A Privacy Preserving and Efficient Multi Authority – CP-ABE Scheme for Secure Cloud Communication

Shardha Porwal* and Sangeeta Mittal

Department of Computer Science Engineering & Information Technology, Jaypee Institute of Information Technology, Noida, India
E-mail: shardha.porwal@jiit.ac.in; sangeeta.mittal@jiit.ac.in
**Corresponding Author*

Received 08 August 2020; Accepted 14 November 2020;
Publication 06 February 2021

Abstract

In the cloud computing environment, Multi authority Ciphertext Policy-Attribute Based Encryption (CP-ABE) schemes are used as a key escrow free solution to securely and efficiently share data over cloud. However, the length of ciphertext in existing Multi Authority-CP-ABE schemes increases with the number of attributes in the access policy. Moreover, these schemes do not protect against dishonest attribute authorities. In this paper, a constant length ciphertext Multi Authority-CP-ABE scheme is proposed that reduces the communication overhead over the network. The scheme also prevents dishonest authority from compromising the system. Apart from this, for enhanced privacy of receivers, the access policy is communicated in hidden form. Thus, the presented scheme provides an efficient corrupt resistant, key escrow free Multi Authority-CP-ABE scheme by generating constant length ciphertext and hidden access structure. Results demonstrate the enhanced security and reduced cost of encryption and decryption by 8% and 48% respectively as compared to other existing works.

Keywords: MA-CP-ABE, hidden access policy, constant length ciphertext, corrupt resistant, key escrow free.

Journal of Cyber Security and Mobility, Vol. 9_4, 601–626.
doi: 10.13052/jcsm2245-1439.945

1 Introduction

Attribute Based Encryption (ABE) techniques support one-to-many encryption to communicate a confidential data over public network and guarantee fine-grained access control [1]. Based on whether the access policy is attached with secret key or the ciphertext, ABE can be implemented as Key Policy based ABE (KP-ABE) and Ciphertext Policy based ABE (CP-ABE) respectively [2, 3]. In KP-ABE scheme, the ciphertexts and decryption keys of users are associated with set of attributes and access policies respectively. The access control is implemented by key issuer instead of data owner who is encrypting the text, making it lesser flexible and scalable than CP-ABE scheme where encryptor decides access policy for data users. The proposed scheme thus uses CP-ABE based implementation scheme, which was first proposed by Bethencourt et al. [3]. In this scheme, setup and attribute based key generation operations, were performed by a Central Authority (CA). An efficient implementation was later proposed in [4]. The drawbacks of schemes given in [3, 4] and its successors were the problem of key escrow and performance bottleneck due to single authority. To overcome the problem of performance bottleneck, Chase proposed the Multi Authority-CP-ABE (MA-CP-ABE) scheme, in which task of key generation is transferred from CA to multiple Attribute Authorities (AAs) [5]. However, this scheme did not resolve the problem of key escrow. Authors in [6] and [9], proposed MA-CP-ABE schemes without and with CA respectively to resolve the problems of key escrow and performance bottleneck. In both the schemes, CA did not have any control over AAs for attribute management and system worked by assuming that AAs should be honest. Therefore, the scheme is not secure in presence of corrupt or compromised AA. Authors in [12, 16] have proposed schemes, where CA has control over attributes of AAs but their schemes are having problems of key escrow.

Another issue in traditional CP-ABE is that size of ciphertext increases with increase in number of attributes in the access policy. Substantial communication overhead and decryption cost can be reduced if the ciphertext length is made constant, thus improving overall efficiency of the scheme [8]. Apart from this, in conventional CP-ABE, access policy is attached in plaintext along with ciphertext which may reveal the identities of users. This can be addressed by embedding the access policy inside the ciphertext in hidden form [13].

Zhang et al. addressed issues related to efficiency and privacy by proposing fully hidden access structure, constant size ciphertext MA-CP-ABE

scheme [17]. Their scheme suffered from unauthorized access attack where the attribute-based components of ciphertext did not require attribute based secret key of users for decryption. As a result, even the users who do not possess required attributes can also read the encrypted data. Moreover, this scheme also does not handle dishonest AAs. Overall, following security issues still exist in existing schemes:

1. CA is not able to protect corrupt AAs
2. Either CA does not have control on AA or the scheme is not key escrow free.
3. If the scheme is not key escrow free then that is CA is able to access user specific data.
4. If CA does not have control on AA the corrupt authority may provide secret key for unauthorized attributes to a user.

It is thus evident that existing MA-CP-ABE schemes either CA does not have control on attributes of AAs or are not key escrow free [12, 16]. Hence, the main focus of this paper is to improve the efficiency of MA-CP-ABE scheme with hidden access policy by generating constant length ciphertext and to handle dishonest AAs. The proposed work designs a key escrow free MA-CP-AE scheme which is corrupt resistant against AAs and generates constant length ciphertext with hidden access policy. In order to address the security issues, following objectives were set for the proposed work.

1. Design a key escrow free MA-CP-AE scheme which is corrupt resistant against AAs.
2. Efficiency of the scheme should be better as compared to existing scheme.
3. Reduce the storage requirement on cloud server.
4. Implement user anonymity with hidden access policy.

In the proposed scheme, if a corrupt AA generates a secret key for an unauthorized user, the user will not be able to decrypt the ciphertext. The proposed scheme is proved to be secure against Chosen Plaintext Attack (CPA) and its efficiency is found to be better than [14–17].

Three main research contributions have been made through this work.

1. The proposed work presents a corrupt-resistant key escrow free MA-CP-ABE scheme with constant length ciphertext and hidden access policy where CA has control on AAs. Hence, the scheme enhances the security by protecting the system from corrupt AA.

2. The efficiency of proposed scheme is better as compared to existing MA-CP-ABE schemes.
3. Reduced requirement of storage space over cloud due to diminished length of ciphertext.

The paper is organized into six sections. The related work is given in Section 2. Section 3 gives the overview of the proposed scheme. The construction of the proposed scheme is explained in detail in Section 4. The results and analysis are given in Section 5, followed by conclusions in Section 6.

2 Related Work

Attribute Based Encryption (ABE) technique was first proposed by Sahai and Waters [1] in 2005. Goyal et al. [2] proposed Key Policy based implementation of ABE in 2006. Ciphertext Policy based implementation of ABE was first proposed by Bethencourt et al. [3] in 2007. An efficient implementation was later proposed in [4]. The schemes from [1–4] are single authority ABE and have the problem of key escrow and performance bottleneck. The first MA-CP-ABE scheme was proposed by Chase [5] in 2007 but their method was not key escrow free. Lin et al. [6] proposed a key escrow free scheme but did not provide security against dishonest AAs.

Partially hidden access policy with ABE was first proposed by Nishide [7] in 2008. Emura et al. [8] proposed a fully hidden access policy CP-ABE scheme with constant length ciphertext. However, being are single authority ABE, the approaches given in [7] and [8] did not resolve the problem of key escrow and performance bottleneck. The authors in [9] proposed a MA-CP-ABE scheme, in which CA generates a secret key for data consumer but CA does not has control of attributes on AAs and the scheme generates variable length ciphertext with open access structure. Chase and Chow's MA-CP-ABE scheme achieves key-escrow by removing CA [10]. The MA-CP-ABE schemes given in [11] improved the efficiency by generating the constant length ciphertext but did not support hidden access policy. Luo et al. [12] proposed a hierarchical MA-CP-ABE scheme in which CA has control on AAs but the scheme is not key-escrow free.

The MA-CP-ABE schemes given in [13] supported hidden access policy but did not generate constant length ciphertext. The scheme given in [14] is key escrow free MA-CP-ABE scheme that does not supports constant length ciphertext and hidden access policy. Ling and Weng proposed a MA-CP-ABE scheme that generates ciphertext with hidden access structure but the

Table 1 Overview of features of various MA-CP-ABE schemes

Techniques	Key Escrow Free	Presence of Central Authority	Ciphertext Length	Hidden Access Policy	Corrupt Resistant Against AA
(Muller et al. [9]), (Vaanchig et al. [14])	Yes	Yes	No	No	No
(Chase and Chow [10])	Yes	No	No	No	No
(Doshi and Jinwala [11])	Yes	Yes	Yes	No	No
(Luo et al. [12]), (Chandrasekaran et al. [16])	No	Yes	No	No	Yes
(Zhong et al. [13])	Yes	No	No	Yes	No
(Ling and Weng [15])	Yes	Yes	No	Yes	No
(Zhang et al. [17])	Yes	No	Yes	Yes	No
Proposed Scheme	Yes	Yes	Yes	Yes	Yes

length of ciphertext varies with number of attributes in access policy [15]. Chandrasekaran et al. [16] proposed a MA-CP-ABE scheme which improves the efficiency of [12] using fast ate pairing, this scheme also has the problem of key escrow. The MA-CP-ABE scheme given in [17] generates constant length ciphertext and supports hidden access policy but the scheme is not correct as it has unauthorized access attack. In the schemes given in [9–11] and [13–17], CA does not have control over attributes of AA but resolves the problem of performance bottleneck and key escrow. The schemes, given in [12] and [16] handle corrupt resistant against AAs but both are not free from key escrow. Table 1 gives the summary of MA-CP-ABE schemes based on the chosen security parameters.

Challagidad and Birje [18] proposed an efficient MA-CP-ABE scheme for cloud storage. The scheme arranges data users in role hierarchy to have multi level and multi authority access control. The scheme reduces the encryption/decryption cost but does not supports corrupt resistance against AAs. Dixit et al. [19] have proposed a secure access control scheme for authentication of data users and a robust data encryption fuction on the principles of Attribute-Based Access Control (ABAC) in a multi-authority environment. The scheme securily store data on cloud server but does not protect data from unautorized user if he/she get access key for unauthorized attributes from corrupt AAs. Li et al. presented a CP-ABE based access control system model of CloudIoT platform. They constructed a CP-ABE

scheme with hidden acces policy, which guarantees the privacy of the users. However, this scheme considered single authority only.

This paper attempts to overcome the gaps in existing literature and proposes an efficient corrupt resistant key escrow free MA-CP-ABE scheme with constant length ciphertext and hidden access policy, which enhances the security by protecting the system from corrupt AA.

3 Overview of Proposed Scheme

In this section, preliminaries, the communication, security model and workflow of the proposed scheme have been presented.

3.1 Preliminaries

3.1.1 Notations

Notations and their descriptions are given in Table 2.

3.1.2 Bilinear Pairing

Assuming $\mathrm{G}_1, \mathrm{G}_2$ and G_T are cyclic groups of prime order p and $\mathrm{g}_1, \mathrm{g}_2$ are randomly chosen generators of G_1 and G_2 respectively. A mapping function e defined as $\mathrm{e} : \mathrm{G}_1 \times \mathrm{G}_2 \rightarrow \mathrm{G}_\mathrm{T}$, is said to be bilinear mapping if it satisfies the given properties.

- Bilinearity: $\forall \mathrm{x}_1 \in \mathrm{G}_1, \forall \mathrm{x}_2 \in \mathrm{G}_2$ and $\forall \mathrm{y}_1, \mathrm{y}_2 \in \mathrm{Z}_\mathrm{p}$ mapping e satisfies following bilinearity property

$$e(x_1^{y_1}, x_2^{y_2}) = e(x_1, x_2)^{y_1 y_2}$$

- Non-degeneracy: Mapping of $\mathrm{e}(\mathrm{g}_1, \mathrm{g}_2) \neq 1$
- Efficiently computable: $\forall \mathrm{x}_1 \in \mathrm{G}_1, \forall \mathrm{x}_2 \in \mathrm{G}_2$ mapping $\mathrm{e} : \mathrm{G}_1 \times \mathrm{G}_2 \rightarrow \mathrm{G}_\mathrm{T}$ is efficiently computable.

3.1.3 Access Policy (W) and Hidden Access Structure $(\mathbb{A})$

The access policy W is described as logical formula consisting of set of attributes and AND/OR/Threshold gates [3]. The access policy is stored in a data structure called access structure and attached with ciphertext. Assume, a set of n attributes $\mathrm{C} = \{\mathrm{C}_1, \mathrm{C}_2, \ldots, \mathrm{C}_\mathrm{n}\}$ of data consumers then collection $\mathbb{A} \subseteq 2^{\{\mathrm{C}_1, \mathrm{C}_2, \ldots, \mathrm{C}_\mathrm{n}\}}$ is known as monotonic access structure if it satisfies the access policy W, where W is defined as for all Y, Z if $\mathrm{Y} \in \mathbb{A}$ and $\mathrm{Y} \subseteq \mathrm{Z}$ then $\mathrm{Z} \in \mathbb{A}$. A data consumer 'DC' is authorized over W if its attribute set $\mathrm{C} \in \mathbb{A}$

Table 2 Notations and descriptions

Notation	Description
CA	Central Authority
AA	Attribute Authority
DP	Data Provider
CS	Cloud Server
DC	Data Consumer
G_{PUB}	public parameter
G_{MSK}	master key
S_{AA}	Set of attributes assigned to attribute authority AA
$SK_{CA,AA}$	attributes' secret key of Attribute Authority AA received from CA
S_{DC}	Set of attributes of DC
$SK_{CA,DC}$	the secret key of Data Consumer DC received from CA
C_{DC}	Certificate of DC generated by CA
$SK_{AA,DC}$	attributes' secret key of DC received from Attribute Authority AA
Pr_{AA}	The private key of AA
Pk_{AA}	The public key of AA
CT_w	The ciphertext of M for defined policy W
G_1, G_2, G_T	Bilinear groups of prime order p
g_1	Random generator of G_1
g_2	Random generator of G_2
Z_p	Set of integers of prime order p
E	Mapping $G_1 \times G_2 \rightarrow G_T$
u	Universal set of attributes
$D_{i,j}$	Component of $ASK_{CA,AA}$ depends on $att_{i,j} \in S_{AA}$
D_0	Component of $SK_{CA,DC}$ depends on attribute set of DC
$S_{AA,DC}$	$S_{AA} \bigcap S_{DC}$
D_1	Component of $SK_{CA,DC}$ independent of attribute set of DC
$D_{1,AA}$	Component of $ASK_{AA,DC}$ depends on attribute set $S_{AA,DC}$
C_0, C_1	Ciphertext components independent of attributes
C_2	Attributes dependent ciphertext components
I_W^{AA}	Index set of AAs, whose attributes are involved in W

otherwise 'DC' is known as unauthorized user. In our protocol, we have used monotonic access structure and assumed $\mathbb{A}$ is set of only authorized users. The access structure is embedded in the ciphertext such that it remains hidden from other users in the system. In the proposed scheme, only AND gate based access policies have been considered. For OR/THRESHOLD based access policies, the proposed hidden access structure will not work.

3.1.4 The Decisional Bilinear Diffie-Hellman (DBDH) Assumption

The DBDH problem in bilinear groups G_1 is described as given below: on input of the tuple $< g_1, g_1^a, g_1^b, g_1^c, Z >\in G_1$ to decide Z is equals to $e(g_1, g_1)^{abc}$ or Z. A simulator A's advantage is ε if ([probability of A $[g_1, g_1^a, g_1^b, g_1^c, e(g_1, g_1)^{abc}] = 0]$ – [probability of A $[g_1, g_1^a, g_1^b, g_1^c, e(g_1, g_1)^Z] = 0]$) is greater than $\varepsilon(k)$. Here, $e(g_1, g_1)^Z \in G_T/e(g_1, g_2)$. If no probabilistic, polynomial time algorithm has at least ε advantage in solving DBDH problem then it is said that DBDH assumption holds.

3.2 Communication Model

Figure 1 represents the communication model of the proposed scheme. There are five entities, namely Central Authority (CA), Attribute Authority (AA), Data Provider (DP), Cloud Server (CS) and Data Consumer (DC).

Central Authority (CA): CA is a fully trusted entity and has control over attributes assigned to each AA. CA generates public parameter G_{PUB} for every communicating entity and global master key G_{MSK}. It then assigns attributes S_{AA} to every AA by generating and distributing attribute set based

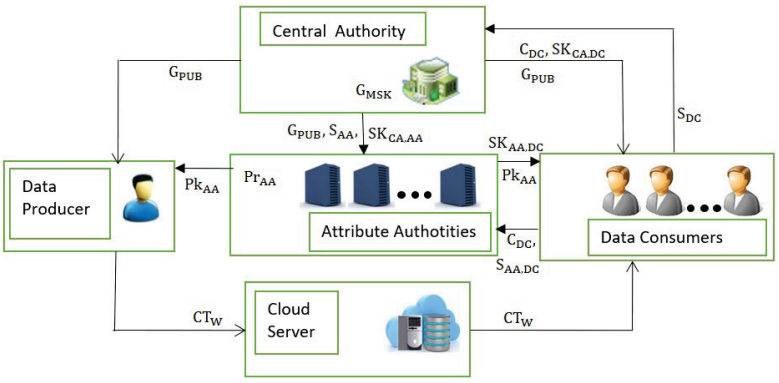

Figure 1 System model.

secret key $\mathrm{SK}_{\mathrm{CA,DC}}$. CA generates a secret key $\mathrm{SK}_{\mathrm{CA,DC}}$ and certificate C_{DC} for every DC based on a set of attributes S_{DC} of DC.

Attribute Authority (AA): AA is a semi-trusted entity and generates its public key $\mathrm{Pk}_{\mathrm{AA}}$ and private key $\mathrm{Pr}_{\mathrm{AA}}$. AA also generates attribute dependent secret key $\mathrm{SK}_{\mathrm{AA,DC}}$ for every DC.

Data Producer (DP): DP generates data M, encrypts M using symmetric encryption key k_{M}. To share k_{M} to data users, DP encrypts k_{M} using global public key $\mathrm{G}_{\mathrm{PUB}}$, attribute authorities public key $\mathrm{Pk}_{\mathrm{AA}}$, access policy W and generates the ciphertext CT_{w}. Finally, DP stores encrypted M and CT_{W} on the CS.

Cloud Server (CS): CS is considered an untrusted entity that stores the ciphertext CT_{w}.

Data Consumer (DC): DC accesses the ciphertext CT_{w} stored on CS and decrypts CT_{w} to get M using a secret key $\mathrm{SK}_{\mathrm{CA,DC}}$ received from CA and secret key $\mathrm{SK}_{\mathrm{AA,DC}}$ received from AA. Data will be correctly retrieved only if DC is a valid data consumer.

3.3 Security Model

To prove the CPA security of the proposed scheme, the adversary A, and the challenger C play the following selective security game.

The CPA Security game for the proposed scheme

Init: C receives the challenging access policy $\mathrm{A}*$ from A.

CA_Setup: C runs CA_Setup() and generates $\mathrm{G}_{\mathrm{PUB}}$ for A.

AA_Registration: C runs AA_Registration() and generates $\mathrm{SK}_{\mathrm{CA,AA}}$

AA_Setup: C runs AA_Setup() and generates $\mathrm{Pk}_{\mathrm{AA}}$ for A.

Phase 1:

• Query:

A provides S_{A} and queries adaptively for secret key $\mathrm{SK}_{\mathrm{CA,DC}}$ and $\mathrm{SK}_{\mathrm{AA,DC}}$ such that $\mathrm{S}_{\mathrm{A}} \notin \mathrm{A}*$ from CA.

Challenge: A provides plaintext data $\{\mathrm{m}_0, \mathrm{m}_1\}$ to the C. C randomly chooses $\mathrm{v} \in \{0, 1\}$ and encrypts m_{v} using the scheme access policy $\mathrm{A}*$ and provides ciphertext CT to A.

Phase 2: repeat Phase 1.

Guess: A randomly chooses $\mathrm{v}' \in \{0, 1\}$. The gain that A has in this security game would be $\Pr[\mathrm{v} = \mathrm{v}*] = \frac{\varepsilon}{2}(1 - \frac{\mathrm{N}^2}{\mathrm{p}})$, where N is $\prod_{\mathrm{j}=1}^{\mathrm{n}} \mathrm{n}_{\mathrm{j}}$.

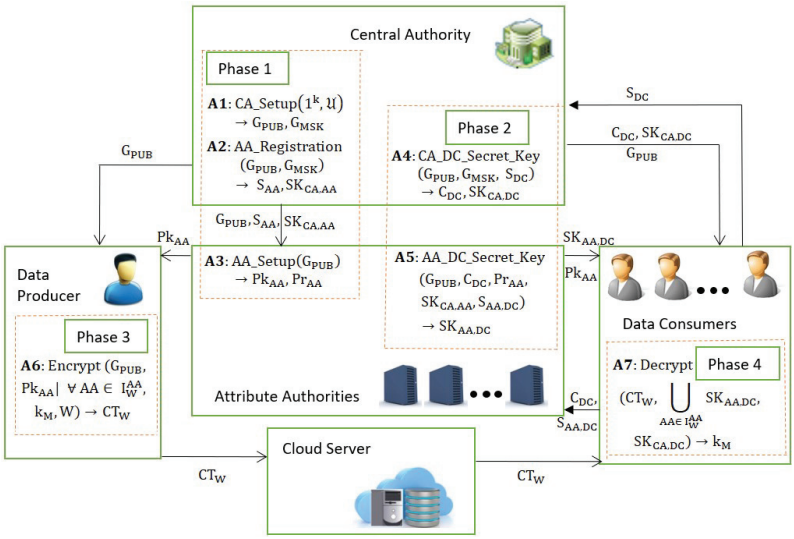

Figure 2 Flow diagram of the proposed scheme.

3.4 Workflow

As shown in flow diagram given in Figure 2, the scheme proposes seven algorithms, A1 to A7 explained in detail in section IV. These algorithms execute in four phases namely, System initialization, Attribute's key generation for data consumer, data encryption and decryption. DC receives a key $\mathrm{SK}_{\mathrm{CA,DC}}$ from CA which is to be used for decryption. This key protects from granting unauthorized data access by dishonest AAs. Step by step communication among entities to securely share the data using cloud storage is as follows:

1. $\mathrm{CA} \rightarrow \mathrm{DCs}, \mathrm{DPs}, \mathrm{AAs} : \mathrm{G_{PUB}}$
2. $\mathrm{CA} \rightarrow \mathrm{AA} : \mathrm{S_{AA}}, \mathrm{SK_{CA,AA}}$
3. $\mathrm{AA} \rightarrow \mathrm{DPs}, \mathrm{DCs} : \mathrm{Pk_{AA}}$
4. $\mathrm{DC} \rightarrow \mathrm{CA} : \mathrm{S_{DC}}$
5. $\mathrm{CA} \rightarrow \mathrm{DC} : \mathrm{SK_{CA,DC}}, \mathrm{C_{DC}}$
6. $\mathrm{DC} \rightarrow \mathrm{AA} : \mathrm{S_{AA,DC}}, \mathrm{C_{DC}}$
7. $\mathrm{AA} \rightarrow \mathrm{DC} : \mathrm{SK_{AA,DC}}$
8. $\mathrm{DP} \rightarrow \mathrm{CS} : \mathrm{CT_{\mathbb{A}}}$
9. $\mathrm{CS} \rightarrow \mathrm{DCs} : \mathrm{CT_{\mathbb{A}}}$

As shown in Figure 4, phase 1 consists of algorithms A1 to A3, phase 2 consists algorithm A4 and A5, phase 3 consists A6 and phase 4 consists algorithm A7. CA executes algorithm A1: $\mathrm{CA\_Setup}(1^{\mathrm{k}}, \mathrm{u})$ and generates

G_{PUB}, G_{MSK}. CA provides G_{PUB} to all other entities DCs, DPs, AAs in step 1.

Each AA has to register with CA, CA executes algorithm A2: AA_Registration(G_{PUB}, G_{MSK}) and provides $S_{AA}, SK_{CA,AA}$ to AA in step 2.

After receiving G_{PUB}, each AA executes algorithm A3: $AA\_Setup(G_{PUB})$ to generates its private key Pr_{AA}, public key Pk_{AA} and as shown in step 3, AA gives Pk_{AA} to DPs, DCs.

Every DC is also required to register with CA and provides S_{DC} to CA in step 4.

Phase 2 is about attribute's key generation for data consumer in which CA executes algorithm A4: $CA\_DC\_Secret\_Key(G_{PUB}, G_{MSK}, S_{DC})$ in step 5 and provides $C_{DC}, SK_{CA,DC}$ to DC. DC gives $S_{AA,DC}, C_{DC}$ to AA in step 6 and then in step 7 AA generates the attribute key $SK_{AA,DC}$ for attribute set $S_{AA,DC}$ by executing algorithm A5: $AA\_DC\_Secret\_Key(G_{PUB}, C_{DC}, Pr_{AA}, SK_{CA,AA}, S_{AA,DC})$ which contains the attributes common to AA and DC.

To communicate the data, DP executes algorithm A6: $Encrypt(G_{PUB}, Pk_{AA} | \forall AA \in I_W^{AA}, k_M, W)$ of phase 3 and generates the ciphertext CT_w and store it in the cloud in step 8.

To access the stored ciphertext CT_w, DC executes the algorithm A7: $Decrypt(CT_w, \cup_{AA \in I_W^{AA}}, SK_{AA,DC}, SK_{CA,DC})$ of phase 4 and get the decrypted data in step 9.

4 Construction of the Proposed Scheme

The proposed scheme works in four phases, namely system initialization, attributes key generation of data consumer, encryption and decryption.

Phase 1: System Initialization

This phase initializes the system by executing algorithm A:CA_Setup(), A2:AA_Registration(), A3:AA_Setup() and A4:CA_Dc_Secret_Key().

A1: CA_Setup$(1^k, u) \rightarrow G_{PUB}, G_{MSK}$:
//Central Authority (CA) *executes the algorithm to generate public parameters* (G_{PUB}) *and master key* (G_{MSK}). *The public parameters are published globally and the master key is kept as secret key of CA*.

This algorithm is executed by (CA) to set system parameters G_{PUB}, G_{MSK}. The algorithm requires a security parameter k and universal

set of attributes $u = \{att_1 \ldots att_N\}$ as input. Here, N is the total number of attributes in u. Each attribute att_1 may have M different values $att_{i,1}, \ldots, att_{i,M}$. The algorithm selects generators $g_1 \in_R G_1$ and $g_2 \in_R G_2$ as well as integers $\alpha, r \in_R Z_p$ and set of numbers $t_{i,j}, r_{i,j} \in_R Z_p$, where $1 \leq i \leq N, 1 \leq j \leq M$. $T_{i,j} = g_1^{t_{i,j}} \forall att_{i,j}$, where $1 \leq i \leq N, 1 \leq j \leq M$ is also initialized. Finally, public parameter, G_{PUB} obtained as in per Equation (1) and master key G_{MSK} as per Equation (2).

$$G_{PUB} = \{g_1, g_2, u, Y = e(g_1, g_2)^{\alpha}, T_{i,j} \forall att_{i,j}, 1 \leq i \leq N, 1 \leq j \leq M\} \tag{1}$$

$$G_{MSK} = \{\alpha, r, t_{i,j}, r_{i,j} \forall att_{i,j}, 1 \leq i \leq N, 1 \leq j \leq M\} \tag{2}$$

Master key G_{MSK} is retained by CA only and it publishes the public parameter G_{PUB} globally.

A2: AA_Registration$(G_{PUB}, G_{MSK}) \rightarrow S_{AA}, SK_{GA,AA}$
// The algorithm is executed by CA to assign attributes and corresponding secret keys to attribute authority.

This algorithm is executed by (CA), which generates a set of attributes S_{AA} and attributes' secret key $SK_{CA,AA}$ for attribute authority AA. The algorithm takes master key G_{MSK}, public parameter G_{PUB} as input. The algorithm assigns a set of attributes S_{AA} to AA and defines attributes' secret key $SK_{CA,AA}$ for AA as given in Equation (3).

$$SK_{CA,AA} = \{S_{AA}, \forall att_{i,j} \in S_{AA} D_{i,j} = g_2^{t_{i,j} r + r_{i,j}}\} \tag{3}$$

A3: AA_Setup$(G_{PUB}) \rightarrow Pk_{AA}, Pk_{AA}$:
//AA runs the algorithm to generate its public key and private key.

This algorithm is run by AA to generate AA's public key and private key. The algorithm takes public parameter G_{PUB} as input. The algorithm chooses $\alpha_{AA} \in_R Z_p$ and generates AA's private key Pr_{AA} and public key Pk_{AA} as per Equations (4) and (5).

$$Pr_{AA} = \{\alpha_{AA}\} \tag{4}$$

$$Pk_{AA} = \{Y_{AA} = e(g_1, g_2)^{\alpha_{AA}}\} \tag{5}$$

Phase2: Attribute set based key generation for the data consumer

This phase generates the secret key for different data consumers according to their attribute set.

A4: CA_DC_Secret_Key$(G_{PUB}, G_{MSK}, S_{DC}) \rightarrow C_{DC}, SK_{CA,DC}$:
//Central Authority runs the algorithm to compute secret key and to generate

certificate for every data consumer.

$$\mathrm{SK_{CA,DC}} = \{\mathrm{D_0} = \mathrm{g_2^{\alpha - r'}}, \mathrm{D_1} = \mathrm{g_2^{r}}\} \tag{6}$$

A5: AA_DC_Secret_Key$(\mathrm{G_{PUB}}, \mathrm{C_{DC}}, \mathrm{Pr_{AA}}, \mathrm{SK_{CA,AA}}, \mathrm{S_{AA,DC}})$
$\rightarrow \mathrm{SK_{CA,DC}}$:
//Attribute Authority executes the algorithm to generate the secret key for every data consumer.

AA executes this algorithm, which generates attribute secret key $\mathrm{SK_{AA,DC}}$ for data consumer DC. The algorithm first verifies $\mathrm{C_{DC}}$ to check the authenticity of DC. If DC is an authenticated data consumer, then the algorithm defines attribute based secret key $\mathrm{SK_{AA,DC}}$ for an attribute set $\mathrm{S_{AA,DC}}$ as Equation (7).

$$\mathrm{SK_{AA,DC}} = \left\{ \mathrm{D_{1,AA}} = \mathrm{g_2^{\alpha_{AA}}} \cdot \prod_{\mathrm{att_{i,j}} \in \mathrm{S_{AA,DC}}} \mathrm{D_{i,j}} \right\} \tag{7}$$

Phase3: Data encryption
This phase shows the encryption operation of the proposed scheme.

A6: Encrypt$(\mathrm{G_{PUB}}, \mathrm{Pk_{AA}} | \forall \mathrm{AA} \in \mathrm{I_W^{AA}}, \mathrm{k_M}, \mathrm{W}) \rightarrow \mathrm{CT_E}$:
//Data producer executes the algorithm to encrypt the data to be communicated to specific data consumers over the network and stores on the cloud.

Data Producer (DP) executes this algorithm to encrypt the data $\mathrm{k_M}$ to be communicated. The algorithm takes input the public parameters $\mathrm{G_{PUB}}$, public key $\mathrm{Pk_{AA}}$ of all AAs whose attributes are involved in specifying access policy W represented as $\mathrm{I_W^{AA}}$ and access policy W. The algorithm encrypts data $\mathrm{k_M} \in \mathrm{G_T}$ under access policy W, where W consists AND gate. The algorithm generates the ciphertext $\mathrm{CT_W}$ as follows.

The algorithm selects an integer $\mathrm{s} \in_{\mathrm{R}} \mathrm{Z_p}$ and defines $\mathrm{C_0}, \mathrm{C_1}, \mathrm{C_2}$ as Equations (8)–(10).

$$\mathrm{C_0} = \mathrm{g_1^s} \tag{8}$$

$$\mathrm{C_1} = \mathrm{k_M} \cdot \mathrm{Y^s} \cdot \left(\prod_{\mathrm{AA} \in \mathrm{I_W^{AA}}} \mathrm{Y_{AA}^s} \right) \tag{9}$$

$$\mathrm{C_2} = \prod_{\forall \mathrm{att_{i,j}} \in \mathrm{W}} (\mathrm{T_{att_{i,j}}})^{\mathrm{s}} \tag{10}$$

The algorithm defines the ciphertext $\mathrm{CT_w}$ as follows:

$$\begin{aligned}\mathrm{CT_w} &= \{\mathrm{C_0} = \mathrm{g_1^s}\},\\ \mathrm{C_1} &= \mathrm{k_M} \cdot \mathrm{Y^s} \cdot \left(\prod_{\mathrm{AA} \in \mathrm{I_W^{AA}}} \mathrm{Y_{AA}^s} \right),\\ \mathrm{C_2} &= \prod_{\forall \mathrm{att_{i,j}} \in \mathrm{W}} (\mathrm{T_{att_{i,j}}})^{\mathrm{s}} \end{aligned} \tag{11}$$

Finally, DP stores the ciphertext $\mathrm{CT_W}$ on the CS.

Phase4: Data decryption
This phase shows the decryption operation of the proposed scheme to reconstruct the data from the ciphertext.

A7: Decrypt$(\mathrm{CT_W}, \bigcap_{\mathrm{AA} \in \mathrm{I_W^{AA}}} \mathrm{SK_{AA,DC}}, \mathrm{SK_{CA,DC}}) \rightarrow \mathrm{k_M}$:
//Data consumer runs the algorithm to get the content key of a data file stored on the cloud.

This algorithm is executed by DC to reconstruct the data $\mathrm{k_M}$. The algorithm takes input ciphertext $\mathrm{CT_W}$, attribute secret keys $\mathrm{SK_{AA,DC}} | \forall \mathrm{AA} \in \mathrm{I_W^{AA}}$ and secret key $\mathrm{SK_{CA,DC}}$ and determines $\mathrm{k_M}$ using Equation (12) if DC has attributes specified in W.

$$\mathrm{k_M} = \frac{\mathrm{C_1} \cdot \mathrm{e(C_2, D_1)}}{\mathrm{e(C_0, D_0)} \cdot \prod_{\mathrm{AA} \in \mathrm{I_W^{AA}}} \mathrm{e(C_0, D_{1,AA})}} \tag{12}$$

Verifiability against corrupt AA

In the proposed scheme, DC receives a key $\mathrm{SK_{CA,DC}}$ from CA which is used for decryption. This $\mathrm{SK_{CA,DC}}$ is constructed as $\{\mathrm{D_0} = \mathrm{g_2^{\alpha - r'}}, \mathrm{D_1} = \mathrm{g_2^r}\}$, where $\mathrm{r'} = \sum_{\mathrm{att_{i,j} \in S_{DC}}} (\mathrm{r_{i,j}})$ and $\mathrm{r_{i,j}}$, r are components of $\mathrm{G_{MSK}}$. DC receives another key $\mathrm{SK_{AA,DC}}$ from AA which is also used for decryption and is constructed as $\{\mathrm{g_2^{\alpha_{AA}}} \cdot \prod_{\mathrm{att_{i,j} \in S_{AA,DC}}} \mathrm{D_{i,j}}\}$, where $\mathrm{D_{i,j}} = \mathrm{g_2^{t_{i,j} \cdot r + r_{i,j}}}$ and $\mathrm{r_{i,j}}$, r, $\mathrm{t_{i,j}}$ are components of $\mathrm{G_{MSK}}$. If a corrupt AA generates $\mathrm{SK_{AA,DC}}$ for those attributes for which DC is not authorized then at decryption both $\mathrm{SK_{AA,DC}}$, $\mathrm{SK_{AA,DC}}$ are required and DC will not be able to decrypt the ciphertext. Hence, the key $\mathrm{SK_{CA,DC}}$ protects from granting unauthorized data access by dishonest AAs.

5 Results and Analysis

The proposed scheme has important feature of corrupt resistant against attribute authorities. The security analysis proves that the presented scheme is secure against the Chosen Plaintext Attack. This section represents a comparative analysis of the security features, theoretical complexities and computational performance of the presented scheme with the schemes given in [14–17].

5.1 Security Analysis

The proposed scheme is proved to be secure against Chosen Plaintext Attack (CPA).

Proof:
The scheme is proved to be CPA secure by reducing the security of the proposed scheme to Decisional Bilinear Diffie Hellman (DBDH) assumption.

Theorem: *Assuming the DBDH assumption holds then the proposed scheme satisfies the indistinguishability of data and chosen-plaintext attack.*

Proof:
Assuming adversary A wins the following CPA game for the proposed scheme with gain ε. Then an algorithm B is constructed to break the assumption with gain $\frac{\varepsilon}{2}(1-\frac{\mathrm{N}^2}{\mathrm{P}})$, where N is $\prod_{\mathrm{j}=1}^{\mathrm{n}} \mathrm{n_j}$.

C randomly chooses generators $\mathrm{g_1} \in \mathrm{G_1}$, $\mathrm{g_2} \in \mathrm{G_2}$, integer values $\mathrm{a, b, c, x} \in \mathrm{Z_p}$ and v belongs to $\{0, 1\}$. If $\mathrm{v} = 0$ then C sets $\mathrm{X} = (\mathrm{g_1}, \mathrm{g_2})^{\mathrm{abc}}$ else if $\mathrm{v} = 1$ then C sets $\mathrm{X} = \mathrm{e}(\mathrm{g_1}, \mathrm{g_2})^{\mathrm{x}}$. Then C submits instance $< \mathrm{g_1}, \mathrm{g_2}, \mathrm{A} = \mathrm{g_1^a}, \mathrm{B} = \mathrm{g_2^b}, \mathrm{C} = \mathrm{g_1^c}, \mathrm{X} >$ to B.

Init: B receives the challenging access policy $\mathbb{A}*$ from A. Assuming $\mathbb{A}* = \{\mathbb{A}*_1, \mathbb{A}*_2, \ldots, \mathbb{A}*_{\mathrm{n}}\}$.

CA_Setup: B randomly chooses $\mathrm{u, r} \in \mathrm{Z_p}$ and computes $\mathrm{h} = \mathrm{B^u} = \mathrm{g_2^{ub}}, \mathrm{Y} = \mathrm{e(A, h)} = \mathrm{e}(\mathrm{g_1^a}, \mathrm{g_2^{ub}}) = (\mathrm{g_1}, \mathrm{g_2})^{\mathrm{uab}}$ also chooses random numbers $\mathrm{t'_{i,j}}, \mathrm{r'_{i,j}} \in \mathrm{Z_p}$ for each attribute, where $\mathrm{i} \in [1 \ldots \mathrm{n}], \mathrm{j} \in [1 \ldots \mathrm{m}]$ and sets $\mathrm{t_{i,j}} = \mathrm{t'_{i,j}}$ and $\mathrm{r_{i,j}} = \mathrm{r'_{i,j}}$ if $\mathrm{att_{i,j}} = \mathbb{A}*_{\mathrm{i}}$. Then B computes $\mathrm{T_{i,j}} = \mathrm{g_1^{t_{i,j}}}$ and sets public parameters $\{\mathrm{Y}, \mathrm{T_{i,j_{i\in[1\ldots n],j\in[1\ldots m]}}}\}$ and gives to A.

AA_Registration:

B sets secret key $\mathrm{SK_{CA,AA}}$ for each $\mathrm{att_{i,j}}$ assigned to AA.

$$SK_{CA,AA} = \{D_{1,j} = g_2^{t_{1,j} \cdot r + r_{i,j}} \forall att_{i,j} \in S_{AA_k}\}$$

AA_Setup: B randomly chooses $u_{AA} \in Z_p$ and computes $h_{AA} = B^{u_{AA}} = g_2^{u_{AA}b}$, $Y_{AA} = e(A, h_{AA}) = e(g_1^a, g_2^{u_{AA}b})$. Then B provides, public key $\{Y_{AA}\}$ to A.

Phase 1:

• Query:

To get the secret key $SK_{CA,A}$, A provides S_A to B, where A is not satisfied by S_A. Then the A's attribute $att_{i,j}$ must $\in$ u, in such a way that $\mathbb{A}*$ is not satisfied by S_A. B determines such $att_{i,j}$. For each $att_{i,j} \in S_A$, B randomly chooses $r'_{i,j}$ and sets $r_i = ab + r'_{i,j}b$. Finally, B computes $r' = \sum_{j=1}^{n} r_i = ab + \sum_{j=1}^{n} r_i b$. The secret key component of $SK_{CA,A}$ is generated as $\prod_{j=1}^{n} \frac{1}{B^{r_{att_{j,2}}}} = g^{-\sum_{j=1}^{n} r_{att_{j,2}} \cdot b} = g^{ab-r'}$.

For getting the other component of $SK_{CA,A}$, attribute secret key $SK_{CA,A}$, A queries adaptively with S_A to B. Since $\mathbb{A}*$ is not satisfied by $S_A, S_{A_i} \wedge att_{i,j} \neq \mathbb{A}_i*$, hence, $\sum_{att_{i,j} \in S_A} t_{i,j}$ can be denoted as $T_1 + bT_2 (T_1, T_2 \in Z_p)$ and $\sum_{att_{i,j} \in S_A} r_{i,j}$ can be denoted as $T_3 + bT_4 (T_3, T_4 \in Z_p)$. B determines T_1 and T_2 from $\sum_{att_{i,j} \in S_A} t_{i,j}$ and T_3 from $\sum_{att_{i,j} \in S_A} r_{i,j}$ and randomly chooses $\beta \in Z_p$, defines $r = \frac{\beta - ua}{T_2}$ and provides other components of secret key as $((g_2^b)^\beta g_2^{\beta \frac{T_1}{T_2}} (g_2^a)^{\frac{T_1 u}{T_2}} g_2^{T_3} (g_2^b)^{T_4}, g_2^{\frac{\beta}{T_2}} (g_2^a)^{\frac{u}{T_2}})$. The validity of these components of secret key is shown as follows:

$$
\begin{aligned}
(g_2^b)^\beta g_2^{\beta \frac{T_1}{T_2}} (g_2^a)^{\frac{T_1 u}{T_2}} g_2^{T_3} (g_2^b)^{T_4} &= g_2^{uab} g_2^{-uab} (g_2^b)^\beta g^{\beta \frac{T_1}{T_2}} (g_2^a)^{\frac{T_1 u}{T_2}} g_2^{T_3} (g_2^b) T_4 \\
&= g_2^{uab} (g_2)^{(\beta - ua)\frac{T_1}{T_2}} (g_2^a)^{(\beta - ua)b} g_2^{T_3} (g_2^b) T_4 \\
&= g_2^{uab} (g_2^{T_1} \cdot g_2^{bT_2})^{\frac{(\beta - ua)}{T_2}} \cdot (g_2^{T_2} \cdot g_2^{bT_4}) \\
&= g_2^{uab} (g_2^{T_1 + bT_2})^{\frac{(\beta - ua)}{T_2}} \cdot (g_2^{T_3 + bT_4}) \\
&= g_2^{y} \left(g_2^{\sum_{att_{i,j} \in S_A} t_i} \right)^r \cdot \left(g_2^{\sum_{att_{i,j} \in S_A} r_i} \right) \\
&= g_2^{y} \left(g_2^{\sum_{att_{i,j} \in S_A} t_{i,j} \cdot r + \sum_{att_{i,j} \in S_A} r_{i,j}} \right)
\end{aligned}
$$

and

$$
g_2^{\frac{\beta}{T_2}} (g_2^a)^{-\frac{u}{T_2}} = g_2^{\frac{(\beta - ua)}{T_2}} = g_2^r.
$$

If T_2 equals to (0 mod p) holds, then the maximum probability for some S_A, such that $\sum_{att_{i,j} \in S_A} t_{i,j} = \sum_{att_{i,j} \in \mathbb{A}*} t_{i,j}$ is $\frac{N^2}{P}$.

• Challenge: B randomly chooses $g \in \{0,1\}$ and generates ciphertext $\{C_1 = m_g \cdot X^u \cdot \prod_{AA \in [1 \ldots n_{\mathbb{A}*}]} Y_{AA}, C_2 = g_1^c, C_3 = \prod_{att_{i,j} \in \mathbb{A}*} (g_1^c)^{t_i}\}$ for A.

Phase 2: Same as phase1.
Guess: A chooses $\wp'$ from $\{0,1\}$. B sets result $= 1$ if $\wp'$ is equals to $\wp$otherwise, B set result $= 0$. The ciphertext (C_1, C_2, C_3) is valid for access policy $\mathbb{A}*$, if $X = e(g_1, g_2)^{abc}$ and gain to A is ε. Therefore, $Prob[B \rightarrow 1|X = e(B_1, B_2)^{abc}]$ is equals to $Prob[\wp' = \wp|X = e(g_1, g_2)^{abc}]$ is equals to $\frac{1}{2} + \varepsilon$. Otherwise if $X = e(g_1, g_2)^X$ then, gain to A is and $Prob[B \rightarrow 0|X = e(g_1, g_2)^X]$ is equals to $Prob[\wp' \neq \wp|X = e(g_1, g_2)^x]$ is equals to $\frac{1}{2}$. Hence, B's gain is $\frac{\varepsilon}{2}(1 - \frac{N^2}{p})$ in this game.

Hence proved.

5.2 Security Features Analysis

Table 3 provides the comparison of security features of the presented scheme and schemes given in [14–17]. A comparative analysis of features is provided on the bases of parameters given in Table 4. Key-escrow free feature of MA-CP-ABE protects accessing of the user's data from CA or AAs, Corrupt resistant from AA protects unauthorized access of data in case of illegal key generation by AAs. Constant length ciphertext reduces the storage space on the cloud, unauthorized attribute key distribution is protected through control of CA on AAs. The scheme given in [14] is key escrow free, MA-CP-ABE and provides protection against unauthorized access. Scheme given in [15] has features of [14] and provides anonymity though hidden access structure, also verify user at decryption end. In the scheme given in [16], CA generates attribute keys for AAs, here AAs are controlled though CA but the scheme has issue of key escrow. Zhang et al. in [17] proposed CA free MA-CP-ABE, their scheme generates constant length ciphertext and protects user privacy using a hidden access policy. The proposed scheme has all the required features of [14–17] and protects the scheme from unauthorized access exists in [17]. CA gives a part of secret key to data consumer which is attribute dependent and is required at the time of decryption. Even if a dishonest AA gives an attribute key for the attributes that DU does not have, then also DU is unable to decrypt the encrypted data.

5.3 Theoretical Complexities

The symbols used in the comparison of theoretical complexities are given in Table 4.

Table 3 Comparison of security features of the proposed scheme with the schemes given in [14–17]

Security Feature	[14]	[15]	[16]	[17]	The Proposed Scheme
Key Escrow Free	Yes	Yes	No	Yes	Yes
Corrupt resistant from AA	No	No	Yes	No	Yes
Constant length ciphertext	No	No	No	Yes	Yes
CA controls AAs	No	No	Yes	No	Yes
AA's attributes	Generated by AAs	Generated by AAs	Generated by CA	Generated by AAs	Generated by CA
DC's keys	Generated by AAs and CA	Generated by respec-tive AAs	Generated by respec-tive AAs	Generated by respec-tive AAs	Generated by CA and AAs both
Prevention of dishonest AA	No	No	No	No	Yes
Protection against unauthorized access	Yes	Yes	Yes	No	Yes
Hidden access policy	No	Yes	No	Yes	Yes
Verification of user	No	Verification at CA	No	No	Verification at AAs

Table 5 shows the comparisons of theoretical complexities of the schemes given in [14–17] and the proposed scheme in terms of theoretical space and time complexity. The theoretical space complexity of secret key of DC in [14–17] is larger than the proposed scheme because they generate variable-length secret key that varies according to number of attributes of data consumer while the size of secret key of proposed scheme is constant and very less as shown in Table 5. The theoretical space complexity of ciphertext is also less in the proposed scheme as only three ciphertext components are generated. The theoretical time complexity of encryption operation of [17] requires extra cost for pairing operations that varies with the number of attributes in access policy $|\mathrm{P}|$ and scheme given in [14, 16] require cost for access structure generation. The scheme given in [15] requires more exponentiation in encryption than the proposed scheme. Hence encryption costs of the

Table 4 Symbols Used in Theoretical Complexity analysis

Symbol	Descriptions
$\lvert N_{AA}\rvert$	Number of AAs involved in generating SK or CT
$\lvert S_{DC}\rvert$	Number of attributes of Data Consumer
$\lvert P\rvert$	Length of access policy in terms of the number of attributes
CE_G	Exponentiation cost in G
CE_{G_T}	Exponentiation cost in G_T
CE_{Z_p}	Exponentiation cost in Z_p
CP	Cost of pairing operation
$\lvert G\rvert$	Size of an element in G
$\lvert G_r\rvert$	Size of an element in G_T
$\lvert Z_p\rvert$	Size of an element in Z_p

schemes given in [14–17] are more than the proposed scheme. The decryption cost of the scheme given in [17] requires two additional pairing operations and one additional exponentiation in G_T. The decryption cost [14] and [16] is more due to computation of access tree structure. The cost of decryption operation of [15] is also more than the proposed scheme. Hence, the cost of decryption operation of the proposed scheme is also lesser than the cost of decryption algorithms given in [14–17].

5.4 Computational Performance Analysis

This section describes the analysis of the computational performance of the proposed scheme. All computations were done on an Intel Core i5, CPU@2.70GHz machine with 8 GB RAM. The scheme is implemented in Java using jpbc library [21]. Encryption operation has been performed on dummy text files. These files were encrypted using symmetric encryption and secret key is encrypted using [14–17] and the proposed scheme.

Figure 3 shows the comparison of computation cost of encryption operation with varying number of attributes of access policy of schemes given in [14–17] and the presented scheme. The computation cost of the encryption operation of the schemes given in [14–17] is more due to extra pairing operations or computation from access structure on varying number of attributes in access policy. For comparison, the simplified access policy having only AND gates has been designed and complete attribute set u has been included for estimating the computational time of encryption/decryption operations.

Table 5 Comparison of Theoretical Complexities of the schemes given in [14–17] and The Proposed Scheme

Parameter	Component	[14]	[15]	[16]	[17]	The Proposed Scheme
Space complexity	Secret keys of Data Consumer	$(4+2\|N_A\|+\|S_{DC}\|)\|G\|$	$2(\|S_{DC}\|)\|G\|$	$(1+2\|N_{AA}\|+2\|S_{DC}\|)\|G\|$	$2(\|N_{AA}\|+\|S_{DC}\|)\|G\|$	$(\|N_{AA}\|+2)\|G\|$
	Ciphertext	$(5+3\|P\|)+\|G_T\|$	$2(\|P\|)\|G\|+\|G_T\|$	$(2\|N_{AA}\|+2\|P\|)\|G\|+\|P\|\|Z_p\|+\|G_T\|$	$3\|G\|+2\|G_T\|$	$2\|G\|+\|G_T\|$
Time complexity	Encryption operation	$(1+4\|P\|)+CE_{G_T}$	$2(\|P\|)CE_G+CE_{G_T}$	$(2\|N_{AA}\|+2+\|P\|)CE_G+CP+CE_{G_T}+\|P\|CE_{Z_p}$	$(1+\|N_{AA}\|+\|P\|)CE_G+2CE_{G_T}+\|P\|CP$	$(1+\|P\|)CE_G+(\|N_{AA}\|+1)CE_{G_T}$
	Decryption operation	$(\|P\cap S_{DC}\|)CE_{G_T}+(2\|P\cap S_{DC}\|+1)CP$	$2(\|P\|)CE_G+2CE_{G_T}$	$2\|P\|CP+(2\|P\|+1)CE_{G_T}$	$5CP+CE_{G_T}$	3CP

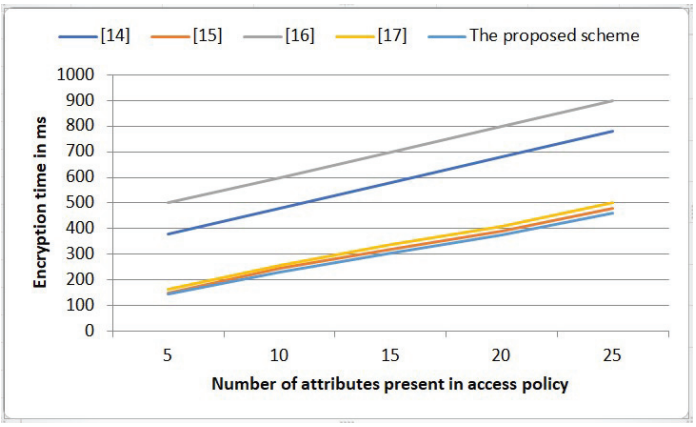

Figure 3 Computational cost of encryption operation of the schemes given in [14–17] and the proposed scheme.

The computational cost of the encryption operation of the proposed scheme and the schemes given in [14–17], if only 5 attributes are specified in the access policy, are 145 ms, 380 ms, 150 ms, 500 ms and 165 ms respectively. For, 15 attributes this cost increases to 305 ms, 580 ms, 320 ms, 700 ms and 337 ms for proposed and schemes in [14–17]. On further enhancing the number of attributes to 25, time taken becomes 460 ms, 780 ms, 480 ms, 900 ms and 500 ms for the given scheme and [14–17] respectively. Thus, the computational cost of the encryption operation of the schemes given in [14–17] are 69%, 4%, 95% and 8% more in comparison with the proposed scheme for 25 attributes.

Hence, it can be inferred that on increasing number of attributes, increase in computation time is lesser for proposed scheme than the schemes presented in [14–17].

Figure 4 shows the comparison of computation cost of decryption operation on varying number of attributes possessed by data consumer according to schemes given in [14–17] and the presented scheme. The computation cost of the decryption operation of the schemes given in [14–17] and the proposed scheme are 180 ms, 110 ms, 200 ms 40 ms and 27 ms respectively for five attributes of data consumer and it is not varying with increase in attributes in proposed scheme. Hence, the computational cost of the decryption operation of the schemes given in [14–17] are 566%, 307%, 640% and 48% more in comparison with the proposed scheme.

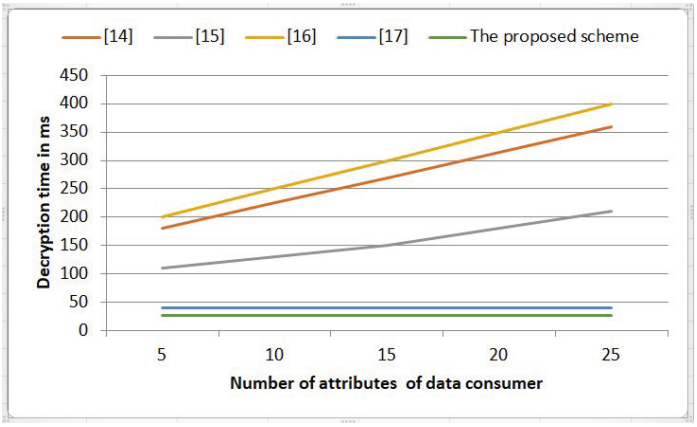

Figure 4 Computational cost of encryption operation of the schemes given in [14-17] and the proposed scheme.

Table 6 Communication cost of transmission of symmetric key and data file using [14–17] and the proposed scheme

Scheme	Communication Cost of Encrypted Symmetric Key (in ms)	Communication Cost of Encrypted Data File (in ms)
[14]	$((5+3\|P\|*\|S\|)\|G\|+\|G_T\|)*t$	$l*t$
[15]	$(2(\|P\|*\|S\|)\|G\|+\|G_T\|)*t$	$l*t$
[16]	$((2\|N_{AA}\|+2\|P\|*\|S\|)\|G\|+\|P\|*\|S\|\|Z_p\|+\|G_T\|)*t$	$l*t$
[17]	$(3\|G\|+2\|G_T\|)*t$	$l*t$
The proposed scheme	$(2\|G\|+\|G_T\|)*t$	$l*t$

$*$Here t = cost required to transmit one bit, l = length of encrypted data file in bits, $|G|$ and $|G_T|$ length of group elements, $|P|$ = Length of access policy in terms of the number of attributes, $|S|$ = number of bits required by an attribute of access policy, $|N_{AA}|$ = Number of AAs involved in generating ciphertext of symmetric key.

5.5 Communication Cost Analysis

In the proposed system model, the data files are encrypted using symmetric encryption schemes. This section analyses the cost of transmitting the encrypted symmetric keys and encrypted data file. The communication cost of sharing an encrypted symmetric key and encrypted data file between DP and CS or CS and DC using schemes [14–17] and the proposed scheme is given in Table 6. Assuming that the transmission cost of 1 bit is 't' and $|S|$ is number of bits required by an attribute of access policy. The symbols used in this analysis are given in Table 4. There is requirement of only one ciphertext

for n number of users. From table 6, it can be seen that the communication cost of the proposed scheme is less as compared to [14–17].

6 Conclusion

This work presents an efficient corrupt resistant key escrow free MA-CP-ABE scheme with constant length ciphertext and hidden access policy. It is secure against dishonest attributes authorities and generates small fixed size secret key also. The results show that the performance of the proposed scheme is better than existing MA-CP-ABE schemes in terms of encryption/decryption cost. The scheme is also proved to be CPA secure. In the proposed scheme, the computational cost of encryption and decryption operations are decreased by atleast 4% and 48% respectively as compared to other existing works. The proposed scheme is secure against corrupt AAs. There is scope to make it more flexible by adding functionality of dynamic attribute assignment to AAs so that the system can be protected from failure in case of down AA. The access policy of the proposed scheme considers only AND gate. The scheme can be extended to consider OR/THRESHOLD based access policies.

References

[1] Sahai A. and Waters B., "Fuzzy identity-based encryption", in Proc. EUROCRYPT, 2005, vol. 3494, pp. 457–473.

[2] Goyal V., Pandey O., Sahai A., Waters B., "Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data", Proceedings of the 13th ACM conference on Computer and communications security, Pages 89-98, Alexandria, Virginia, USA — October 30–November 03, 2006. (cited by 3142)

[3] Bethencourt J., Sahai A., and Waters B., "Ciphertext-Policy Attribute-Based Encryption", 28th IEEE Symposium on Security and Privacy (Oakland), May 2007.

[4] Cheung, Ling, and Calvin Newport. "Provably secure ciphertext policy ABE." Proceedings of the 14th ACM conference on Computer and communications security. ACM, 2007.

[5] Chase M. Multi-authority attribute based encryption. In Theory of cryptography conference 2007 Feb 21 (pp. 515–534). Springer, Berlin, Heidelberg.

[6] Lin H, Cao Z, Liang X, Shao J. Secure threshold multi authority attribute based encryption without a central authority. InInternational Conference on Cryptology in India 2008 Dec 14 (pp. 426–436). Springer, Berlin, Heidelberg.

[7] Nishide, Takashi, Kazuki Yoneyama, and Kazuo Ohta. "Attribute-based encryption with partially hidden encryptor-specified access structures." International conference on applied cryptography and network security. Springer, Berlin, Heidelberg, 2008.

[8] Emura, Keita, et al. "A ciphertext-policy attribute-based encryption scheme with constant ciphertext length." International Conference on Information Security Practice and Experience. Springer, Berlin, Heidelberg, 2009.

[9] Muller S, Katzenbeisser S, Eckert C. On multi-authority ciphertext-policy attribute-based encryption. Bulletin of the Korean Mathematical Society. 2009;46(4):803–19.

[10] Chase M, Chow SS. Improving privacy and security in multi-authority attribute-based encryption. In Proceedings of the 16th ACM conference on Computer and communications security 2009 Nov 9 (pp. 121–130). ACM.

[11] Doshi N, Jinwala D. Constant ciphertext length in multi-authority ciphertext policy attribute based encryption. In 2011 2nd International Conference on Computer and Communication Technology (ICCCT-2011) 2011 Sep 15 (pp. 451–456). IEEE.

[12] Luo E, Liu Q, Wang G. Hierarchical multi-authority and attribute-based encryption friend discovery scheme in mobile social networks. IEEE Communications Letters. 2016 Jun 23;20(9):1772–5.

[13] Zhong H, Zhu W, Xu Y, Cui J. Multi-authority attribute-based encryption access control scheme with policy hidden for cloud storage. Soft Computing. 2018 Jan 1;22(1):243–51.

[14] Vaanchig N, Xiong H, Chen W, Qin Z. Achieving Collaborative Cloud Data Storage by Key-Escrow-Free Multi-Authority CP-ABE Scheme with Dual-Revocation. IJ Network Security. 2018 Jan 1;20(1):95–109.

[15] Ling J, Weng AX. A scheme of hidden-structure attribute-based encryption with multiple authorities. In IOP Conference Series: Materials Science and Engineering 2018 May (Vol. 359, No. 1, p. 012005). IOP Publishing.

[16] Chandrasekaran B, Nogami Y, Balakrishnan R. An Efficient Hierarchical Multi-Authority Attribute Based Encryption Scheme for Profile

Matching using a Fast Ate Pairing in Cloud Environment. Journal of communications software and systems, vol. 14, no. 2, June 2018.

[17] Zhang Y, Li J, Yan H. Constant Size Ciphertext Distributed CP-ABE Scheme with Privacy Protection and Fully Hiding Access Structure. IEEE Access. 2019 Apr 4;7:47982–90.

[18] Challagidad, P.S. and Birje, M.N., 2020. Efficient Multi-authority Access Control using Attribute-based Encryption in Cloud Storage. *Procedia Computer Science*, *167*, pp. 840–849.

[19] Dixit, S., Joshi, K.P. and Choi, S.G., 2019, July. Multi Authority Access Control in a Cloud EHR System with MA-ABE. In 2019 IEEE International Conference on Edge Computing (EDGE) (pp. 107–109). IEEE.

[20] Li, J., Zhang, Y., Ning, J., Huang, X., Poh, G.S. and Wang, D., 2020. Attribute based encryption with privacy protection and accountability for CloudIoT. IEEE Transactions on Cloud Computing.

[21] De Caro A, Iovino V. jPBC: Java pairing based cryptography. In2011 IEEE symposium on computers and communications (ISCC) 2011 Jun 28 (pp. 850–855). IEEE.

Biographies

Shardha Porwal is Assistant Professor in department of computer science engineering and information technology in Jaypee Institute of Information Technology, Noida. Her research interest includes Cryptography and Network Security, Data structure and algorithm. She has published several papers in international conferences and journals. She received her M.Tech. degree from Maulana Azad National Institute of Technology, Bhopal, India. She has 10+ years of teaching experience.

Sangeeta Mittal is Associate Professor in Jaypee Institute of Information Technology, Noida. Her areas of research interest include Software Defined Networks, Computer Networks, Network Security, Cryptography, Sensor Based Smart Environments and Wireless Sensor Networks. She has published several papers in international conferences and journals of repute. She earned her PhD from Jaypee Institute of Information Technology, M.E. in Computer Engineering from Punjab University Chandigarh India and BE from Maharishi Dayanand University, Rohtak, India. She has 16+ years of experience in teaching UG and PG computer science courses. She is a member of ACM and life member of CSI.