to Π, there exist constants $\{w_i \in Z_p\}_{i \in I}$ such that $\sum_{i \in I} w_i \delta_i = s$. This constant does not exist for unauthorized sets.

4. Models and Assumptions

4.1. System Model

In cloud storage environment, the data providers and the data users rely on remote public cloud servers for storing, accessing and processing vast amount of data sets. The system model of CP-ABE scheme used in the proposed scheme is comprised of entities like: data provider, trusted authority, multiple attribute authorities, cloud server, proxy server and data user. The sensitive plaintext data to be outsourced to the untrusted cloud server is transformed into ciphertext by encrypting it with a fine-grained access policy. This encryption is efficiently performed by the data provider who holds the data. A trusted central authority distributes universal IDs and Attribute authority IDs to data users and attribute authorities respectively. The attribute authorities of our scheme are accountable for determining and supplying the public keys and secret keys to data providers and data users respectively. In our scheme, the attribute set is maintained by multiple authorities. A data user with a proper secret key can access the data and view the data only after successful decryption. Some data users may have limited computing resources which makes decryption difficult. Under such circumstances, decryption is performed by outsourced devices like proxy servers available in the cloud. The proposed system architecture is shown in Fig. 1.

4.2. Security Framework

The security framework of the proposed technique is designed based on the selective security model between Challenger C and Adversary A as in [40]. The key queries undergone by A are transferred to C while receiving the public parameters. In our technique, the adversary A is allowed to corrupt certain attribute authorities to examine the malicious attack. The corrupted attribute authorities would continue to be the same till the completion of game. The security analysis is performed as given below.

Setup

C runs the setup phase and issues the public parameters to *A*.

Adversary's Queries

The adversary queries as follows.

- It picks an attribute set of corrupted authority and generates the public keys of that authority and sends it to *C*.
- It picks an attribute set of non-corrupted authority and enquires the public keys of these set.
- It enquires the private keys by providing a sequence pair (universal identity and attribute set) such that, it does not query the private keys of corrupted authority since it can generate it by its own.
- It queries the transformation keys by providing the sequence pair (universal identity and attribute set) such that, it does not query the transformation keys of universal identities that are queried for their private keys.
- It provides two equal size messages $\mathrm{M}_1, \mathrm{M}_2$ and an access policy (M, ρ) to *C* for a challenging ciphertext. This access policy should not match the attribute sets that are queried for private keys.
- It queries the updated cluster key by providing a sequence pair and revoked attribute.
- It queries the ciphertext updating key by providing a new challenging access policy.

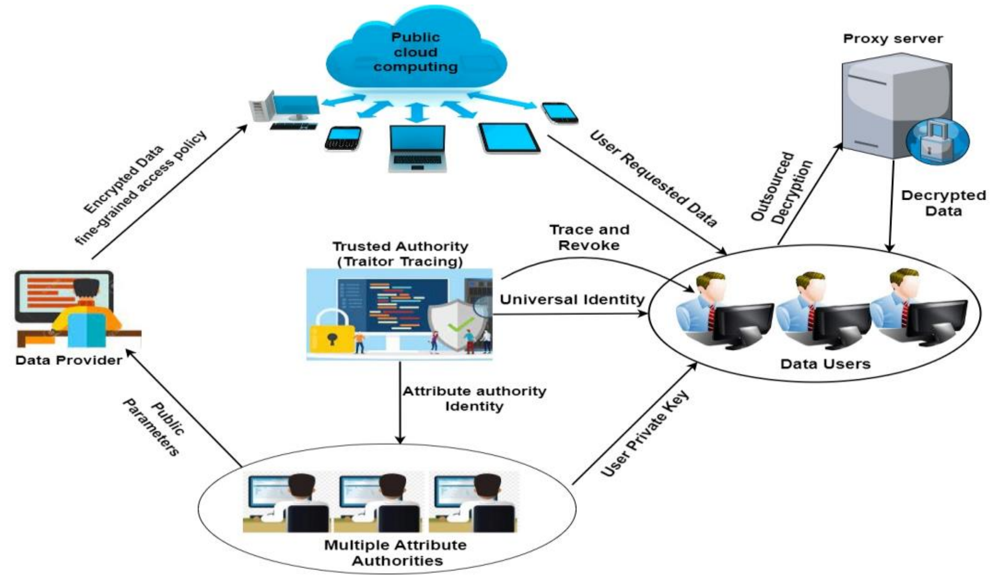


Fig.1. System architecture

Challenger's Replies

Challenger *C* tosses a random coin $\beta \epsilon \{0, 1\}$ and provides the adversary with:

- Public keys of non-corrupted authority.
- Private keys of submitted sequence pairs.
- Transformation key.
- Challenging ciphertext.
- Updated attribute cluster key and
- Ciphertext update key.

Guess

An output a guess β' for β.
The benefit of *A* in this security framework is defined as:

$$\Pr[\beta = \beta'] - \frac{1}{2} \tag{4}$$

Definition 4

Traitor traceable and revocation-oriented large attribute multi-authority based CP-ABE technique with proxy decryption is proved to be selectively secure (against static corruption of authorities) if all polynomial time adversaries gain at most negligible advantage from the above security model.

5. Traitor Traceable and Revocation-oriented Large Universe Multi-authority based CP-ABE

This section describes the construction of traitor traceable and revocation-oriented large attribute universe multi-authority based CP-ABE model supporting efficient encryption, outsourced decryption, traitor tracing, attribute revocation and ciphertext updation.

5.1. System Initialization

This step performs two functions like: Initial setup and Multi-authority setup.

- *Initial setup:* Consider a security parameter (λ) as input and perform bilinear mapping on two multiplicative cyclic groups of prime order p. The bilinear mapping is $e: G_1 \times G_1 \rightarrow G_2$ and g is the generator of the group G_1. The large attribute universe and the multi-attribute authorities are denoted as AU and MA respectively. The universal identities of all the users are given as UID. In addition, it includes three characteristics such as X, Y and Z. X maps the universal identities to components of G_1. Y maps the users' attributes to components of G_1. Finally, Z maps each and every attribute to corresponding identity of attribute authority. Then the public parameter is evaluated as:

$$PP = (G_1, G_2, p, e, g, e(g, g), AU, MA, UID, X, Y, Z) \tag{5}$$

- *Multi-authority setup:* This algorithm is executed by a trusted central authority. For each attribute authority $aa \in MA$, two random values are chosen: $(\alpha_{aa}, \gamma_{aa}) \in Z_p$. The Secret key and public key of the attribute authorities are respectively calculated as:

$$SK_{aa} = \{\alpha_{aa}, \gamma_{aa}\} \text{ and } PK_{aa} = \{e(g, g)^{\alpha_{aa}}, g^{\gamma_{aa}}\} \tag{6}$$

5.2. User Registration

This step undergoes two processes such as: Attribute key formation and Attribute cluster key formation.

- *Attribute key formation:* If a data user with identity $uid \in UID$ wants to establish a connection with an attribute authority aa, an attribute set $AS_{uid,aa}$ is generated for the data user. For each of the attribute j in this attribute set, a random value is chosen: $t_j \in Z_p$. Then it evaluates,

$$K_{uid,j} = g^{\alpha_{aa}} X(uid)^{\gamma_{aa}} Y(j)^{t_j} \text{ and } K'_{uid,j} = g^{t_j} \tag{7}$$

The private key of the user is:

$$UPK_{AS_{uid,aa}} = \{K_{uid,j}, K'_{uid,j}\}_{j \in AS_{uid,aa}} \tag{8}$$

- *Attribute cluster key formation:* This algorithm first creates a binary tree for each attribute authority. As depicted in Fig. 2, for each of their attribute a, the attribute authority maintains an attribute cluster AC_a, which defines the data users who are using this attribute. Each node n_k of the binary tree is provided with a random key $rk_k \in Z_p$. The members of the cluster AC_a are assigned to the child node of binary tree. Then the path key for each user is determined from the child node to the source node. That is, the path key of the data user usr_4 is,

$$Pa_{key} = \{rk_7, rk_3, rk_1\} \tag{9}$$

Furthermore, a unique attribute cluster key $ACK_a \in Z_p$ is chosen by the attribute authorities and it is distributed to the cloud servers. In the path key, rk is used as a symmetric key for encrypting ACK_a.

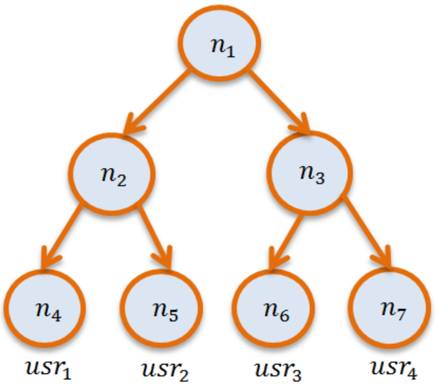


Fig.2. Binary tree for attribute cluster key formation

5.3. *Data Encryption*

Before outsourcing the sensitive data to cloud servers, the data providers encrypt the data with a fine-grained access policy as given in the following procedure.

- *Encryption:* The inputs to the encryption algorithm are: plaintext message msg, an access policy $AP = (M, \rho)$, public parameter PP and a group of attribute-authority public key PK_{aa}. In the access structure (M, ρ), M is a $m \times n$ dimensional matrix and ρ maps each row of the matrix into an attribute $\rho(j)$. Since each row of the matrix represents an attribute, a function β maps each row to its corresponding attribute authority. i.e., $\beta(j) = Z(\rho(j))$. Data encryption is proceeded as follows: the data owner pick some random values $(r_1, r_2, \dots \dots ., r_m) \in Z_p$ and random vectors v_1 and v_2 such that, $v_1 = (s, v_2, \dots \dots, v_n)^T$ and $v_2 = (0, z_2, \dots \dots, z_n)^T$. For $j = 1$ to m, calculate $\lambda_j = M_j v_1$, $w_j = M_j v_2$. Then the data owner estimates the ciphertext as:

$$CT_1 = msg.e(g,g)^s,\ CT_{2,j} = e(g,g)^{\lambda_j} e(g,g)^{\alpha_{\beta(j)} r_j},\ CT_{3,j} = g^{-r_j},\ CT_{4,j} = g^{\gamma_{\beta(j)} r_j} g^{w_j},\ CT_{5,j} = Y(\rho(j))^{r_j} \tag{10}$$

The resultant ciphertext is obtained as:

$$CT = (AP, CT_1, \{ CT_{2,j}, CT_{3,j}, CT_{4,j}, CT_{5,j}\}_{j \in [m]} \tag{11}$$

where, $[m] = \{1,2, \dots \dots, l\}$.

5.4. *Data Re-encryption*

The encrypted data is transmitted to the remote cloud computers. While receiving the encrypted data, the cloud servers will re-encrypt the data for efficient storage and transmission.

- *Re-Encryption:* In the Ciphertext, for each attribute $\rho(j)$ in the access structure AS, its attribute cluster key $ACK_{\rho(j)}$ is used for re-encryption. The re-encrypted ciphertext is obtained as follows:

$$RCT = (AP, CT_1, \{ CT_{2,j}, CT_{3,j}, CT_{4,j}, CT_{5,j}^{'} = (CT_{5,j})^{ACK_{\rho(j)}}\}_{j \in [m]} \tag{12}$$

The re-encrypted ciphertext is then stored in the cloud servers.

5.5. *Data Decryption*

If the private key of the data user fulfils the access policy encrypted in the ciphertext, then the data user can decrypt the data as given in the following procedure.

- *Decryption:* If a data user wants to decrypt the encrypted text, the users attribute set AS_{uid} should match the attributes $\rho(j)$ encrypted in the access policy of ciphertext. The algorithm outputs §, if $AS_{uid} \notin AP$. Else, if $AS_{uid} \in AP$, the algorithm returns the constants $\{c_a : a \in I\}$ such that $\sum_{a \in I} c_a M_a = (1,0, \dots \dots, 0)$, where $I = \{a : \rho(a) \in AS_{uid}\}$. Further, for each attribute $\rho(j) \in AS_{uid}$, the data user reconstructs the attribute cluster key $ACK_{\rho(j)}$ by using the path keys and evaluate:

$$K_{uid,\rho(j)}^{\approx} = (K_{uid,\rho(j)}^{'})^{1/ACK_{\rho(j)}} \tag{13}$$

Then it computes:

$$UPK_{AS,uid,aa}^{'} = \{K_{uid,\rho(j)}, K_{uid,\rho(j)}^{\approx}\}_{\rho(j)\in AS_{uid,aa}} \tag{14}$$

The message is decrypted as given below.

$$\begin{aligned}
&\prod_{a\in I}(CT_{2,a}.\,e(\,K_{uid,\rho(a),}CT_{3,a}).\,e\big(X(uid),CT_{4,a}\big).\,e(K_{uid,\rho(a)}^{\approx},CT_{5,a}^{'}))^{c_a} \\
&= \prod_{a\in I}\begin{pmatrix} e(g,g)^{\lambda_a}e(g,g)^{\alpha_{\beta(a)}r_a}.\,e(g^{\alpha_{\beta(a)}}X(uid)^{\gamma_{\beta(a)}}Y\big(\rho(j)\big)^{t_j},g^{-r_a}). \\ e(X(uid),g^{\gamma_{\beta(a)}r_a}g^{w_a}).\,e(g^{\frac{t_a}{ACK_{\rho(a)}}},Y\big(\rho(a)\big)^{r_a.ACK_{\rho(a)}}) \end{pmatrix}^{c_a} \\
&= \prod_{a\in I}(e(g,g)^{\lambda_a}e(X(uid),g^{w_a})^{c_a} \\
&= e(g,g)^s \\
msg &= CT_1/e(g,g)^s
\end{aligned} \tag{15}$$

5.6. *Outsourcing Decryption*

For a data user with limited computing facilities, the complex calculations on decryption will be performed by the proxy devices available in the cloud as given below.

- *Transformation key formation:* For each attribute $\rho(j)\in AS_{uid}$, an user with identity uid computes $K_{uid,\rho(j)}^{\approx} = (K_{uid,\rho(j)}^{'})^{1/ACK_{\rho(j)}}$ by using the corresponding attribute cluster key $ACK_{\rho(j)}$ and evaluate $UPK_{AS,uid,aa}^{'} = \{K_{uid,\rho(j)}, K_{uid,\rho(j)}^{\approx}\}_{\rho(j)\in AS_{uid,aa}}$. The data user selects a random value $f\in Z_p$ and estimate:

$$T_{uid,\rho(j)}^{1} = (K_{uid,\rho(j)})^{1/f},\, T_{uid,\rho(j)}^{2} = (K_{uid,\rho(j)}^{\approx})^{1/f} \tag{16}$$

The transformation and recovery key are established as:

$$Transkey_{AS,uid} = \{T_{uid,\rho(j)}^{1}, T_{uid,\rho(j)}^{2}\}_{\rho(j)\in AS_{uid}} \text{ and } Reckey_{AS,uid} = f \tag{17}$$

The data users slightly modify the re-encrypted ciphertext by using the recovery key before outsourcing it to the proxy servers. Thus, the modified ciphertext takes the form:

$$MCT = (AP, CT_1, \{CT_{2,j}^{'} \;= CT_{2,j}^{1/f}, CT_{3,j}, CT_{4,j}^{'} = CT_{4,j}^{1/f}, CT_{5,j}^{'}\}_{j\in[m]}) \tag{18}$$

This modified ciphertext is then transferred to the proxy device along with the transformation key. The recovery key is preserved by the data user.

- *Transform:* This algorithm is operated by the proxy server. It obtains an error message, if $AS_{uid}\notin AP$. Else, if $AS_{uid}\in AP$, it sets $I = \{a{:}\,\rho(a)\in AS_{uid}$ and estimate the constants $\{c_a\in Z_p\}_{a\in I}$ such that $\sum_{a\in I} c_a M_a = (1,0,\ldots\ldots,0)$. Then it calculates:

$$\begin{aligned}
&\prod_{a\in I} CT_{2,a}^{'}.\,e(\,T_{uid,\rho(j)}^{1},CT_{3,a}).\,e\big(X(uid),CT_{4,a}^{'}\big).\,e(T_{uid,\rho(j)}^{2},CT_{5,a}^{'}))^{c_a} \\
&= \prod_{a\in I}\begin{pmatrix} e(g,g)^{\lambda_a/f}e(g,g)^{\alpha_{\beta(a)}r_a/f}.\,e(g^{\alpha_{\beta(a)}/f}X(uid)^{\gamma_{\beta(a)}/f}Y\big(\rho(j)\big)^{t_j/f},g^{-r_a}). \\ e\big(X(uid),g^{\gamma_{\beta(a)}r_a/f}g^{w_a/f}\big).\,e(g^{\frac{t_a}{ACK_{\rho(a)}}},Y\big(\rho(a)\big)^{r_a.ACK_{\rho(a)}}) \end{pmatrix}^{c_a} \\
&= \prod_{a\in I}(e(g,g)^{\lambda_a/f}e(X(uid),g^{w_a/f})^{c_a} \\
&= e(g,g)^{s/f}
\end{aligned} \tag{19}$$

The partially decrypted ciphertext is:

$$PDCT = \{AP, CT_1, e(g,g)^{s/f}\} \tag{20}$$

- *Decryption:* It is run by the data user. If $AS_{uid}\notin AP$, it outputs §. Otherwise, if $AS_{uid}\in AP$, it determines

$$(e(g,g)^{s/f})^{Reckey_{AS,uid}} = e(g,g)^s \tag{21}$$

and therefore,

$$msg = CT_1/e(g,g)^s \tag{22}$$

5.7. *Traitor Tracing*

The traitor tracing algorithm is run by trusted authority by performing the following two operations.

- *Key sanity check:* If the private key of the user is distrusted or found to be well formed, then the trusted authority runs the key sanity check as given below.
- The users private key is in the form:

$$UPK_{AS_{uid,aa}} = \{K_{uid,j}, K'_{uid,j}\}_{j \in AS_{uid,aa}}, \text{ where } K_{uid,j} \in G_2 \text{ and } K'_{uid,j} \in Z_p \tag{23}$$

$$\boldsymbol{e}\left(K_{uid,j}, g\right) = e(K'_{uid,j}, g^{\alpha_{aa}}) \tag{24}$$

$$\boldsymbol{e}\left({K_{uid,j}}^{K'_{uid,j}}, g^{\alpha_{aa}}, g^{K'_{uid,j}}\right) = e(g,g)^{\alpha_{aa}}\, e(K_{uid,j}, K'_{uid,j}) \tag{25}$$

$$\exists a \in UPK, s.t.\, \boldsymbol{e}\left(K'_{uid,j}, g\right) = e\left(GID_a, K_{uid,j}\right) \neq 1 \tag{26}$$

If the data user passes the key sanity check, the algorithm outputs a value 1. Else, it outputs a value 0 defining that the user's key is a well formed key.

- *Trace:* If the user's private key cannot pass the key sanity check, then the identity of traitor is traced as shown below.

    - It extracts user's identity from $K_{uid,j}$.
    - The trusted authority searches this identity in attribute cluster key ACK_a, if the same identity is found, it reports the corresponding malicious user or reports another user who is not found in ACK_a.
    - The trusted central authority sends the traitor identity to attribute authority.

5.8. *Attribute Revocation*

If a user drops any of the attribute (say a_r) in their attribute set, the attribute authority should choose a new attribute cluster key $ACK_a^{\approx} \in Z_p$ for the dropped attribute. In the binary tree, if an attribute is revoked by the user usr_3, then the unrevoked users for that attribute are usr_1, usr_2 and usr_4. The new attribute cluster key is transferred to the cloud server as well as the data users. This updated key is known only to the unrevoked users and thus the revoked users do not collude with the cloud servers. Attribute revocation is achieved by two steps as given below.

- *Attribute cluster key updation:* Let us consider the revoked attribute as $\rho(j') = a_r$. After receiving the newly updated attribute cluster key, the users update their secret key as follows.

$$\begin{gathered} UPK'_{AS,uid,aa} = (K_{uid,j} = g^{\alpha_{\beta(j)}} X(uid)^{\gamma_{\beta(j)}} Y\left(\rho(j)\right)^{t_{\rho(j)}}, \\ \forall j[m] \setminus \{j'\} :\ K_{uid,j}^{\approx} = g^{\frac{t_{\rho(j)}}{ACK_{\rho(j)}}}, \\ j = j' :\ K_{uid,j}^{\approx} = g^{t_{\rho(j)}/ACK'_{a_r}}) \end{gathered} \tag{27}$$

5.9. *Ciphertext Updation*

In this step, the ciphertext connected with the revoked attribute a_r is updated by the cloud server.

- *Updated Ciphertext:* Before ciphertext updation, the cloud server chooses two random vectors such as r_1' and r_2' such that, $r_1' = (s', v_2', \dots\dots, v_n')^T$ and $r_2' = (0, z_2', \dots\dots, z_n')^T$. For $j = 1$ to m, pick random values $(x_1', \dots\dots, x_m')$ and calculate $\lambda_j' = M_j r_1'$, $w_j' = M_j r_2'$. Then the cloud server updates the ciphertext as:

$$\begin{gathered} CT^{\approx} = (CT_1 e(g,g)^{s'}, \left\{ CT_{2,j} e(g,g)^{\lambda'_j} e(g,g)^{\alpha_{\beta(j)} r'_j}, CT_{3,j} g^{-r'_j}, CT_{4,j} g^{\gamma_{\beta(j)} r'_j} g^{w'_j} \right\}_{j \in [m]}, \\ \{(CT_{5,j} Y\left(\rho(j)\right)^{r'_j})^{ACK_{\rho(j)}}\}_{j \in [m] \setminus \{j'\}}, \{(CT_{5,j} Y\left(\rho(j)\right)^{r'_j})^{ACK'_{a_r}}\}_{j \in j'} \end{gathered} \tag{28}$$

6. Analysis of Proposed Method

6.1. Security Proof Analysis

The non-adaptive security evidence of the proposed technique is explained in random oracle model below.

Lemma 1

The proposed scheme is statically secure with an assumption that CP-ABE method in [40] is statically secure.

Proof

If there occurs a polynomial-time adversary ***A*** that has benefit against the proposed method in our security model, a simulator S is constructed with benefit against the RW (Rouselakis-Waters) scheme in [40] and the challenger in their scheme be ***C***.

Setup

C sends the public parameters, $PP = (Gr_1, Gr_2, e, g, AU, MA, UID, X, Y, Z)$ to simulator S. Then S transfers the public parameters to ***A***.

Adversary's Queries

A selects an attribute set of corrupted authority, $MA_C \subset MA$ and generates the public keys using RW scheme. ***A*** respond to S with:

- Corrupted attribute authority set, $MA_C \subset MA$
- Non-corrupted attribute authority set, $MA_{NC} \subset MA$, where $MA_{NC} = MA - MA_C$.
- A sequence pair $(uid, AS_{uid,aa})$, which indicates that ***A*** queries the private key of the universal identity uid for the attribute set $AS_{uid,aa}$.
- A set of Challenge access policy $AP = \{(M_1, \rho_1), (M_2, \rho_2), \dots \dots, (M_N, \rho_N)\}$ and two equal length messages M_0 and M_1.

Challenger's Replies

The simulator S obtains the above queries and performs the following tasks.

- ***C*** computes the public keys of non-corrupted attribute authority set and sends it to S. S sends these keys to ***A.***
- S sends the private key queries of ***A*** to ***C. C*** determine the private keys, $UPK_{AS_{uid,aa}} = \{K_{uid,j}, K_{uid,j}^{'}\}_{j \in AS_{uid,aa}}$ and corresponding path key for each attribute and transmit the final output to S
- For the transform key query from ***A***, S undergoes secret key query to obtain $UPK_{AS_{uid,aa}} = \{K_{uid,j}, K_{uid,j}^{'}\}_{j \in AS_{uid,aa}}$ from ***C***. Then S modifies these keys as: $UPK_{AS,uid,aa}^{'} = \{K_{uid,j}, K_{uid,j}^{\approx} = (K_{uid,j}^{'})^{ACK_j})\}_{j \in AS_{uid,aa}}$. S picks a random number $f \in Z_p$ and establishes the transformation key, $Transkey_{AS,uid,aa} = \{T_{uid,j}^{1}, T_{uid,j}^{2}\}_{j \in AS_{uid,aa}}$ and transmits it to ***A.***
- S provides the messages and access policies of ***A*** to C. C toss a random coin $\beta \epsilon \{0, 1\}$ and encrypts the message M_β under these access policies and transmits the encrypted text corresponding to each access policies $\{CT_1, CT_2, \dots \dots, CT_N\}$ to S. Now, S evaluates the attribute cluster key and re-modifies the ciphertext:

$$CT^{\approx}{}_{k} = ((M_k, \rho_k), CT_{k,1}, \{CT_{2,j}, CT_{3,j}, CT_{4,j}, CT_{5,j}^{'} = (CT_{5,j})^{ACK_{\rho_k(j)}}\}_{j \in [m_k]}), \forall\ k \in (1, N). \tag{29}$$

Later, this modified ciphertext is transferred to ***A***.

- S operates the attribute cluster key updation algorithm and generates $UPK_{AS,uid,aa}^{'}$. This updated attribute cluster key is then transmitted to ***A***.
- The ciphertext updation key query of ***A*** is transmitted to C through S. However, no advantage is gained by S in this process. C operates the ciphertext key updation algorithm and returns the key to S. S transmits it to ***A***.

Guess

Finally, $\boldsymbol{A}$ makes a guess $\beta' \epsilon \{0, 1\}$, $\boldsymbol{S}$ outputs β'. This concludes the simulation. Therefore, if $\boldsymbol{A}$ can break our technique with an advantage ε, then $\boldsymbol{S}$ can break the RW scheme with similar probability.

6.2. Performance Evaluation

In this section, the performance of proposed technique is estimated and compared with the previous techniques both theoretically and experimentally.

A. *Feature Comparison*

The feature comparison of our technique and the existing techniques are shown in Table 1. From Table 1, it is clear that our scheme and the schemes in [21, 23, 32, 33, 34, 37] allow multiple attribute authorities to control the attributes used in the system. For the PHOABE technique in [32], the complex calculation on decryption is partially performed by semi trusted cloud servers. Traceability based CP-ABE methodologies were presented in [18,19,34]. However, the technique in [34] allows multi-attribute authority to generate keys whereas a single authority estimates the data user key in [18,19]. Multi-attribute authority based security assisted cloud sharing CP-ABE approach is offered in [21,37]. Both these techniques can drop the attributes from the system, but the scheme in [37] could not update the ciphertext after attribute revocation. The multi-authority based CP-ABE technique in [23] would provide ciphertext updation after revoking the user from the system but it does not revoke any attributes. The CP-ABE techniques in [19,21,23] are capable of supporting attribute revocation.

From this comparison, our scheme is superior to existing CP-ABE techniques in terms of supporting multi-attribute authorities, large attribute universe, traitor tracing, attribute revocation and ciphertext updation. Furthermore, our scheme can reduce the computation burden by outsourcing the heavy computation on decryption to proxy servers. Hence, our technique is found to be more suitable for cloud storage scenarios.

Table 1. Features comparison

Technique	Multi-attribute authority	Large attribute space	Outsourced Decryption	Traitor Traceable	Attribute Revocation	Ciphertext Updation
[18]	×	√	×	√	√	×
[19]	×	×	×	√	×	√
[21]	√	×	×	×	√	√
[23]	√	×	×	×	×	√
[32]	√	×	√	×	×	×
[33]	√	√	×	×	×	×
[34]	√	√	×	√	×	×
[37]	√	×	×	×	√	×
Ours	√	√	√	√	√	√

Table 2. Notations used in performance evaluation

Notation	Meaning
B_z	bit size of prime order field Z_p
B_1	bit size of group Gr_1
B_2	bit size of group Gr_2
a	amount of attributes in the system
usr	amount of users in the system
a_u	amount of attributes associated with an user
A_i	amount of attribute authorities
a_r	amount of revoked attributes

B. *Storage Overhead*

Storage overhead represents the storage facilities required for storing the keys, encrypted data's and so on. Table 3 shows the various notations used for estimating the storage overhead. The storage overhead of different entities of our scheme is determined and presented in Table 3 below.

Table 3. Storage overhead

Attribute Authority	Data provider	Data user	Cloud storage server
$2\|B_z\|$	$\|B_z\|+\|B_1\|+A_i(\|B_1\|+\|B_2\|)$	$2a_u\|B_1\|+A_i log(usr)\|B_z\|$	$a(3\|B_1\|+\|B_2\|)+\|B_2\|$

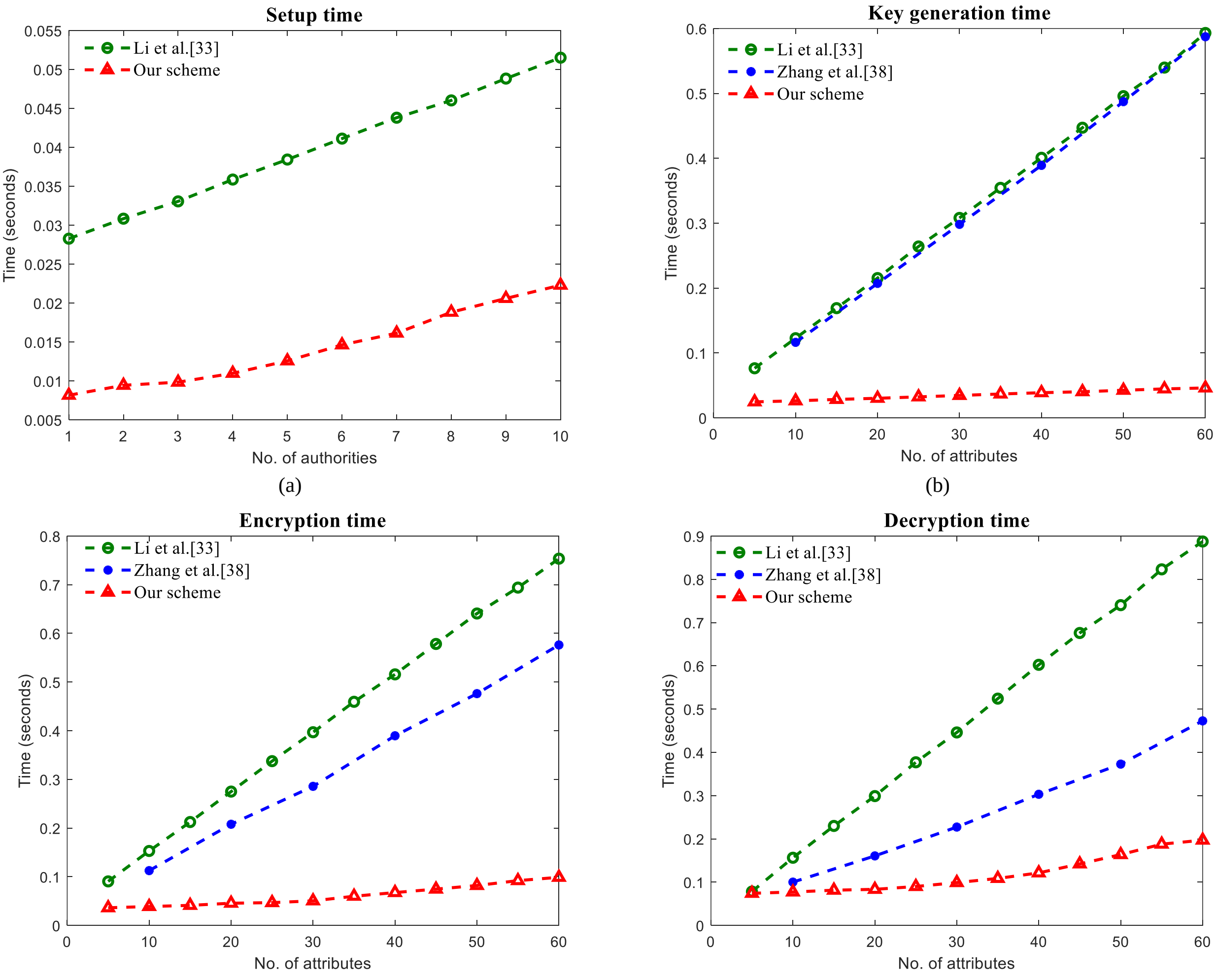
- *Storage overhead of Attribute authorities:* The attribute authorities of our scheme will store the attribute information's associated with their attribute sets. Each attribute contains a private key and a public key and the attribute authorities have to store only the secret keys of the attributes. Thus, the storage overhead of attribute authorities are computed from the attribute secret keys.
- *Storage overhead of Data Providers:* The storage overhead of data providers is computed from the length of public parameters and public key, which is utilized to encrypt the data.
- *Storage overhead of Data users:* The secret keys of attributes in the attribute set determines the storage

overhead of data users. In our method, because of attribute revocation the user has to store path keys along with the secret keys.

- *Storage overhead of Cloud servers:* The length of the ciphertext provides the storage overhead of cloud computing servers.

C. Computation Complexity

The performance of our scheme is experimentally verified on Matlab 2018a software running on Intel core based Windows 8.1 Operating System. The implementation follows an elliptic curve pairing group that is drawn over the algebraic equation $y^2 = x^3 + ax + b \; mod \; p$. The experimental results are then compared with the techniques implemented in [1, 13, 19, 21, 33, 35, 38]. The experimental results are shown in Fig. 3. Fig. 3(a) shows the computation time of setup algorithm for different number of attribute authorities. It is detected that the computation time linearly increases with the amount of attribute authorities. Compared to the scheme in [33], our scheme requires less amount of time for generating the public parameters of universal identities. The elapsed time for secret key generation of our scheme is compared with the schemes in [33,38] in Fig. 3(b). The computation time taken for key generation is noted for different number of attributes. It is clear that the proposed algorithm requires less time for key generation than the existing methods. As shown in Fig. 3(c) and Fig. 3(d), while comparing the time-taken for encryption and decryption, it is detected that the proposed scheme requires less time than the schemes in [33,38]. As shown in Fig. 3(e), due to outsourced decryption, the decryption time is much reduced in our technique than the techniques in [1,13]. As shown in Fig. 3(f), the time efficiency required for tracing the traitor is determined and compared with the traitor tracing algorithms in [19,35]. Our scheme will reduce the amount of time needed to trace the traitor compared to previous methods. Moreover, as visualized in Fig.3(g) and Fig. 3(h), our scheme requires less time for performing secret key updation and ciphertext updation as compared to [21]. Thus, it is clear that the proposed technique performs all the operations within lesser amount of time, and it is better than the existing techniques in terms of computation overhead.

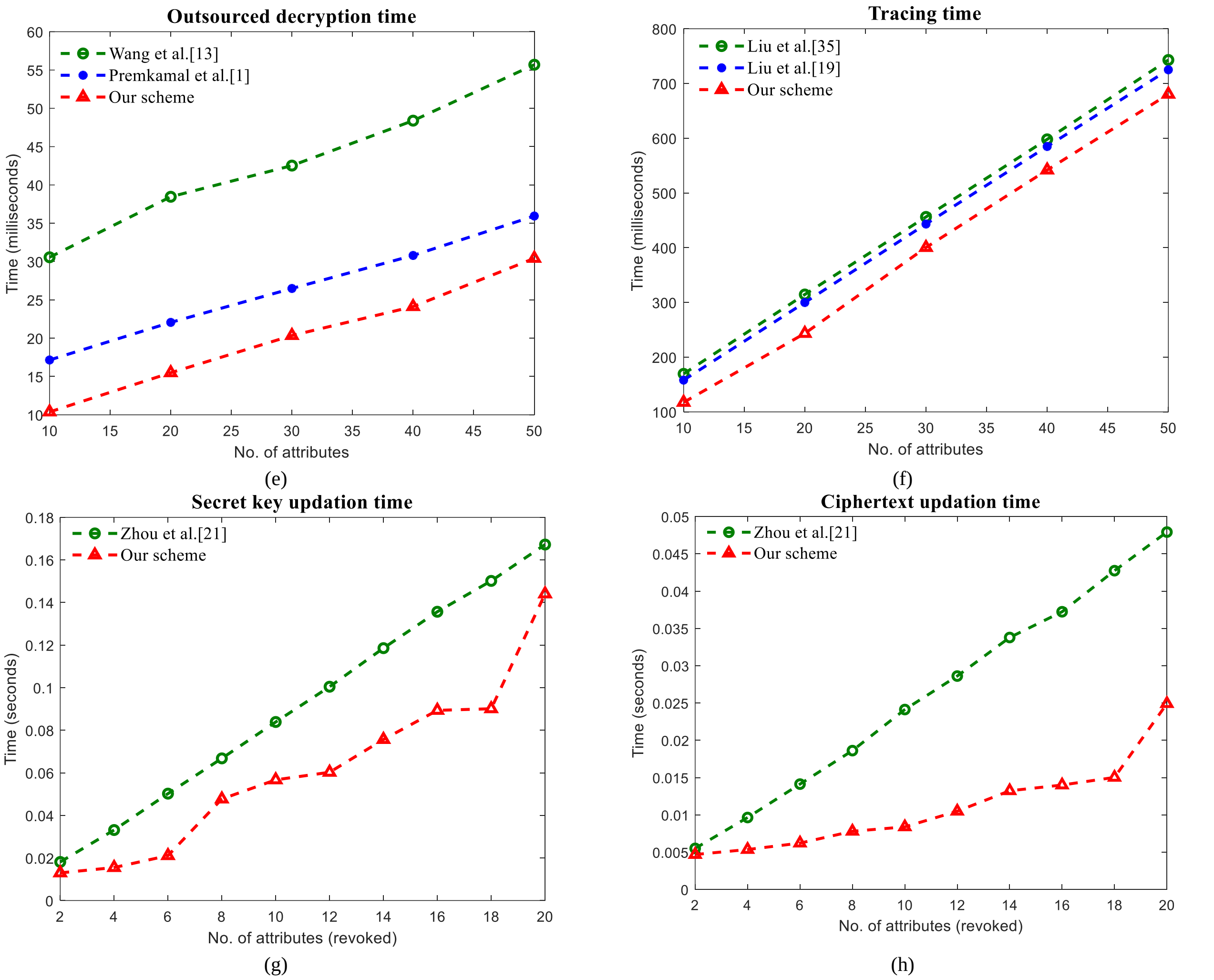


(a)

(b)

(c)

(d)



(e)

(f)

(g)

(h)

Fig.3. Comparison of computation time of: (a) Setup; (b) Key generation; (c) Encryption; (d) Decryption; (e) Outsourced decryption; (f) Traitor tracing; (g) Secret key updation; (h) Ciphertext updation.

7. Conclusions

In this paper, we proposed traitor traceable and revocation-oriented large attribute multi-authority based CP-ABE technique that offers fine-grained data access control and privacy for cloud storage infrastructure. It helps the data providers to store their sensitive data in encrypted form and allows only the authenticated entity to fetch the data. Further, this scheme supports features like outsourced decryption, traitor tracing, attribute revocation and ciphertext updation. With large attribute universe, the attribute space can be dynamically updated at any time, and it maintains constant public parameter length. The computation complexity on decryption is reduced by outsourcing the complex calculations to proxy cloud servers. The malicious user is traced and their respective attributes are revoked. Simultaneously, the ciphertext associated with the revoked attribute is updated without colluding with the unrevoked attribute users. Moreover, the security of our technique is verified and proved in random oracle model. On analysing the computation time complexity, the traitor tracing time of the proposed approach is 10% less than the existing approaches. Likewise, the outsourced decryption time of the proposed approach is 47% less than the existing approaches. Thus, the experimental results show that the proposed technique is efficient in terms of computation complexity. It would be more interesting to construct multi-authority black box traceability based attribute and user revocation scheme supporting large attribute universe and outsourced computation in the future.

Acknowledgment

I confirm that all authors listed on the title page have contributed significantly to the work, have read the manuscript, attest to the validity and legitimacy of the data and its interpretation, and agree to its submission.

Conflict of Interest

The authors declare that they have no conflict of interest.

References

[1] P. K. Premkamal, S. K. Pasupuleti, and P. J. A. Alphonse, “A new verifiable outsourced ciphertext-policy attribute based encryption for big data privacy and access control in cloud,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, pp. 2693-2707, 2018, doi: 10.1007/s12652-018-0967-0.

[2] X. Yongliang, J. Chunhua, Q. Wenyu, S. Jinsong, J. Ying, “Secure fuzzy identity-based public verification for cloud storage,” *Journal of Systems Architecture*, vol. 128, 2022, doi: 10.1016/j.sysarc.2022.102558.

[3] L. Xiaofeng, F. Songbing, J. Cheng, and L. Pietro, “A Fine-Grained IoT Data Access Control Scheme Combining Attribute-Based Encryption and Blockchain,” *Security and Communication Networks*, vol. 2021, 2021, doi: 10.1155/2021/5308206.

[4] K.N. Ambili, and J. Jimmy, “Ensuring Accountability and Outsourced Decryption in IoT Systems using Ciphertext-Policy Attribute-Based Encryption,” Cryptology ePrint Archive, vol. 2021, 2021, https://eprint.iacr.org/2022/040.

[5] P.Chinnasamy, P. Deepalakshmi, A. K. Dutta, J. You, G.P. Joshi, “Ciphertext-Policy Attribute-Based Encryption for Cloud Storage: Toward Data Privacy and Authentication in AI-Enabled IoT System,” *Mathematics*, vol. 10, no. 1, 2022, doi: 10.3390/ math10010068.

[6] S. Ma, J. Lai, R. H. Deng, and X. Ding, “Adaptable key-policy attribute-based encryption with time interval,” *Soft Computing*, vol. 21, no. 20, 2017, pp.6191-6200. doi: 10.1007/s00500-016-2177-z.

[7] J. Li, Q. Yu, Y. Zhang, and J. Shen, “Key-Policy Attribute-Based Encryption against Continual Auxiliary Input Leakage,” *Information Sciences*, vol. 470, 2018, pp. 175-188, doi: 10.1016/j.ins.2018.07.077.

[8] Y. Seongwon, J. K. Eshraghian, H. C. Iu, and K. Cho, “Low-Power Wireless Sensor Network Using Fine-Grain Control of Sensor Module Power Mode,” *Sensors*, vol. 21, no. 9, 2021, https://doi.org/10.3390/s21093198

[9] F. Luo, S. Al-Kuwari, F. Wang, and K. Chen, “Attribute-based proxy re-encryption from standard lattices,” *Theoretical Computer Science*, vol. 865, pp. 52–62, 2021, doi:10.1016/j.tcs.2021.02.036.

[10] S. S. D. Mohd, M. Hussin, Z. M. Hanapi, M. A. Mohamed, “Towards Virtuous Cloud Data Storage Using Access Policy Hiding in Ciphertext Policy Attribute-Based Encryption,” *Future Internet*, vol. 13, no. 11, 2021, doi: 10.3390/fi13110279.

[11] E. G. Hassan, T. Ahmed, “Efficient Ciphertext-Policy Attribute-Based Encryption Constructions with Outsourced Encryption and Decryption,” *Security and Communication Networks*, vol. 2021, 2021, doi: 10.1155/2021/8834616.

[12] H. Kwon, D. Kim, C. Hahn, and J. Hur, “Secure authentication using ciphertext policy attribute-based encryption in mobile multi-hop networks,” *Multimedia Tools and Applications*, vol. 76, no. 19, pp.19507-19521, 2017, doi: 10.1007/s11042-015-3187-z.

[13] S. Luo, “User Privacy Protection Scheme Based on Verifiable Outsourcing Attribute-Based Encryption,” *Security and Communication Networks*, vol. 2021, 2021, doi: 10.1155/2021/6617669

[14] T. Wang, Y. Zhou, H. Ma, R. Zhang, “Flexible and Controllable Access Policy Update for Encrypted Data Sharing in the Cloud,” *The Computer Journal*, 2022, doi: 10.1093/comjnl/bxac024.

[15] Y. Zhang, D. Zheng, and R. H. Deng, “Security and privacy in smart health: efficient policy-hiding attribute-based access control,” *IEEE Internet of Things Journal*, vol. 5, no. 3, pp.2130-2145, 2018, doi: 10.1109/JIOT.2018.2825289

[16] B. Qin, Q. Zhao, and D. Zheng, “Bounded Revocable and Outsourceable ABE for Secure Data Sharing,” *The Computer Journal*, vol. 61, no. 8, pp.1259-1268, 2018, doi: 10.1093/comjnl/bxy063.

[17] P. Li, J. Li, Z. Huang, C. Z. Gao, W. B. Chen, and K. Chen, “Privacy-preserving outsourced classification in cloud computing,” *Cluster Computing*, vol. 21, pp.277-286, 2017, doi: 10.1007/s10586-017-0849-9.

[18] M. Bouchaala, C. Ghazel, and L. A. Saidane, “TRAK-CPABE: A novel Traceable, Revocable and Accountable Ciphertext-Policy Attribute-Based Encryption scheme in cloud computing,” *Journal of Information Security and Applications*, vol. 61, 2021, doi: 10.1016/j.jisa.2021.102914.

[19] Z. Liu, S. Duan, P. Zhou, and B. Wang, “Traceable-then-revocable ciphertext-policy attribute-based encryption scheme,” *Future Generation Computer Systems*, vol. 93, pp. 903-913, 2017, doi: 10.1016/j.future.2017.09.045.

[20] K. Huang, “Secure efficient revocable large universe multi-authority attribute-based encryption for cloud-aided IoT,” *IEEE Access*, vol. 9, pp.53576-53588, 2021, doi: 10.1109/ACCESS.2021.3070907.

[21] J. Zhou, H. Duan, K. Liang, Q. Yan, F. Chen, F. R. Yu, J. Wu, and J. Chen, “Securing outsourced data in the multi-authority cloud with fine-grained access control and efficient attribute revocation,” *The Computer Journal*, vol. 60, no. 8, pp.1210-1222, 2017, doi: 10.1093/comjnl/bxx017.

[22] L. Xue, Y. Yu, Y. Li, M. H. Au, X. Du, and B. Yang, “Efficient attribute-based encryption with attribute revocation for assured data deletion,” *Information Sciences*, vol. 479, pp. 640-650, 2018, doi: 10.1016/j.ins.2018.02.015

[23] H. Zhong, W. Zhu, Y. Xu, and J. Cui, “Multi-authority attribute-based encryption access control scheme with policy hidden for cloud storage,” *Soft Computing*, vol. 22, no. 1, pp.243-251, 2018, doi: 10.1007/s00500-016-2330-8

[24] S. P. P. Kumar, “Enriching Controlled Information Sharing in Healthcare Systems using Attribute based Encryption with Break-Glass Policy,” *Journal of Excellence in Computer Science and Engineering*, vol. 4, no. 2, pp. 35-41, 2018, doi: 10.18831/djcse.in/2018021004

[25] L. J. Xu, R. Hao, J. Yu, and P. Vijayakumar, “Secure deduplication for big data with efficient dynamic ownership updates,” *Computers & Electrical Engineering*, vol. 96, 2021, doi: 10.1016/j.compeleceng.2021.107531

[26] H. Wang, Z. Zheng, L. Wu, and P. Li, “New directly revocable attribute-based encryption scheme and its application in cloud storage environment,” *Cluster Computing*, vol. 20, no. 3, pp.2385-2392, 2017, doi: 10.1007/s10586-016-0701-7

[27] H. Aqeel, and S. T. Ali, “A Provable and User Revocable Ciphertext-Policy Attribute-Based Encryption with Updatable Ciphertext,” in *Innovations in Computer Science and Engineering*, In: Saini, H., Sayal, R., Govardhan, A., Buyya, R. (eds), Singapore, Springer, 2019, pp. 391-399

[28] J. Cui, H. Zhou, H. Zhong, and Y. Xu, “AKSER: Attribute-based keyword search with efficient revocation in cloud computing,” *Information Sciences*, vol. 423, pp.343-352, 2018, doi: 10.1016/j.ins.2017.09.029

[29] M. Chase, “Multi-authority Attribute Based Encryption,” in *Theory of Cryptography*, In: Vadhan, S.P. (eds), Berlin, Heidelberg, Springer, 2007, pp.515-534.

[30] H. Lin, Z. Cao, X. Liang, and J. Shao, “Secure threshold multi authority attribute based encryption without a central authority,”

Information Sciences, vol. 180, no. 13, pp.2618-2632, 2010, doi: 10.1016/j.ins.2010.03.004

[31] X. L. Xu, Q.T. Zhang, and J. L. Zhou, “NC-MACPABE: Non-centered multi-authority proxy re-encryption based on CP-ABE for cloud storage systems,” *Journal of Central South University*, vol. 24, no. 4, pp. 807-818, 2017, doi: 10.1007/s11771-017-3483-z.

[32] S. Belguith, N. Kaaniche, M. Laurent, A. Jemai, and R. Attia, “PHOABE: Securely outsourcing multi-authority attribute based encryption with policy hidden for cloud assisted IoT,” *Computer Networks*, vol. 133, pp.141-156, 2018, doi: 10.1016/j.comnet.2018.01.036.

[33] C. Li, Y. Fang, X. Zhang, C. Jin, Q. Shen, Z. Wu, “A practical construction for large universe hierarchical attribute-based encryption,” *Concurrency and Computation Practice Experience*, vol. 29, no. 17, 2017, doi:10.1002/cpe.3957.

[34] K. Zhang, H. Li, J. Ma, and X. Liu, “Efficient large-universe multi-authority ciphertext-policy attribute-based encryption with white-box traceability,” *Science China Information Sciences*, vol. 61, no. 3, 2018, doi: 10.1007/s11432-016-9019-8.

[35] Z. Liu, Z. Cao, and D. S. Wong, “White-box traceable ciphertext-policy attribute-based encryption supporting any monotone access structures,” *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 1, pp.76-88, 2013, doi: 10.1109/TIFS.2012.2223683.

[36] J. Ning, Z. Cao, X. Dong, and L. Wei, “White-box traceable CP-ABE for cloud storage service: How to catch people leaking their access credentials effectively,” *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 5, pp.883-897, 2018, doi: 10.1109/TDSC.2016.2608343.

[37] D. Tiwari, and G. R. Gangadharan, “SecCloudSharing: Secure data sharing in public cloud using ciphertext-policy attribute-based proxy re-encryption with revocation,” *International Journal of Communication Systems*, vol. 31, no. 5, 2018, doi: 10.1002/dac.3494.

[38] P. Zhang, Z. Chen, J. K. Liu, K. Liang, and H. Liu, “An efficient access control scheme with outsourcing capability and attribute update for fog computing,” *Future Generation Computer Systems*, vol. 78, pp.753-762, 2018, doi: 10.1016/j.future.2016.12.015.

[39] S. Fugkeaw, and H. Sato, “Scalable and secure access control policy update for outsourced big data,” *Future Generation Computer Systems*, vol. 79, pp. 364-373, 2018, doi: 10.1016/j.future.2017.06.014.

[40] J. Sun, Y. Yang, Z. Liu, Y. Qiao, “Multi-Authority Criteria-Based Encryption Scheme for IoT,” *Security and Communication Networks*, vol. 2021, 2021, doi: 10.1155/2021/9174630.

Authors’ Profiles

G. Sravan Kumar received his B.Tech. Degree from JNTUH, Hyderabad, Telangana, India. He obtained his M.Tech. Degree JNTUH, Hyderabad, Telangana India and Persuing Ph.D. Degree from ANU, Guntur, Andhrapradesh, India, he is currently working as an Associate Professor in Sreyas institute of engineering and technology, Hyderabad, Telangana India. His current Research Interests are in Computer networks & Information Security and Algorithms. He can be contacted at Email: Sravanphdcse@gmail.com.

Dr. Atluri Srikrishna received her AMIE (ECE), Institute of Engineers, Kolkata, India. She obtained her M.Tech. Degree JNTUH, Hyderabad, Andhra Pradesh, India and her Ph.D. Degree from JNTUK, Kakinada, Andhra Pradesh, India. She is currently working as a Professor & Head in the Department of Information Technology, R.V.R. & J.C. College of Engineering, Chowdavaram, Guntur, India. Her current Research Interests are in Image Processing & Computer Vision, Information Security and Algorithms. She can be contacted at Email: atlurisrikrishna@gmail.com.

How to cite this paper: G. Sravan Kumar, A. Sri Krishna, "Traitor Traceable and Revocation-oriented Attribute Based Encryption with Proxy Decryption for Cloud Devices", International Journal of Computer Network and Information Security(IJCNIS), Vol.16, No.3, pp.37-51, 2024. DOI:10.5815/ijcnis.2024.03.04