International Journal of Safety and Security Engineering
Vol. 13, No. 3, June, 2023, pp. 527-538
Journal homepage: http://iieta.org/journals/ijsse

A Proxy-Based and Collusion Resistant Multi-Authority Revocable CPABE Framework with Efficient User and Attribute-Level Revocation (PCMR-CPABE)

Shobha Chawla*, Neha Gupta

School of Computer Applications, Manav Rachna International Institute of Research and Studies, Faridabad 121001, India

Corresponding Author Email: shobha.chawla@gmail.com

https://doi.org/10.18280/ijsse.130315

Received: 11 May 2023
Accepted: 26 June 2023

Keywords:
multi-authority, CPABE, decentralized, user revocation, attribute-level revocation, LSSS, collusion resistant, forward and backward secrecy

ABSTRACT

The presence of multiple authorities in multi-authority ciphertext policy attribute based encryption (CPABE) schemes hinders an adversary's ability to compromise security. As each authority is responsible to provide secret keys to the users, thus enforcement of fine-grained access control should be carefully designed to ensure data confidentiality. The current study critically reviews the methodologies employed to address user-level and attribute-level revocation in the existing studies. The study has focused on the revocation methodology of those CPABE schemes that are implemented using bilinear pairing cryptography for the encryption and Linear Secret Sharing Scheme (LSSS) for the access structure. It has been observed that the approaches implemented in the existing schemes are computationally expensive and are vulnerable to collusion attacks caused by the cloud and revoked users. Thus, an efficient proxy-based and collusion resistant multi-authority revocable CPABE framework (PCMR-CPABE) is proposed in the current study. The proposed framework is decentralized, dynamic, scalable, and ensures forward/backward secrecy. Additionally, the proposed framework is computationally efficient and is practical to implement as it does not require secret key or group secret key and ciphertext update to address revocation. Furthermore, the incorporation of time and identity-based components allows the proposed framework to resist collusion attacks efficiently.

1. INTRODUCTION

Since cloud computing has become the "buzzword" in the information technology industry, researchers are looking into and identifying a number of solutions to the security challenges encountered by cloud users. The key security concerns of cloud users included enforcement of data confidentiality and fine-grained access control. Goyal et al. [1] have introduced attribute-based encryption (ABE) scheme to address and restrict unauthorized access to the cloud resident sensitive data and realized one-to-many encryption in the scheme. Ciphertext policy attribute based encryption (CPABE) is a class of ABE and has been introduced by Brethencourt et al. [2]. CPABE allowed one-to-many encryptions along with the enforcement of fine-grained access control. CPABE recommended that a data owner shall formulate an access policy and embed it with the encrypted sensitive files before outsourcing to the cloud. Additionally, the scheme states that any data user who wants to access the encrypted file should hold certain attributes such as name, pan card number, driving license, etc. These possessed attributes are used by a data user to obtain a secret key from the attribute authority. The attribute authority is the entity that manages attributes and distributes secret keys to the users. The obtained secret key is used by the data user to decrypt the ciphertext. Furthermore, the secret key of the data user should satisfy the defined access policy to successfully decrypt the ciphertext. For example, (Branch=CS AND (Profile=Faculty OR (Profile=Student AND Batch=2023))) is an access policy defined by the data owner of encrypted file. The access policy says a data user who is faculty of CS Branch or a student studying in CS Branch in 2023 can only satisfy the access policy. Only the data users whose secret key possess sufficient attributes could successfully decrypt the ciphertext. Existing studies have implemented CPABE either using single-authority or multi-authority systems.

In the single-authority CPABE scheme, the responsibility of generating and distributing the secret key to data users has been delegated to a single authority. Such an approach turns into an impractical approach if the attributes possessed by a data user are issued by distinct authorities. In the real world, attributes included in the access policy are issued by several authorities. For example, the income tax authority issues pan card, driving licences are issued by transport authorities, and so on. In such cases, the attributes possessed by a data user can be verified only by the issuing authorities. The multi-authority CPABE concept was therefore designed to satisfy the aforementioned need. Additionally, the threat of data loss escalates with the single authority. It is because the entire system could get affected if it were compromised by any attacker. The presence of multiple attribute authorities in the multi-authority CPABE scheme makes it challenging for the attackers to breach security.

The existing studies have proposed majorly two implementations in the different versions of the multi-authority CPABE scheme viz: centralized multi-authority CPABE scheme and decentralized multi-authority CPABE scheme [3, 4]. The former says that the central authority controls the distribution of the secret key of the user apart from handling users' and attributes authorities' registration. Whilst

the latter implementation provides distributed control and the attribute authorities independently manage the attributes held by the data user as depicted in Figure 1.

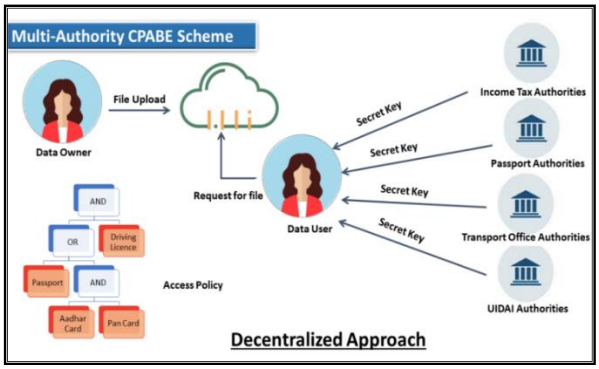

Figure 1. Decentralized approach

Figure 1 exhibits the communication and data flow between various entities involved in the multi-authority CPABE scheme employing distributed control. The encrypted file uploaded by a data owner has a defined access policy. A user whose secret key satisfies the access policy can only decrypt the file successfully. For example, the access policy in Figure 1 states that the encrypted file shall be accessible to the user only if the data user possesses an aadhar card and pan card along with a driving licence or if he only possesses a driving licence and passport. It is demonstrated in Figure 1 that all the attributes are issued by distinct authorities and are independently controlled.

The researchers found numerous challenges with the base CPABE approach, including policing hiding, traceability, single-authority, revocability, etc. In this paper, the revocability challenge within the multi-authority CPABE scheme has been studied. Data users of the system in an organization leave or are denied access if traced as malicious users. It has remained a major challenge for researchers to provide an effective method for dynamically revoking such users with an effective computing capability. Apart from system-level revocation of a user, an efficient solution to attribute-level revocation has also remained a challenge amongst researchers. This paper contributes the solution to the revocation issue in multi-authority CPABE framework. The proposed framework of revocable decentralized multi-authority CPABE framework based on bilinear pairing cryptography contributes the following properties:

1. User and Attribute-level Revocation – the proposed PCMR-CPABE framework provide solutions to the user as well as attribute-level revocation. The proposed framework has employed a trusted proxy server to enforce fine-grained access control. The secret decryption key of a data user has two parts: secret key and proxy key. The secret keys are issued by the associated attribute authority to the data user and the proxy key is issued by the proxy server. The proxy key is time bound and is invalidated by the proxy server whenever the access privileges of the data user changes. The design of the framework allows dynamic revocation of access rights of a user if found malicious or if he exits the system. Many a time, users lose certain attributes however, are still a member of the system. Thus, the design of the framework dynamically allows the revocation of lost attributes and the data user shall be allowed to access only those files which are accessible with the remaining attributes.

2. Collusion Resistant – The proposed PCMR-CPABE framework's design makes it harder for the revoked users to collude in an attack as well as for the cloud service provider to collude with the revoked users. The research that currently exists paid little attention to collusion attacks carried on by dishonest cloud service providers. The proposed PCMR-CPABE framework gave minimal privileges to cloud service provider. The cloud service provider has been delegated no role in key distribution and decryption unlike existing schemes. Thus, the provider has no means to access or store the secret key or proxy key of data user. The proposed framework contributes an efficient collusion-resistant design of a multi-authority CPABE scheme.

3. Computationally Efficient – the proposed PCMR-CPABE framework is computationally efficient as it does not require an update of non-revoked users' secret keys or ciphertext updates to enforce revocation. The design of the proposed framework employed a proxy server to control unauthorized access. Therefore, it has improved computational efficiency in comparison to the existing literature.

4. Dynamic and Scalable – as users in an organization leave and join frequently, thus their access privileges should be immediately updated to avoid unauthorized access. Similarly, the job roles of users keep changing in an organization and so as their access privileges. Additionally, these changes ought to be made immediately to prevent unauthorized access. The design of the proposed PCMR-CPABE framework ensures instant and scalable revocation as it only requires updating of the proxy key of revoked user to deny access.

The rest of the paper is organized as follows: the second section critically reviews the methodology adopted by the existing revocable multi-authority CPABE schemes, the third section discusses the mathematical background required for the proposed implementation, the fourth section proposes an efficient framework of revocable multi-authority CPABE scheme based on bilinear pairing cryptography, the fifth section assesses the strength of proposed framework against security attacks, the sixth section discusses and compares the performance of proposed framework with the existing schemes, the seventh section presents the implementation results. The ninth section serves as the conclusion of the study.

2. LITERATURE SURVEY

An efficient CPABE scheme should address revocation effectively. In this section, the methodologies adopted to employ revocation by the existing multi-authority CPABE schemes, based on bilinear pairing cryptography with LSSS, have been critically reviewed to identify the research challenges. Studied schemes included either the centralized or decentralized approach. CPABE scheme enforces authorized access by allowing a data user to decrypt the ciphertext only if his secret key, based on his attributes, could satisfy the access policy. Apart from this, the scheme should also revoke unauthorized access privileges instantaneously. Moreover, the access privileges should be controlled both at the system level and attribute level. Existing schemes have provided the solution to either user revocation or attribute-level revocation. Very few studies have addressed both the level of access rights revocation.

Table 1. Comparitive review of studied literature

S.No.	Scheme	Contribution	Research Gaps
1.	[5]	1. User Revocation 2. Dynamic 3. Forward and Backward Secrecy	1. Lack of Attribute-Level Revocation 2. Key Update of Non-Revoked Users or Attribute Group Key 3. Ciphertext Update or Re-encryption 4. Partial Collusion Resistant
2.	[6]	1. User Revocation 2. Dynamic 3. Fully Collusion Resistant 4. Forward and Backward Secrecy	1. Lack of Attribute-Level Revocation 2. Key Update of Non-Revoked Users or Attribute Group Key
3.	[7]	1. User Revocation 2. Dynamic 3. Forward and Backward Secrecy	1. Lack of Attribute-Level Revocation 2. Key Update of Non-Revoked Users or Attribute Group Key 3. Ciphertext Update or Re-encryption 4. Partial Collusion Resistant
4.	[8]	1. User Revocation 2. Dynamic 3. Forward and Backward Secrecy	1. Lack of Attribute-Level Revocation 2. Key Update of Non-Revoked Users or Attribute Group Key 3. Ciphertext Update or Re-encryption 4. Partial Collusion Resistant
5.	[9]	1. User Revocation 2. Dynamic 3. Forward and Backward Secrecy	1. Lack of Attribute-Level Revocation 2. Key Update of Non-Revoked Users or Attribute Group Key 3. Ciphertext Update or Re-encryption 4. Partial Collusion Resistant
6.	[10]	1. User Revocation 2. Dynamic 3. Fully Collusion Resistant 4. Forward and Backward Secrecy	1. Lack of Attribute-Level Revocation 2. Key Update of Non-Revoked Users or Attribute Group Key
7.	[11]	1. User Revocation 2. Dynamic 3. Forward and Backward Secrecy	1. Lack of Attribute-Level Revocation 2. Key Update of Non-Revoked Users or Attribute Group Key 3. Ciphertext Update or Re-encryption 4. Partial Collusion Resistant
8.	[12]	1. User Revocation 2. Dynamic 3. Forward and Backward Secrecy	1. Lack of Attribute-Level Revocation 2. Key Update of Non-Revoked Users or Attribute Group Key 3. Partial Collusion Resistant
9.	[13]	1. Attribute -Level Revocation 2. Dynamic 3. Forward and Backward Secrecy	1. Lack of User Revocation 2. Key Update of Non-Revoked Users or Attribute Group Key 3. Ciphertext Update or Re-encryption 4. Partial Collusion Resistant
10.	[14]	1. Attribute-Level Revocation 2. Fully Collusion Resistant 3. Forward and Backward Secrecy	1. Lack of User Revocation 2. Key Update of Non-Revoked Users or Attribute Group Key 3. Lack of dynamicity
11.	[15]	1. Attribute-Level Revocation 2. Dynamic 3. Forward and Backward Secrecy	1. Lack of User Revocation 2. Key Update of Non-Revoked Users or Attribute Group Key 3. Ciphertext Update or Re-encryption 4. Partial Collusion Resistant
12.	[16]	1. Attribute-Level Revocation 2. Dynamic 3. Forward and Backward Secrecy	1. Lack of User Revocation 2. Key Update of Non-Revoked Users or Attribute Group Key 3. Ciphertext Update or Re-encryption 4. Partial Collusion Resistant
13.	[17]	1. Attribute-Level Revocation 2. Dynamic 3. Forward and Backward Secrecy	1. Lack of User Revocation 2. Key Update of Non-Revoked Users or Attribute Group Key 3. Ciphertext Update or Re-encryption 4. Partial Collusion Resistant
14.	[18]	1. Attribute-Level Revocation 2. Dynamic 3. Forward and Backward Secrecy	1. Lack of User Revocation 2. Key Update of Non-Revoked Users or Attribute Group Key 3. Ciphertext Update or Re-encryption 4. Partial Collusion Resistant
15.	[19]	1. Attribute-Level Revocation 2. Dynamic 3. Forward and Backward Secrecy	1. Lack of User Revocation 2. Key Update of Non-Revoked Users or Attribute Group Key 3. Ciphertext Update or Re-encryption 4. Partial Collusion Resistant
16.	[20]	1. User and Attribute-Level Revocation 2. Dynamic 3. Forward and Backward Secrecy	1. Key Update of Non-Revoked Users or Attribute Group Key 2. Ciphertext Update or Re-encryption 3. Partial Collusion Resistant
17.	[21]	1. User and Attribute-Level Revocation 2. Dynamic 3. Forward and Backward Secrecy	1. Key Update of Non-Revoked Users or Attribute Group Key 2. Ciphertext Update or Re-encryption 3. Partial Collusion Resistant
18.	[22]	1. User and Attribute-Level Revocation 2. Forward and Backward Secrecy	1. Key Update of Non-Revoked Users or Attribute Group Key 2. Ciphertext Update or Re-encryption 3. Partial Collusion Resistant 4. Lack of dynamicity

S.No.	Scheme	Contribution	Research Gaps
19.	[23]	1. User and Attribute-Level Revocation 2. Dynamic 3. Forward and Backward Secrecy	1. Key Update of Non-Revoked Users or Attribute Group Key 2. Ciphertext Update or Re-encryption 3. Partial Collusion Resistant

2.1 User revocation

Whenever a user is identified as malicious or is leaving the organization, his access privileges should be instantly revoked. It has been identified that the existing multi-authority CPABE schemes mainly advocated updating of non-revoked users' key and ciphertext update or re-encryption as the solution to enforce access control on revoked users from accessing the ciphertext. Employing this approach on each user's revocation increases computational cost. Therefore, existing schemes outsourced ciphertext re-encryption to the proxy server or cloud service provider to enforce revocation and reduce computational cost. Additionally, the studied literature assumed cloud service providers as semi-trusted servers and thus, the possibility of collusion between the cloud service providers and the revoked users have received the least attention [5-12].

2.2 Attribute-level revocation

Many times a user is not revoked, however, his role has changed. Such changes cause changes in the attributes held by the user. For example, an employee shifted from the accounts department to the sales department. In such cases, it is the responsibility of the scheme to deny access to data users on encrypted files that were earlier accessible by the revoked attributes unless the remaining attributes still satisfy the access policy. Revoking rights at the attribute-level helps to update the access permissions of the data users. An update of the attribute group key has been employed by the studied schemes to address attribute-level revocation. Furthermore, the ciphertext is re-encrypted to control unauthorized access. It has been observed that if a user loses any attribute, the existing schemes approach lead to updates of ciphertext and subsequent update of non-revoked users' key to achieve forward and backward secrecy. Consequently, instantaneous revocation caused increased computational overhead [13-20].

The study of existing multi-authority CPABE schemes based on bilinear pairing cryptography exhibits that very few schemes have addressed both levels of revocation however, the approaches implemented to restrict the access privileges of the revoked users have not considered the potentiality of the cloud service provider to collude with the revoked users [21-23]. Huang [23] too updated users' key and re-encrypted ciphertext to enforce user and attribute-level revocation. All attributes held by the user have been revoked by the involved attribute authorities in the proposed approach to address user revocation.

The approach used to revoke users [13, 22] has been implemented using temporal-based access control. Although, the temporal-based approach has not updated the ciphertext, however, the schemes lacked dynamicity.

Table 1 exhibits the comparative study on contribution and research gaps identified in the existing studies. It has been identified in the existing studies, instantaneous revocation of access rights caused increased computational overhead and very little attention has been given to the possibility of the cloud turning malicious and its capability to collude with the revoked users. The existing techniques are also computationally expensive because they update non-revoked users' keys or attribute group keys and re-encrypt ciphertext with each addressed revocation. An efficient revocable multi-authority CPABE scheme using a centralized or decentralized approach based on bilinear pairing cryptography must include the following properties:

1. User and Attribute-Level Revocation: A scheme addressing revocation should allow for instantaneous and scalable revocation of access rights both at the system and attribute level and should ensure forward and backward secrecy. Forward secrecy restricts revoked users to access newly uploaded files with the secret keys which they lately possessed. Whilst backward secrecy restricts revoked users to access old files which were earlier accessible with the held secret keys.
2. Collusion Resistance: Collusion is possible between revoked users or between cloud and revoked users or between non-revoked users and revoked users. An efficient scheme should prevent the possibility of every type of collusion.
3. Computational Overhead: Increased computation cost reduces the efficiency of any scheme. Computational cost increases exponentially as a result of updating non-revoked users' keys and ciphertext re-encryption with each revocation to prevent revoked users from accessing.

The proposed PCMR-CPABE framework aims to eliminate all the research gaps and design an efficient multi-authority revocable CPABE framework.

3. MATHEMATICAL BACKGROUND

This section discusses the basic definitions of bilinear pairing cryptography and access structure, which serves as the foundation for the suggested framework.

3.1 Bilinear pairing

Definition 1. Suppose G_1 , G_2 and G_T are three multiplicative cyclic groups of prime order p, g_1 and g_2 be generator of G_1 and G_2 respectively, and e is a bilinear map such that e: $G_1 \times G_2 \rightarrow G_T$ [24]. The bilinear map e should satisfy the following properties:

- Bilinearity: $\forall\, u \in G_1, \forall\, v \in G_2$ and $a, b \in Z_p$ there is a bilinear map e such that $e(u^a, v^b) = e(u, v)^{ab}$
- Non-degeneracy: $e(g, g) \neq 1$

3.2 Access structure

Definition 2. Suppose $P = \{P_1, P_2, P_3 \dots \dots \dots P_n\}$ be a set. A collection $\mathbb{A} \subseteq 2^{\{P_1, P_2, P_3 \dots\dots\dots P_n\}}$ of non- empty subsets of $\{P_1, P_2, P_3 \dots \dots \dots P_n\}$ is called as monotone access structure if for any C and D: if $C \in \mathbb{A}$ and $C \subseteq D$, then $D \in \mathbb{A}$.The sets belonging to $\mathbb{A}$ are termed as authorized sets, and the sets which are not belonging to $\mathbb{A}$ are termed as unauthorized sets [25].

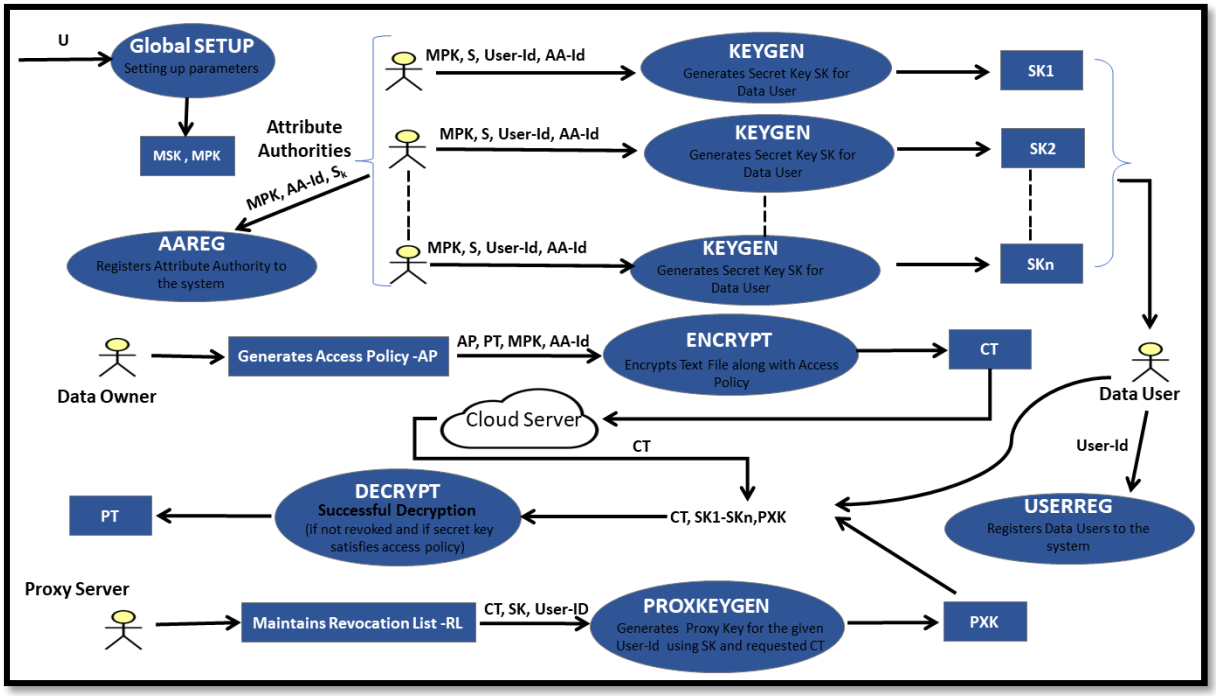

Figure 2. PCMR-CPABE framework

4. PCMR-CPABE FRAMEWORK

This section proposes an efficient revocable proxy-based de-centralized multi-authority CPABE framework, constructed using bilinear pairing cryptography. The proposed PCMR-CPABE framework consists of five entities, each of which is responsible for carrying out one of the seven modules, as shown in the Figure 2. Each entity has its designated responsibility.

As depicted in Figure 2, the entities participating in the proposed framework are:

1. Attribute Authorities: Attribute Authorities are the entities that issues secret key to data user after verifying the possessed attributes. In the multi-authority construction, each authority owns its universe of attributes. Moreover, the attributes possessed by the data user belong to distinct authorities. Thus, the data user acquires secret keys from various attribute authorities of systems.
2. Data Owner: Sensitive files are encrypted before outsourcing to the cloud service provider and made publicly available to data users. The entity that owns the sensitive files and who also determines the access policy of the file is termed the data owner.
3. Cloud Service Provider: Cloud service provider is a third party who offers cloud-based platforms to the sensitive data of data owners.
4. Data User: The data user is the end user who requests access permissions for the sensitive files stored at cloud-based platforms. A data user is assigned a secret decryption key based on the attributes held by them. The secret decryption key has two parts- the secret key issued by the attribute authorities and the proxy key issued by the proxy server. The data user's secret decryption key could successfully decrypt the ciphertext if the issued secret decryption key satisfies the access policy associated with the ciphertext.
5. Proxy Server: The proxy server handles revocation both at the user and attributes level. The proxy key regulates access control and causes unsuccessful decryption if the user loses access rights to the requested ciphertext.

The following modules are run by the entities in the proposed PCMR-CPABE framework:

Step 1:

$$\text{Global\_Setup}() \rightarrow (\text{MPK}, \text{MSK})$$

This module initializes the base setup parameters of the framework. The Global_Setup module generates the master public key MPK and master secret key MSK. The module chooses a bilinear cyclic group G of prime order p where g is generator of G and $\alpha, v \, \epsilon Z_p$ are randomly chosen elements.

$$MPK = \{g, p, q = g^{\alpha}, e(g,g)^{\alpha}, Z = g^{v}\}, MSK = \{\alpha, v\} \quad (1)$$

Step 2:

$$\text{UserReg}(\text{ID}_u) \rightarrow PK_u$$

Let S_u denotes set of users in the system. UserReg module registers the new joining user u to the system. It reads identity of user ID_u and generates the private key PK_u for the user. The generated private key is stored at system level and also shared to user u, where $u \in S_u$. In this module three random components $(q_u, a_u, b_u)\epsilon \, Z_p$ are generated to compute private key components. The private key of the user is unique and used as identity component of user in other modules.

$$PK_u = ((q_u, a_u, b_u)\epsilon \, Z_p, U0 = q_u(a_u + b_u), U1 = q_u.a_u.b_u, U2 = q_u.a_u) \quad (2)$$

The generated PK_u is shared with the data user through secure channel.

Step 3:

$$\mathrm{AAReg}\left(\mathrm{ID_k}, \mathrm{MPK}, S_{A_k}\right) \rightarrow (\mathrm{PK_k}, \mathrm{SK_k})$$

Let S_A and S_{A_k} denotes set of authorities and set of attributes managed by the attribute authority AA_k (where $k \in S_A$) in the system respectively. AAReg module registers the new joining attribute authority AA_k to the system and generates a public key PK_k and secret key SK_k, where $k \in S_A$, for the attribute authority. The module requires identity of attribute authority ID_k, S_{A_k} and MPK to generate PK_k and SK_k. Furthermore, $\{t_1, t_2, \ldots\ldots\ldots\ldots, t_{A_k}\} \in Z_p$ are randomly chosen values for the attributes controlled by the authorities AA_k. The PK_k and SK_k are distributed to the attribute authority through secure channel. As decpicted in Figure 2, it is assumed in the framework that each authority manages disjoint set of attributes.

$$SK_k = (\delta_\mathrm{k}, \beta_\mathrm{k}, \gamma_\mathrm{k}) \epsilon\, Z_p\,,\ PK_k = (\,AA1_k = Z^{\delta_\mathrm{k} \cdot \beta_\mathrm{k}}\,,\ AA2_k = g^{\delta_\mathrm{k} \cdot \beta_\mathrm{k}},\ Y_i = g^{t_i}\ where\ i \epsilon S_{A_k}) \quad (3)$$

Step 4:

$$\mathrm{Encrypt}\left(\,\mathrm{MPK}, \mathrm{PK_k}, \mathcal{M}, (\mathrm{W}, \rho)\right) \rightarrow \mathrm{CT}$$

The encrypt module is run by the data owner to encrypt message $\mathcal{M}$ before uploading to the cloud. Let I_A denotes the attribute authorities who control the attributes that are comprised in the access policy. The encrypt module processes a master public key MPK, set of public key PK_k of involved attribute authorities, where $k \in I_A$ and linear secret sharing scheme (LSSS) based access structure along with the Message $\mathcal{M}$. The access structure $\mathbb{A} = (W, \rho)$ with m rows and n cols helps to generate the ciphertext. The algorithm of Linear Secret Sharing Scheme (LSSS) matrix is widely used to express monotone access structure. According to the algorithm, for realizing an access structure $\mathbb{A}$ and to be considered as linear over Z_p a secret sharing scheme Π for a set of parties P should satisfy the following properties [26]:

- The shares of secret s of each party or attribute from the set P represent a vector whose base is a finite field Z_p.
- In accordance with LSSS, the module generates a vector $\vec{v} = (s, r_1, r_2, \ldots\ldots\ldots\ldots, r_n) \epsilon Z_p$ where s is a shared secret and $(\,r_1, r_2, \ldots\ldots\ldots\ldots, r_n) \in Z_p$ are randomly generated numbers. For $i = 1\ to\ m$, the module computes a vector $\lambda_i = W_i.\vec{v}$ that holds m shares of secret and each share belongs to the attribute i_k [25].

$$CT = C = \mathcal{M}.e(g,g)^s\,,\ C0 = g^s,\ C1_{i_k} = g^{\lambda_{i_k}}, C2_{i_k} = g^{r_{i_k}},\ C3_{i_k} = g^{\lambda_{i_k}}.Y_{i_k}^{-r_{i_k}},\ C4_{i_k} = Z^{-r_{i_k}} \quad (4)$$

Step 5:

$$\mathrm{KeyGen}(\mathrm{ID_u}, \mathrm{S_x}, PK_u, \mathrm{PK}_k, MPK) \rightarrow (\mathrm{SK}_{u_k})$$

The data user needs a secret decryption key to decrypt ciphertext. The secret decryption key has two parts – the secret key and the proxy key. The module for key generation is invoked by the data user to acquire the secret key. This module reads master public key MPK, unique identification ID_u of the user u, sets of attributes S_x owned by user u , private key PK_u of user u, and public key PK_k of authority managing attribute x, where $x \in S_x$ and $k \in S_A$.

The generated secret keys from all the involved attribute authorities are distributed to the data user through secure channel. All the received secret keys along with proxy key are submitted to decrypt module to acquire plain text.

$$SK_{u_k} = (K0_k = (g^{U0}.g^{U1}), K1_k = g^{q_u}, K2_{x_k} = Y_{x_k}{}^{U2}.AA1_k, K3_k = g^{U2}, K4_k = AA2_k) \quad (5)$$

Step 6:

$$\mathrm{ProxKeyGen}\left(\mathrm{CT}, \mathrm{PK_u}, \mathrm{SK}_{u_k}\right) \rightarrow \mathrm{PXK_u}$$

To successfully decrypt the ciphertext, a data user also needs a second part of the secret decryption key. The proxy server invokes the ProxKeyGen module and communicates the proxy key to the data user. The input parameters of this module are private key PK_u of user u, secret keys SK_{u_k} of user u and the requested ciphertext by the data user. The proxy key expires after a certain time-period to prevent collusion between the cloud service provider and the revoked users. A proxy key issued to a user is found to be timed out after its expiry. As a result, even if the revoked user attempts decryption with the proxy key issued prior to revocation, the decryption process fails due to the expired proxy key. Subsequently, the newly requested proxy key by the revoked users denies them access to the ciphertext and imposes fine-grained access control. For imposing time out on the proxy key, the module calculates a time component t_e for the proxy key.

For the user u,

$$PXK_u = (For\ each\ attribute\ i\ of\ ciphertext, C\_1_i = C1_i{}^{(b_u)}, For\ x\ attributes\ of\ user\ u, managed\ by\ authority\ k)$$

$$K\_2_{x_k} = (K2_{x_k}, T_s = C0^{t_e}) \quad (6)$$

The proxy key enables enforcement of fine-grained access control. Proxy key controls revocation both at system-level and attribute level as explained below:

User Revocation - For revoking a malicious user, the proxy server invalidates the value of b_u. Thus, the received proxy key leads to the failure of the decryption process.

Attribute-level Revocation – For revoking a user at attribute level, the revoked attributes are negated by the proxy server. Consequently, the files that were earlier accessible with the revoked attributes of the user will be accessible only if the rest of the attributes satisfy the access policy else the decryption process fails.

For non-revoked attributes l of user u,

$$PXK_u = (For\ each\ attribute\ i\ of\ ciphertext, C_{1_i} = C1_i{}^{(b_u)}, For\ non-revoked\ attributes)$$

For non-revoked attributes l managed by attribute authority k,

$$K\_2_{l_k} = (K2_{l_k}, T_s = C0^{t_e}) \quad (7)$$

Step 7:

$$\text{Decrypt}(\text{CT}, \text{SK}_\text{u}, \text{PXK}_u) \rightarrow \mathcal{M}$$

The decrypt module reads a secret key SK_u and proxy key PXK_u of the user to decrypt the ciphertext CT to message $\mathcal{M}$. In the first place, the module calculates the current time component t_c valid time-period V for the proxy key to examine the validity of proxy key. Subsequently, if the proxy key has been proven valid and the secret decryption key satisfies the access policy, the ciphertext gets successfully decrypted. The proxy key causes unsuccessful decryption for the revoked user. Let $I = \{I_{A_k}\}_{k \in I_A}$ represents a set of all attributes included in ciphertext from different attribute authorities k. Furthermore, for successful decryption, the module also incorporates the linear reconstruction property and calculates $w_i \epsilon Z_p$ where $i \in I$. Consequently $s = \sum_{i \in I} w_i . \lambda_i$ is reconstructed to determine valid shares λ_i. The decryption module computes $\mathcal{M}$ as follows:

$$\left\{ if\ |C0^{t_c} - T_s| < V \middle| \mathcal{M} = \frac{C}{B}, otherwise\ \mathcal{M} = \perp \right\}$$

where $C = \mathcal{M}.e(g,g)^{\alpha s}$ and

$$B = \frac{\prod_{k \in K} e(C0, K0_k)}{\prod_{i_k \in I_{A_k}} (e(C\_1_i, K1_k).e(C2_i, K\_2_{i_k}).e(K3_k, C3_i).e(K4_k, C4_i))^{w_i}} \quad (8)$$

$$B = e(g,g)^{U1.s} \\ M = C/(B)^{-U1} \quad (9)$$

The proposed framework efficiently decrypts the ciphertext and the proxy server addresses revocation both at the system and attributes level. The construction of 7 modules makes the framework dynamic, collusion resistant, and computationally efficient. The strength of the framework in terms of security is discussed in the following section.

5. SECURITY ANALYSIS

Security analysis aims to overview and assess the security threats against the proposed framework. In this section, security against data confidentiality and collusion attack has been proven. In addition, the realization of forward and backward secrecy in the framework has been assessed in the following propositions:

5.1 Proposition 1

If the decisional q-parallel BDHE assumption is true, then the adversaries' algorithms that run in polynomial time and have LSSS share matrix of $m^* \times n^*$ where $m^*, n^* \leq q$ as a challenge, will have a negligible advantage when trying to selectively compromise the security of the proposed framework.

Provability:

Init: The challenger *C* accepts q-parallel BDHE challenge $\vec{y}, T$ and receives the challenge access matrix (M^*, ρ^*) by adversary *A*, where M^* is a matrix of *m* rows and *n* cols with $m^*, n^* \leq q$ and ρ^* functions as mapping function. Here, $\vec{y} = \left(g, g^s, g^a, \ldots\ldots, g^{(a^q)}, g^{a^{q+1}}, \ldots\ldots\ldots\ldots g^{a^{2q}}\right)$ and if the message stays secret from the adversary then a random element will be generated through group G_T and assigned to *T*, otherwise $T = e(g,g)^{a^{q+1}v}$.

Setup: The challenger in the first place chooses a random parameter $v \in Z_p$ and sets $s = va^{q+1}$ and then runs two modules Global_Setup() and AAReg() and shares g with the adversary. The adversary then chooses a set of compromised authorities S_A' and shares with the challenger. The challenger then chooses three random parameters $(\delta_\text{k}, \beta_\text{k}, \gamma_\text{k}) \epsilon Z_p$ for each uncompromised authorities $AA_k, where\ k \in S_A - S_A'$.

Furthermore, the challenger computes a random oracle y_{x_k} and chooses random parameter $z_{x_k} \epsilon Z_p$ for each attribute x in the system of k attribute authority and *I* represents the set of indices such that

$$y_{x_k} = g^{x_k} \prod_{i \in I} g^{a.\mathcal{M}_{i,1}/b_i} . g^{a^2 \mathcal{M}_{i,2}/b_i} \ldots\ldots\ldots\ldots g^{a^n \mathcal{M}_{i,n}/b_i}$$

If $I = 0$, such that there does not exist any i for $\rho^*(i) = x$, then $y_{x_k} = g^{z_x}$.

Thereafter, the challenger computes the public key PK_k using the above mentioned $SK_k = (\delta_\text{k}, \beta_\text{k}, \gamma_\text{k}) \epsilon Z_p$ for all uncorrupted authorities such that

$$PK_k = (\ AA1_k = Z^{\delta_\text{k} \cdot \beta_\text{k}}\ ,\ AA2_k = g^{\delta_\text{k} \cdot \beta_\text{k}}, \\ Y_i = g^{t_i}\ where\ i \epsilon S_{A_k})$$

Moreover, adversary is distributed a unique user ID PK_u by the challenger such that,

$$PK_u = ((g_u, a_u, b_u) \epsilon\ Z_p, U0 = g_u(a_u + b_u), U1 \\ = g_u.a_u.b_u, U2 = g_u.a_u)$$

where, $(g_u, a_u, b_u) \epsilon\ Z_p$ are randomly chosen parameters.

Phase1: In the phase 1, the challenger responds to multiple queries issued by the adversary on secret key and the proxy key where each query has two inputs $ID_u\ and\ S_k$, user id and set of attributes respectively. It is assumed that S_k belongs to uncorrupted authorities. Subsequently, the revocation list $R_1^* = \{u_0^*, u_1^*, \ldots\ldots u_k^*\}$ is shared to challenger and the challenger updates its list R^*. The challenger returns null value on the condition that the S_k shared by the adversary satisfies the access matrix (M^*, ρ^*) however, $\{u_j^*\} \notin R^*$. Otherwise, following conditions holds true:

- If S_k of adversary *A* does not satisfies (M^*, ρ^*) and the user $\{u_j^*\} \notin R^*$, then a vector $\vec{r} = (r_1, r_2, \ldots.. r_{n^*}) \in Z_p$ is generated where $w_1 = -1\ and\ \forall\ i,\ \rho^*(i) \in S_k'$ and $M^*.\vec{r} = 0$. Furthermore, the challenger randomly computes $x, y, z \in Z_p$ and defines q_u, a_u, b_u as follows:

$$q_u = t1_u = x' + w_1.a^q \\ + w_2.a^{q-1} \ldots\ldots + w_n.a^{q-n^*+1} \quad (10)$$

$$a_u = t2_u = y' + w_1.a^q \\ + w_2.a^{q-1} \ldots\ldots + w_n.a^{q-n^*+1} \quad (11)$$

$$b_u = t3_u = z' + w_1.a^q \\ + w_2.a^{q-1} \ldots\ldots + w_n.a^{q-n^*+1} \quad (12)$$

and further computes

$$K0_k = (g^{U0}.g^{U1}) = (g^{g_u(a_u+b_u)}.g^{g_u.a_u.b_u}) = \\ (g^{t1_u(t2_u+t3_u)}.g^{t1_u.t2_u.t3_u}) = \\ (g^{x'.\prod_{i=1,n^*}(g^{a^{q-i+1}})^{w_i}.(y'.\prod_{i=1,n^*}(g^{a^{q-i+1}})^{w_i} + z'.\prod_{i=1,n^*}(g^{a^{q-i+1}})^{w_i}}). \quad (13)$$

$$g^{x'.\prod_{i=1,n^*}(g^{a^{q-i+1}})^{w_i}.y'.\prod_{i=1,n^*}(g^{a^{q-i+1}})^{w_i}.z'.\prod_{i=1,n^*}(g^{a^{q-i+1}})^{w_i}}$$

$$K1_k = g^{g_u} = g^{t1_u} = g^{x'.\prod_{i=1,n^*}(g^{a^{q-i+1}})^{w_i}} \quad (14)$$

$$K2_{x_k} = Y_{x_k}{}^{U2}.AA1_k = Y_{x_k}{}^{g_u.a_u}.Z^{\delta_k.\beta_k} = Y_{x_k}{}^{t1_u.t2_u}.Z^{\delta_k.\beta_k} =$$

$$g^{z_x}.\prod_{i\in I^*}\prod_{i=1,n^*}\left(g^{\left(\frac{a_j}{b_i}\right)^{x'_i}}\prod_{k=1,..,n^*,k\neq j}\left(g^{a^{q+1+j-\frac{k}{b_i}}}\right)^{w_k}\right)^{w^*_{i,j}}. \quad (15)$$

$$\prod_{i\in I^*}\prod_{i=1,n^*}\left(g^{(a_j/b_i)^{y'_i}}\prod_{k=1,..,n^*,k\neq j}\left(g^{a^{q+1+j-k/b_i}}\right)^{w_k}\right)^{w^*_{i,j}}.Z^{\delta_k.\beta_k}$$

$$K3_k = g^{U2} = g^{g_u.a_u} = g^{t1_u.t2_u} = g^{x'.\prod_{i=1,n^*}(g^{a^{q-i+1}})^{w_i}.y'.\prod_{i=1,n^*}(g^{a^{q-i+1}})^{w_i}} \quad (16)$$

$$K4_k = AA2_k = g^{\delta_k.\beta_k}$$

For each x_k in the access structure, $K2_{x_k}$ computation turns harder as the term $g^{a^{q+1/b_i}}$ is hard to compute.

- If the user $\{u_j^*\} \in R^*$, then as explained above all the secret decryption key components are computed. Subsequently, proxy key PXK is generated by the challenger on the request of adversary. If the $\{u_j^*\} \in R^*$, the b component of PXK is invalidated in the game to cause failure of the decryption process. In the same way, if $\{u_j^*\}$ loses x^* attributes that is $S_j' = S_j - x^*$, and the attacker computes valid secret decryption key then also, if S_j' is insufficient to satisfy the challenge access matrix then also PXK fails the decryption process.

Challenge: In the challenge phase, the adversary shares two equal-length messages $M_0, M_1 \epsilon G_T$ along with the access policy (M^*, ρ^*) to the challenger, such that either $u_i^* \epsilon R^*$ *or* S_k do not satisfy access matrix. At this phase, challenger has to flip a coin c to select one of the acquired messages M_c (where $c\epsilon 0,1$). M_c is used to generate the ciphertext.

The challenger computes the ciphertext as follows:

$$C = \mathcal{M}.e(g,g)^s = C = \mathcal{M}.e(g,g)^{va^{q+1}}, C0 = g^{va^{q+1}s} \quad (17)$$

In addition, a vector $\vec{v} = (s, sa + \overline{y_2}, sa^2 + \overline{y_3}, \ldots\ldots, sa^{n-1} + \overline{y_n}) \in Z_p^{n^*}$ is computed, where s is secret that has to be shared and $\overline{y_2}$,……. $\overline{y_n}$ are randomly chosen. Furthermore, the challenger randomly chooses $r'_1, r'_2, \ldots\ldots r'_l$. Subsequently, it generates R_i for $1, \ldots, n^*$. R_i includes all $k \neq i, where\ \rho^*(k) = \rho^*(i)$. Thus, the ciphertext components are computed as follows:

$$C1_{i_k} = g^{\lambda_{i_k}} = \prod_{j=2,..,n^*} g^{M^*_{i,j}.y'_j} \quad (18)$$

$$C2_{i_k} = g^{r_{i_k}} = g^{r'_{i_k}}.g^{sb_{i_k}} \quad (19)$$

$$C3_{i_k} = g^{\lambda_{i_k}}.y_{i_k}^{-r_{i_k}} = y_{i_k}^{r'_i}\left(\prod_{j=2,..,n^*} g^{M^*_{i,j}.y'_j}\right).(g^{sb_i})^{-z_{i_k}}.\left(\prod_{k\in R_i}\prod_{j=1,..n^*}\left(g^{s.(b_i/b_k)}\right)^{M^*_{k,j}}\right) \quad (20)$$

$$C4_{i_k} = g^{-u.r_{i_k}} = g^{r'_{i_k}}.g^{sb_{i_k}}{}^{-u}$$

Phase 2: Same as phase 1

Guess: In phase 2, the adversary has to guess c'. If $c' = c$, and the challenger returns 0 which means $T = e(g,g)^{a^{q+1}s}$ and, 1 indicates T is some random element in G_T and message secrecy is maintained. If $T = e(g,g)^{a^{q+1}s}$, then we get $\Pr[B(\vec{y}, T = e(g,g)^{a^{q+1}s}) = 0] = \frac{1}{2} + Adv_A$, and if T is random element, we have $\Pr[B(\vec{y}, T = R) = 0] = \frac{1}{2}$. Consecutively, it can be stated that the adversary has non-trivial advantage in q-parallel BDHE security game and the proposed framework has been proven secure.

5.2 Proposition 2

The proposed framework is secured against unauthorized access and ensures traceability and data confidentiality.

Provability: In the proposed framework each user is assigned a unique identity component $PK_u = (U0, U1, U2)$. Additionally, the identity component is incorporated in the secret key and the proxy key of the user. Thus, no two users who possess similar attributes can have identical keys. Consequently, the framework allows quick traceability of the malicious user. Moreover, the secret decryption key decrypts the ciphertext only if the key satisfies the access policy. The access policy in the proposed framework has been defined using a linear secret sharing scheme (LSSS) and every LSSS scheme holds linear reconstruction property [25]. The share matrix W of the secret sharing scheme Π has m rows and n cols. Each row of W is mapped to the associated attribute through a mapping function ρ. Thus, $\forall\ i = 1, \ldots\ldots, m$, $\rho(i)$ is the attribute that labels row i. For generating secret shares a vector $\vec{v} = (s, r_2, r_3, \ldots\ldots\ldots, r_n)$ is defined, where $s\epsilon Z_p$ represents shared secret and $r_2, r_3, \ldots\ldots\ldots, r_n \epsilon Z_p$ are randomly chosen numbers, then according to Π, $W.\vec{v}$ is a vector of l shares of secret. Each share $W.\overrightarrow{v_i}$ belongs to attribute $\rho(i)$.

For the access structure $\mathbb{A}$, let $S \in \mathbb{A}$ be any authorized set, and for the access matrix M with m rows and n cols, let $I \subset \{1,2, \ldots\ldots\ldots, m\}$ where, $I = \{i: \rho(i) \epsilon S\}$, there exists constants $w_i \in Z_p$ where $i \in m$. Thus, for valid shares λ_i, $\sum_{i\in I} w_i\lambda_i = s$. Additionally, it is also proved in [25] that the constants $w_i \in Z_p$, where $i \in I$, can be found in polynomial time and for any unauthorized sets, no such w exists.

Thus, the LSSS employability for access structure in the proposed framework ensures data confidentiality and avoids unauthorized access.

5.3 Proposition 3

The proposed framework is resilience to collusion attacks.

Provability: The proposed framework has addressed the possibility of revoked users colluding with each other as well as the capability of the cloud service provider to collude with the revoked users. As in the proposed framework, each user has been assigned a unique identity component PK_u, thus no two users with insufficient attributes can collude. As the users' ID has been embedded in the secret key of the user, thus users cannot combine their attributes in decryption. Furthermore, in the encryption algorithm, the message $\mathcal{M}$ has been blinded while generating ciphertext component $C = \mathcal{M}.e(g,g)^s$. As explained in proposition 2, for the valid value of s, $\sum_{i\in I} w_i\lambda_i = s$. Thus, any user who wants to access the encrypted file and recover $\mathcal{M}$ must recover the component $B = e(g,g)^{U1.s}$ in decrypt module by pairing secret key, proxy key and ciphertext components. If the users possess matching keys, this term will cancel out and message $\mathcal{M}$ is

retrieved else the term cannot be canceled. Thus, if two revoked users attempt to conspire and collude together, while using different IDs, this term do not get cancelled. Consequently, the decryption process fails.

Furthermore, the secret keys and the proxy key are directly distributed to the data user by the authorities. Since the cloud do not participate in any module and do not have any information of the secret key, they cannot collude with the revoked users. Even if they attempt to collude, the time bound proxy key fails the collusion. The proxy key has an associated expiry time with it and thus, the older version of the proxy key is not useful for the revoked user to decrypt the ciphertext. Consequently, the cloud service provider has no means to aid in unauthorized access by colluding with the revoked user. Therefore, our proposed framework achieves full collusion resistance

5.4 Proposition 4

The proposed framework ensures forward and backward secrecy.

Provability: The proxy server ensures forward and backward secrecy in the proposed framework. Backward secrecy implies that the revoked user cannot decrypt files that were earlier accessible. In the proposed framework, to successfully execute the decryption process, the data user is required to acquire proxy key PXK from the proxy server. The generated proxy key is associated to ciphertext as it calculates $C\_1_i = C1_i^{(b_u)}$, where *C1* is the component of requested ciphertext and *b* is the component of private key of user. When the user's access privileges are revoked, the proxy key generated by the proxy server invalidates the component *b* while calculating the C_1_i term. The generated proxy key causes the failure of decryption process and thus, denies revoked user's access to the files which were earlier accessible. Moreover, the proxy key is time bound. Once it is expired, it cannot be reused. The user has to again request for the new proxy key to access encrypted file. Consequently, the proxy key helps to enforce fine-grained access control.

Similarly, the forward secrecy implies denial of access of newly uploaded files to revoked user. Here also the requested proxy key imposes fine-grained access control by causing the failure of decryption process and thus, refraining the revoked users. Consequently, the proposed framework makes it harder for the revoked user to breach security.

6. PERFORMANCE ANALYSIS

In this section, the performance of the proposed PCMR-CPABE framework has been analyzed by comparing the functional specifications with the framework proposed in SEM-ACSIT [19]. Additionally, a comparison of the computational overhead of the two frameworks has been carried out.

Table 2 exhibits that the proposed PCMR-CPABE framework has implemented a decentralized multi-authority CPABE framework, whilst the SEM-ACSIT framework is a centralized multi-authority CPABE framework. Additionally, only the proposed PCMR-CPABE framework addressed revocation both at the system and attribute level and is fully resilient to collusion attacks, whilst the SEM-ACSIT framework did not give attention to the possibility of cloud server turning dishonest and colluding with the revoked user by keeping the older version of ciphertext or key in store. Furthermore, the proposed PCMR-CPABE framework is more computationally efficient than the SEM-ACSIT framework while addressing revocation. The proposed PCMR-CPABE framework has been implemented using the proxy server that controls unauthorized access by revoked or malicious users. In order to prevent revoked users from accessing the ciphertext, the proxy server does not update the ciphertext or the key of non-revoked users. Consequently, it can be stated that the proposed PCMR-CPABE framework is functionally efficient in comparison to the SEM-ACSIT framework.

Table 2. Functionality comparison

Parameters	SEM-ACSIT [19]	PCMR-CPABE
Approach	Centralized	Decentralized
Type of Revocation	Attribute-level Revocation	User and Attribute-level Revocation
Collusion Resistance	Partial	Full
Ciphertext Update	Yes	No
Key update of Non-Revoked Users	Yes	No
Forward and Backward Secrecy	Yes	Yes

Table 3. Comparison of computation overhead

Modules	SEM-ACSIT [19]	PCMR-CPABE
Encryption	$N_k\|P\| + (1 + 5N_{cx_k})\|E\| + 2N_{cx_k}\|M\|$	$\|P\| + (1 + 5N_{cx_k})\|E\| + N_{cx_k}\|M\|$
Decryption	$(N_k + 4N_{ux_k})\|P\| + (2 + 3N_{ux_k})\|M\| + (N_{ux_k} + 1)\|E\|$	$(N_k + 4N_{ux_k})\|P\| + (2 + 3N_{ux_k})\|M\| + (N_{ux_k} + 1)\|E\|$
Key Generation	$(N_k + 4N_{ux_k})\|P\| + (2 + 3N_{ux_k})\|M\| + (N_{ux_k} + 1)\|E\|$	$(5N_k + N_{ux_k})\|E\| + (N_k + N_{ux_k})\|M\|$
Revocation	Attribute-level Revocation 1. Ciphertext Update- $N_{cy_k}(\|E\| + 2\|M\|)$ 2. Key Update- $2\|E\| + (N_{uy_k} + 1)\|M\|$	Proxy Key Generation 1. No Revocation- $(1 + N_{cx_k})\|E\|$ 2. User Revocation- $(1 + N_{cx_k})\|E\|$ 3. Attribute-level Revocation- $(1 + N_{cx_k})\|E\|$

N_{cx_k}: number of attributes in an access policy embedded in the ciphertext. N_k: number of attribute authorities. N_{ux_k}: number of attributes held by a user. N_{cy_k}: number of ciphertext containing revoked attribute y_k . N_{uy_k}: number of non-revoked users holding revoked attribute y_k. |P|: Number of pairing operation. |E|: Number of exponential operation. |M|: Number of multiplication operation.

The computational efficiency of the modules of both frameworks has been compared in Table 3. The computational cost of modules has been calculated in terms of the number of pairing operations, the number of exponential operations, and the number of multiplication operations required for providing a solution to the problem. The encryption module of both framework varies with N_{cx_k} and the encryption module of SEM-ACSIT framework also varies with N_k. The comparison illustrates that the encryption module of the proposed PCMR-CPABE framework requires fewer exponential and multiplication operations to generate ciphertext than SEM-ACSIT framework. In addition, the proposed PCMR-CPABE framework's and the SEM-ACSIT framework's computation costs for the decryption module are equal. Both the modules

vary with N_{ux_k} and N_k.

Furthermore, the secret key generation module of both framework varies in terms of N_k and N_{ux_k}. Although the proposed PCMR-CPABE framework requires more exponential operations in comparison to the SEM-ACSIT framework, the computational cost for the key generation module of the SEM-ACSIT framework increases more than the proposed PCMR-CPABE framework as the number of user attributes N_{ux_k} shows an upward trend. Moreover, a solution only to attribute-level revocation has been proposed by the SEM-ACSIT framework, whilst the PCMR-CPABE framework proposed solution to both the system level and attribute-level revocation. The computational overhead incurred by the revocation module addressing attribute-level revocation in the PCMR-CPABE framework is very low in comparison to the SEM-ACSIT framework as depicted in Table 3. Consequently, it can be stated that the all modules proposed in the PCMR-CPABE framework are more computationally efficient than those of SEM-ACSIT framework.

7. IMPLEMENTATION

The Stanford Pairing-based Crypto Library from the Charm-Crypto framework has been employed to construct the proposed framework [27]. Charm-Crypto framework is a python based framework that provides libraries to support implementation of pairing based cryptography. The proposed framework has employed Toolbox modules from the Charm-Crypto architecture. The Charm-Crypto architecture is based on OpenSSL, GMP and PBC libraries to efficiently construct attribute based encryption schemes. The Toolbox module is a library of various python or C based sub-modules such as pairinggroup, integergroup, Hash, secretutil, msp, etc. These modules help in pairing parameters generation, parsing access policy and implementing attribute based encryption schemes.

The proposed implementation is based on a singular symmetric elliptic curve group ("SS512") of 160-bit order. All the experimental trials have been conducted using Python 3.7.13 over the Oracle Virtual Box 6.1. The virtual platform operates on a Windows 11 machine with an Intel Core i3 processor running at 1.20 GHz and 8 GB of RAM to run Ubuntu 22.04.

The experimental results delineate the computing time required by all the algorithms of the proposed framework on various criteria. Stable experimental results have been obtained by an average of 15 experimental trials. The algorithms of the proposed framework are mainly based on pairing, multiplication and exponentiation operation. The framework has employed randomly selected attribute sets of equal size for keygen module and a text file of 12 kb size for encryption and decryption module to perform experiments. Figure 3 exhibits the computing time consumed by the KeyGen, Encrypt, Decrypt, and ProxKeyGen algorithm when the number of attribute authorities is varied from 2 to 20 against the number of attributes owned by the attribute authorities is fixed to 5. It has been observed that the increase in the number of attribute authorities linearly increases the execution time of algorithms. However, increase in the number of attribute authorities has little impact on the ProxKeyGen algorithm, whilst the encryption and decryption algorithm's computing time grows linearly with the increasing number of attribute authorities. Thus, the observations exhibit the computational efficiency of the proposed framework and conclude that the framework stays computationally efficient even when the load increases in the form of increasing number of attribute authorities.

In addition, Figure 4 expresses the computing time consumed by the Keygen algorithm when attribute authorities are fixed to 5 and user attributes are varied from 5 to 20. The execution time of the KeyGen algorithm grows linearly with the increasing number of user attributes. After comparing Figures 3 and 4, it has been observed that changing the number of attribute authorities has a significant impact on the keygen algorithm's execution time. Figures 5 and 6 depict the execution time of the Encryption and Decryption algorithm. Both the algorithms are dependent on the number of policy attributes. A linear relationship exists between the number of policy attributes and algorithmic computation time. To calculate the experimental findings, policy attributes, in this case were varied from 5 to 20. Since the decryption algorithm varies with the matched user attributes, thus the decryption algorithm consumes less computing time than the encryption algorithm. Furthermore, the Proxkeygen algorithm computing time result has been considered in all the cases viz: no revocation, user revocation, and attribute-level revocation. Figure 7 demonstrates the impact of policy attributes on the computing time of the Proxkeygen algorithm in case of no revocation and user revocation. Both algorithms consume an equivalent amount of computing time. In the same way, Figure 8 depicts the execution time consumed during the attribute-level revocation. In this experiment it has been demonstrated that the execution time of the proxkeygen algorithm increases linearly with the increasing number of revoked attributes. Consequently, it can be stated that the proposed framework avoids heavy computation costs and is efficiently implementable.

Every experimental operation depicts that the increase in the load in terms of number of attributes, number of policy attributes or number of attribute authorities in the algorithm add minimal increase in computational cost due to the presence of proxy server. The incorporation of proxy server eliminated the need of ciphertext re-encryption or key update of non-revoked users with every change in access privileges. The proposed framework also eliminated the need to outsource partial decryption to cloud service provider.

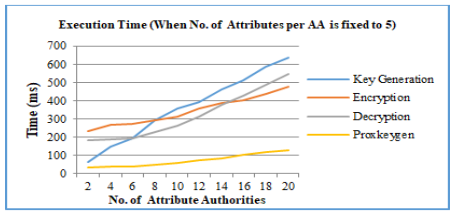

Figure 3. Execution time

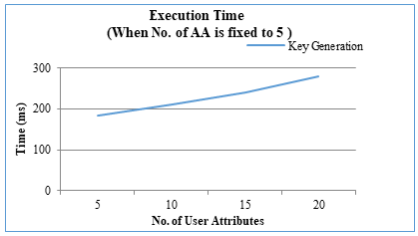

Figure 4. Key generation algorithm

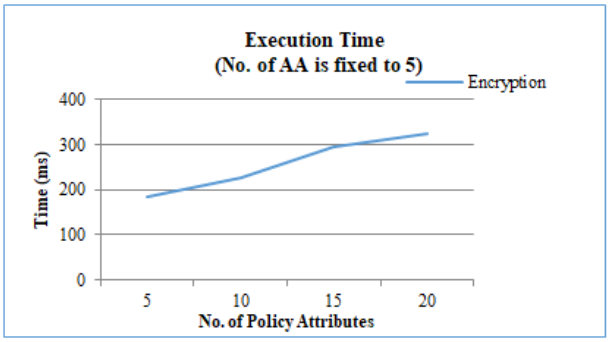

Figure 5. Encryption algorithm

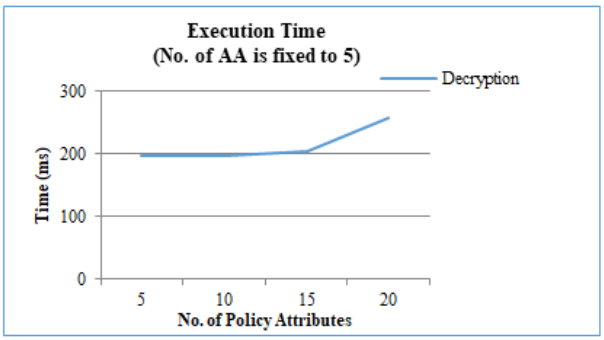

Figure 6. Decryption algorithm

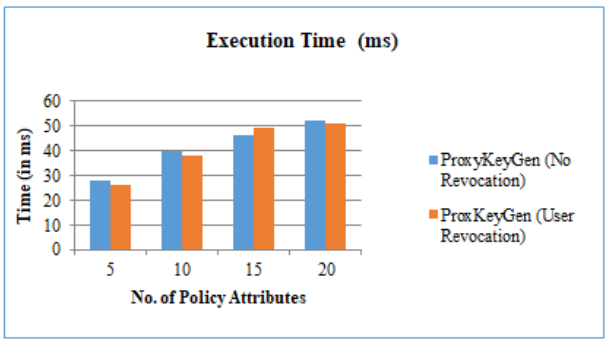

Figure 7. Proxy key generation algorithm (1)

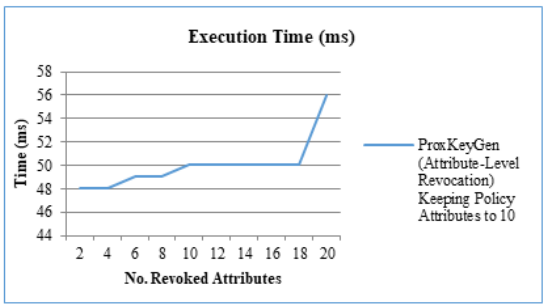

Figure 8. Proxy key generation algorithm (2)

8. CONCLUSION

In the current study challenges pertaining to revoking access rights at the system and attribute level in multi-authority CPABE schemes have been considered. The paper has outlined the research gaps observed in the existing schemes toward addressing collusion attacks, and incorporating dynamicity, and scalability along with forward and backward secrecy. The proposed framework has contributed an efficient proxy based solution to address the revocation of malicious users dynamically. In addition, the proposed framework also suggested a solution to attribute-level revocation. The proxy server in the proposed framework controls the generation of proxy key, and the incorporation of time and identity components in the algorithm allowed the framework to enforce fine-grained access control.

Furthermore, the security analysis of the proposed framework outlined the strength of algorithms in avoiding collusion attacks; both conspired by revoked users or by cloud service providers. The security analysis also presented the strength of the algorithm against unauthorized access and in maintaining forward and backward secrecy. Security analysis has also proved that the proposed framework is secure against q-parallel BDHE assumption.

Moreover, the performance analysis of the proposed framework delineated in detail the computational efficiency of the algorithms in terms of running time. The execution time has been calculated by varying number of attributes, number of attribute authorities and no. of policy attributes. Unlike existing schemes, the proposed framework does not update non-revoked user's or attribute group key to refrain revoked users. It also does not require re-encryption of ciphertext to prevent revoked users from accessing the sensitive data. Consequently, the framework has been proven to be better in terms of computational efficiency and practicability in comparison to the exiting schemes in addressing research challenges pertaining to the enforcement of stringent access control.

In distrustful cloud computing environment, an efficient revocable multi-authority CPABE framework ensures data confidentiality to the data owner and data users. However, pairing-based cryptography is vulnerable to quantum computing. The effective solution to quantum threat is lattice-based cryptography. The proposed framework can be expanded to lattice-based cryptography in the upcoming research work, satisfying all the criteria mentioned in the current study for a successful revocable CPABE scheme.

REFERENCES

[1] Goyal, V., Pandey, O., Sahai, A., Waters, B. (2006). Attribute-based encryption for fine-grained access control of encrypted data. In Proceedings of the 13th ACM Conference on Computer and Communications Security, pp. 89-98. https://doi.org/10.1145/1180405.1180418

[2] Bethencourt, J., Sahai, A., Waters, B. (2007). Ciphertext-policy attribute-based encryption. In 2007 IEEE Symposium on Security and Privacy (SP'07), Berkeley, CA, USA, pp. 321-334. https://doi.org/10.1109/SP.2007.11

[3] Chase, M. (2007). Multi-authority attribute based encryption. In: Vadhan, S.P. (eds) Theory of Cryptography. TCC 2007. Lecture Notes in Computer Science, vol 4392. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-70936-7_28

[4] Lewko, A., Waters, B. (2011). Decentralizing attribute-based encryption. In: Paterson, K.G. (eds) Advances in Cryptology – EUROCRYPT 2011. EUROCRYPT 2011. Lecture Notes in Computer Science, vol 6632. Springer,

Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-20465-4_31
[5] Xu, X., Zhou, J., Wang, X., Zhang, Y. (2016). Multi-authority proxy re-encryption based on CPABE for cloud storage systems. Journal of Systems Engineering and Electronics, 27(1): 211-223.
[6] Al-Dahhan, R.R., Shi, Q., Lee, G.M., Kifayat, K. (2018). Revocable, decentralized multi-Authority access control system. In 2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion), Zurich, Switzerland, pp. 220-225. https://doi.org/10.1109/UCC-Companion.2018.00088
[7] Zhong, H., Zhu, W., Xu, Y., Cui, J. (2018). Multi-authority attribute-based encryption access control scheme with policy hidden for cloud storage. Soft Computing, 22: 243-251. https://doi.org/10.1007/s00500-016-2330-8
[8] Zhang, X., Wu, F., Yao, W., Wang, Z., Wang, W. (2019). Multi-authority attribute-based encryption scheme with constant-size ciphertexts and user revocation. Concurrency and Computation: Practice and Experience, 31(21): e4678. https://doi.org/10.1002/cpe.4678
[9] Wu, Z., Zhang, Y., Xu, E. (2020). Multi-authority revocable access control method based on CP-ABE in NDN. Future Internet, 12(1): 15. https://doi.org/10.3390/fi12010015
[10] Vaanchig, N., Xiong, H., Chen, W., Qin, Z. (2018). Achieving collaborative cloud data storage by scheme with dual-revocation. International Journal of Network Security, 20(1): 95-109. https://doi.org/10.6633/IJNS.201801.20(1).11
[11] Zhang, X., Chen, Y., Yan, X., Jia, H. (2019). Multi-authority attribute-based encryption with user revocation and outsourcing decryption. Journal of Physics: Conference Series, 1302(2): 022026. https://doi.org/10.1088/1742-6596/1302/2/022026
[12] Sethi, K., Pradhan, A., Bera, P. (2021). PMTER-ABE: A practical multi-authority CP-ABE with traceability, revocation and outsourcing decryption for secure access control in cloud systems. Cluster Computing, 24: 1525-1550. https://doi.org/10.1007/s10586-020-03202-2
[13] Li, Q., Ma, J., Li, R., Liu, X., Xiong, J., Chen, D. (2016). Secure, efficient and revocable multi-authority access control system in cloud storage. Computers & Security, 59: 45-59. https://doi.org/10.1016/j.cose.2016.02.002
[14] Yang, K., Liu, Z., Cao, Z., Jia, X., Wong, D.S., Ren, K. (2012). TAAC: Temporal attribute-based access control for multi-authority cloud storage systems. Cryptology ePrint Archive. Cryptol. ePrint Arch.
[15] Kalmani, V.H., Goyal, D., Singla, S. (2015). An efficient and secure solution for attribute revocation problem utilizing CP-ABE scheme in mobile cloud computing. International Journal of Computer Applications, 129(1): 16-21. https://doi.org/10.5120/ijca2015906807
[16] Yang, K., Jia, X., Ren, K., Zhang, B., Xie, R. (2013). DAC-MACS: Effective data access control for multiauthority cloud storage systems. IEEE Transactions on Information Forensics and Security, 8(11): 1790-1801. https://doi.org/10.1109/TIFS.2013.2279531
[17] Liu, Z., Jiang, Z.L., Wang, X., Yiu, S.M. (2018). Practical attribute-based encryption: Outsourcing decryption, attribute revocation and policy updating. Journal of Network and Computer Applications, 108: 112-123. https://doi.org/10.1016/j.jnca.2018.01.016
[18] Tu, S., Waqas, M., Huang, F., Abbas, G., Abbas, Z.H. (2021). A revocable and outsourced multi-authority attribute-based encryption scheme in fog computing. Computer Networks, 195: 108196. https://doi.org/10.1016/j.comnet.2021.108196
[19] Xiong, S., Ni, Q., Wang, L., Wang, Q. (2020). SEM-ACSIT: Secure and efficient multiauthority access control for IoT cloud storage. IEEE Internet of Things Journal, 7(4): 2914-2927. https://doi.org/10.1109/JIOT.2020.2963899
[20] Ramu, G., Reddy, B.E., Jayanthi, A., Prasad, L.N. (2019). Fine-grained access control of EHRs in cloud using CP-ABE with user revocation. Health and Technology, 9(4): 487-496. https://doi.org/10.1007/s12553-019-00304-9
[21] Li, L., Wang, Z., Li, N. (2020). Efficient attribute-based encryption outsourcing scheme with user and attribute revocation for fog-enabled IoT. IEEE Access, 8: 176738-176749. https://doi.org/10.1109/ACCESS.2020.3025140
[22] Zhang, Z., Zhang, W., Qin, Z. (2020). Multi-authority CP-ABE with dynamical revocation in space-air-ground integrated network. In 2020 International Conference on Space-Air-Ground Computing (SAGC), Beijing, China, pp. 76-81. https://doi.org/10.1109/SAGC50777.2020.00026
[23] Huang, K. (2021). Accountable and revocable large universe decentralized multi-authority attribute-based encryption for cloud-aided IoT. IEEE Access, 9: 123786-123804. https://doi.org/10.1109/ACCESS.2021.3110824
[24] Waters, B. (2011). Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In: Catalano, D., Fazio, N., Gennaro, R., Nicolosi, A. (eds) Public Key Cryptography – PKC 2011. PKC 2011. Lecture Notes in Computer Science, vol 6571. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-19379-8_4
[25] Beimel, A. (1996). Secure schemes for secret sharing and key distribution. PhD. Thesis, the Senate of the Technion { Israel Institute of Technology.
[26] Lewko, A., Waters, B. (2011). Unbounded HIBE and Attribute-Based Encryption. In: Paterson, K.G. (eds) Advances in Cryptology – EUROCRYPT 2011. EUROCRYPT 2011. Lecture Notes in Computer Science, vol 6632. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-20465-4_30
[27] Akinyele, J.A., Garman, C., Miers, I., Pagano, M.W., Rushanan, M., Green, M., Rubin, A.D. (2013). Charm: A framework for rapidly prototyping cryptosystems. Journal of Cryptographic Engineering, 3: 111-128. https://doi.org/10.1007/s13389-013-0057-3