Computer Systems Science & Engineering
DOI: 10.32604/csse.2023.033961
Article

Tech Science Press

Secured Access Policy in Ciphertext-Policy Attribute-Based Encryption for Cloud Environment

P. Prathap Nayudu and Krovi Raja Sekhar*

Department of Computer Science and Engineering Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, India
*Corresponding Author: Krovi Raja Sekhar. Email: rajasekhar_cse@kluniversity.in

Received: 02 June 2022; Accepted: 15 November 2022

Abstract: The cloud allows clients to store and share data. Depending on the user's needs, it is imperative to design an effective access control plan to share the information only with approved users. The user loses control of their data when the data is outsourced to the cloud. Therefore, access control mechanisms will become a significant challenging problem. The Ciphertext-Policy Attribute-Based Encryption (CP-ABE) is an essential solution in which the user can control data access. CP-ABE encrypts the data under a limited access policy after the user sets some access policies. The user can decrypt the data if they satisfy the limited access policy. Although CP-ABE is an effective access control program, the privacy of the policy might be compromised by the attackers. Namely, the attackers can gather important information from plain text policy. To address this issue, the SHA-512 algorithm is presented to create a hash code for the user's attributes in this paper. Depending on the created hash codes, an access policy will be formed. It leads to protecting the access policy against attacks. The effectiveness of the proposed scheme is assessed based on decryption time, private key generation time, ciphertext generation time, and data verification time.

Keywords: Cloud computing; access policy; CP-ABE; hash code; SHA-512; attribute; ciphertext; encryption; decryption

1 Introduction

Cloud computing has been one of the most critical information technology techniques since 2007. The government and industry have paid much attention to this technology [1]. There are four main components to enhance the performance of the cloud that are Service-Oriented Architecture (SOA), virtualization, a variety of services, and deployment architecture [2]. It provides services in the form of payment when you use it. Many features are available, including cost, measurement, and on-demand access to use resources efficiently. The cloud offers many advantages to users, but still, there exist some challenges in security and data storage. Data stored in the cloud is generally sensitive and confidential, such as medical data and military information [2]. To protect data, additional security measures like authentication and access control are necessary. It is critical to encrypt users' data before it is outsourced to the cloud to achieve advanced data protection [3].

Several algorithms are proposed and implemented to encrypt user data. Public and secret key algorithms are the most desirable cryptographic algorithms since they hold multiple copies of the same file whenever data is shared across multiple users. To overcome this issue, generating a secret key for every user is the solution in the data-sharing mechanism. The main concerns in data outsourcing are key management and distribution [4]. Many access control mechanisms have been developed since the 1960s [5]. Among them, Bell-la-palda [6] and Biba [6] are well-known access models. Access control models can be implemented in different ways, each with a different control purpose and with different functions and resources.

The attribute-based encryption (ABE) [7] provides a desirable solution to access control issues. ABE has two main variants: KP-ABE (key-principle attribute-based encryption) [8], where access policies are determined based on user keys and data, and CP-ABE [9], related to secret keys and access policies related to ciphertext. Compared to KP-ABE and fuzzy identity-based encryption [10], CP-ABE [11] provides more efficient fine-grained access control. The access policy is encrypted within the CP-ABE. However, CP-ABE ciphertext is considered encrypted by all users. Although the CP-ABE is an effective access control program, it is affected by the policy's privacy protection. Therefore, opponents can gather important information from simple text policy. To address this issue, the SHA-512 algorithm is presented. This algorithm is used to create a hash code for every user's attributes. Depending on the created hash codes, an access policy is created. It leads to protecting the access policy against attacks. The performance of the proposed system is evaluated based on decryption time, private key generation time, ciphertext generation time, and data verification time.

The main objective of this paper is secure data storage in the cloud. Due to security issues, data leakage and attacks are possible on the cloud. To avoid these issues, HCP-ABE is proposed. In this approach, the data is encrypted using the CP-ABE algorithm. To enhance the system's security, the access policy is created based on the user's attributes. These access policies are encrypted using the SHA-512 algorithm. Using this method, attribute information and user data can be secured against attackers. The main contribution of the proposed approach is listed below;

- An access policy structure is generated based on the user's attribute. To improve the access policy's privacy, the SHA-512 algorithm is presented.
- To decrease the loss of data, the CP-ABE algorithm is presented. It is used to encrypt sensitive data.
- The effectiveness of the proposed approach is analyzed based on different metrics such as security level, encryption time, decryption time, uploading time, downloading time, and memory usage.

2 Literature Survey

Many researchers have developed access policies based on secure data transactions on the cloud. Some of the works are analyzed here; Chinnasamy et al. [12] developed an encryption approach based on ciphertext policies for cloud storage. In this approach, the authors developed a CP-ABE model for user data security and privacy enhancement by hiding the access policy. Then, the authors generated a constant-size ciphertext to reduce the storage overhead. Besides, they used a short signature scheme to identify the inside attackers. Furthermore, they developed a secure fine-grained access control system in this approach. They compared the model's performance with other CP-ABE methods, and the results showed the outperformance of the model.

Gafif et al. [13] constructed efficient ciphertext-policy attributes by outsourcing encryption and decryption. In the approach, the authors used two models such as entirely untrusted and semi-trusted ABE. This method is used to decrease the user's encryption and decryption costs. Both programs were selectively chosen-plaintext attack (CPA)-secure in random Oracle, and neither program cooperated with abusive users. The effectiveness analysis showed that both models outperformed the reviewed CP-ABE model based on storage costs, user-side computation, and communication.

Wang et al. [14] introduced a model for cloud computing based on CP-ABE with delegated equality test. In the approach, the authors used a combination of public key encryption with equality test (PKE-ET) and CP-ABE notion. During the representation of the equation test, the cloud server could not obtain knowledge about the message which was encrypted under the access policy. With this approach, the authors developed a concrete CP-ABE-ET program, which provided security proof. From the results, this model achieved better performance.

Raj et al. [15] developed a ciphertext polity attribute simplification based on encryption for multimedia applications. The attribute string was designed in a recommended way that can be used by the data owner, encryption, and encryption process. Besides, they compared the model's effectiveness with various existing methods based on various evaluation metrics. From the results, this model outperformed with the maximum level of security and fewer computational requirements.

Wang et al. [16] developed a traceable attribute-based encryption scheme for cloud storage using ciphertext policies and attribute-level user revocation. In this approach, the authors used linear secret-sharing schemes (LSSS). In this model, the trust authority identifies the culprits and sends the identification of the defect to the attribute manager. After that, each user's identity was distributed to a leaf tip on the key-encryption key (KEK) tree. A secret key update and a cybertext update are used in this model to prevent coordinated attacks between users. Furthermore, the proposed model reduced the computational burden of the data.

Wang et al. [17] introduced an encryption scheme with attribute-level retrieval in cloud storage using CP-ABE. Some attributes are revoked in this approach, which is updated and decrypts the ciphertext successfully. The parallel Billionaire Diffie-Hellman Exponent (BDHE) hypothesis was safe using the selected-structure method. The authors compared the model's effectiveness with the current retrieval method, and this model was made to increase the attribute retrieval expectation and storage space for the private key.

Liu et al. [18] introduced a CP-ABE with a partially hidden access structure in cloud computing. In this approach, each of the attributes has two parts, such as attribute name and value. In the approach, the ciphertext's access structure was not completed by the user's private key attributes. Specific details of the access structure are hidden, while other details are publicly available. Later, the model was used to create an electronic recording system that protects data privacy in the cloud environment. From the results, this model achieved better performance for constructing privacy-preserving Electronic medical records (EMR) systems.

Hwang et al. [19] developed CP-ABE access control to prevent retrieved users from accessing the dynamic cloud. In the paper, the authors defined three critical phases that control the access of deleted users by user registration list and withdrawal list, publishing standard size encryption, and the size of the user's encryption and decryption algorithm. In this approach, the AC server authenticates the user if the user registers and the user received partial encryption. Besides, the user attained the final encryption with a secret key sent from a trusted third party (TTP). The results of the article showed that the storage overhead of the proposed scheme was decreased than the current CP-ABE method.

3 System Model

The proposed system consists of four essential parts, namely cloud service providers (CSP), data owners (DOs), attribute authority (AA), and users. Each part has a different working function. Initially, users register their details on the cloud. After the registration process, the user sends the request to the CSP if they want to access any data from the CSP. The CSP sends the request to the DO. The DO checks the details; if correct, AA sends the secret key to the data user, and CSP sends the requested cipher text to the user. The user decrypts the file using a secret key. The data transmission system model is given in Fig. 1.

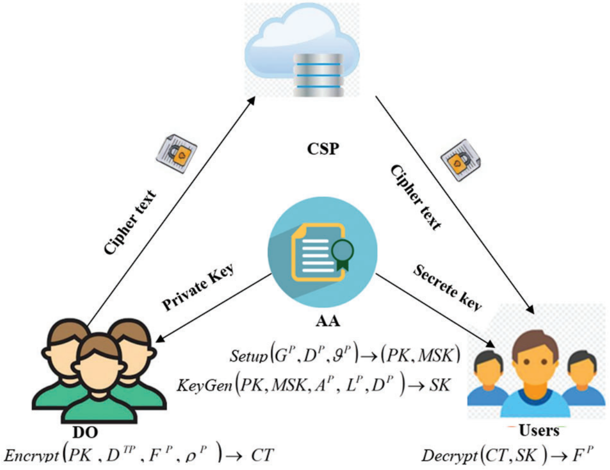
CSP
Cipher text
Cipher text
Private Key
Secrete key
AA
$Setup(G^P, D^P, \vartheta^P) \rightarrow (PK, MSK)$
$KeyGen(PK, MSK, A^P, L^P, D^P) \rightarrow SK$
DO
$Encrypt(PK, D^{TP}, F^P, \rho^P) \rightarrow CT$
Users
$Decrypt(CT, SK) \rightarrow F^P$

Figure 1: System model

In Fig. 1, CSP is used to store data from DOs. Users can retrieve data from the CSP depending on their requests. However, this is not entirely reliable. They need to have data files for their users. DOs' data files are stored in CSP and may be shared with users via CSP. DOs encrypt data files and may define an access policy. Finally, they save the ciphertext (CT) to the CSP. AA is the central hub for the formation of PK and SK. It assigns PK to DO as well as SK to users. Thus each user can attain different access rights depending on their attributes.

The main objective of the proposed methodology is to securely access the data on the cloud using a secured access policy with CP-ABE. Initially, the data stored in the cloud are encrypted using CP-ABE. To enhance the CP-ABE's performance, the access policies are encrypted using the SHA-512 algorithm. It leads to a decrease in the users' information leakage. The proposed system consists of three main stages that are the registration phase, the secure data uploading stage, and the decryption stage. The structure of the proposed methodology is given in Fig. 2.

3.1 Registration Phase

The registration phase is a significant stage for secure data transactions on the cloud. In this phase, initially, the users register their details in the cloud for data access. Before using the system, they must apply for approval as an administrator. Typically, users will be asked to register their username or username, password, age, and gender.

3.1.1 Login Phase

A login is a set of certificates used to verify a customer. Often, this will contain a username with a password. Logins are used to obtain login and command of any PCs or administrations. The client enters his client ID with the secret key for information access. During the verification phase, the system checks whether the user is authorized or not.

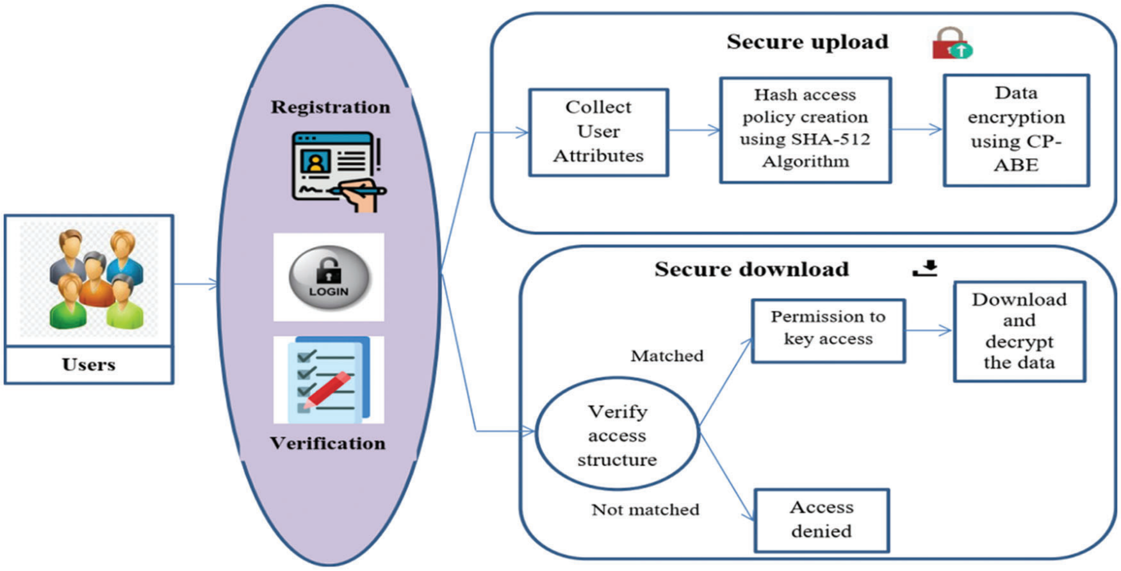

Figure 2: Structure of proposed secure data transmission

3.1.2 Verification Phase

During the verification phase, login details of the user are verified. The user details are correctly verified means, the user is allowed to access the stored data. Else, the user request is rejected. Moreover, the authorized user only decrypts the encrypted data because they only know the hash encryption P_{key}

3.2 Hash Access Policy Generation

The feature sets are given to the SHA-512 algorithm to create a hash access policy. The features are converted into hash code. Then, the access policy is created with the hash code, which is used for future authentication processes. To enhance data confidentiality, the structured access policy is encrypted using the SHA-512 algorithm.

3.2.1 Access Policy Structure

Consider $\{U_1, \ldots, U_n\}$ which denotes the collection of the user's attributes. A collection $C \subseteq 2^{\{U_1, \ldots, U_n\}}$ is called monotonous, for $\forall X$, Y, if $X \in C$ and $X \subseteq Y$, there is $Y \in C$. An access structure is described as the subset of collection C. The collection in C is called the set of authentication and the collection without C is called the unauthorized package. In this work, the structure of access is converted into a Boolean function.

3.2.2 Secure Access Policy Creation Using SHA-512 Algorithm

The access policy is developed based on user attributes. The access policy contains sensitive information. So, the created access policy is encrypted using the SHA-512 technique. The SHA-512 technique is developed from the SHA-1. The function of the SHA-512 algorithm is most similar to SHA-2. The introduced SHA-512 technique utilizes a volume of 1024 bits and it accepts a large 2^128 bits long string as input. In this work, the attributes are combined with extra bits to create multiples of 1024 bits. The generated blocks are divided into smaller parts of 1024 bits. In this, the first volume is connected to the vector initialization, and the hash code (HC) is created. Subsequent blocks are linked to previously created HCs. The access policy is formulated based on these HCs. SHA-512 has the following operating principle, as shown in Fig. 3.

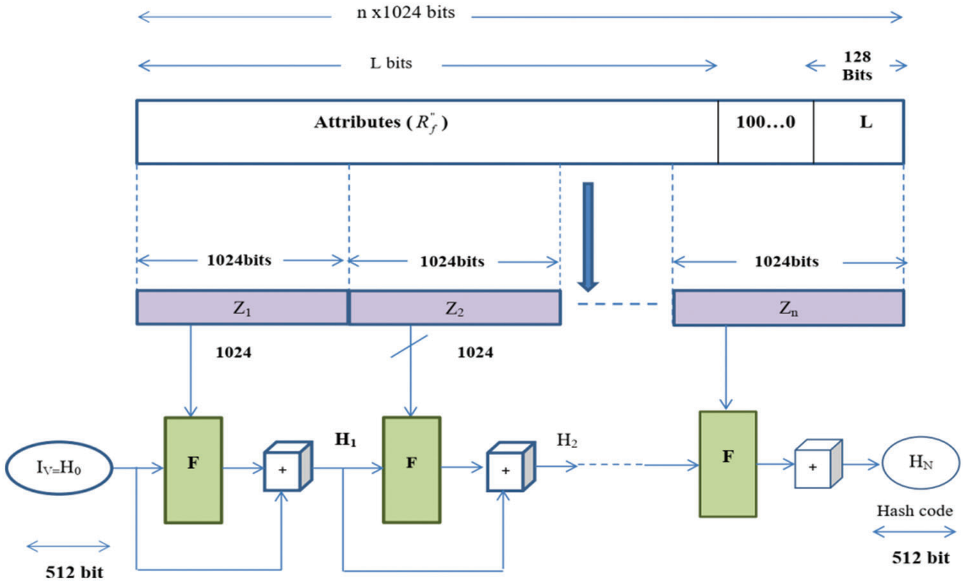

Figure 3: Structure of the proposed SHA-512 algorithm

The technique consists of four stages: input design, hash cache initialization, message processing, and output. Each stage explanation is given below;

- **Input formatting**

Consider the input message and check that the message size is suitable for further processing. If the size of the input message is sufficient, any padding bits will not be added to it; otherwise, padding bits will be used to get the required size. Typically, the padding bits are '1' and multiple '0's (100000 … 000). Also, according to SHA-512, there should be even a bit of padding. So a single dump bit would simply be '1'. Here, it connects the 128-bit module with the input message. Also, this module contains the unsigned 128-bit integer (originally the most significant byte) and the length of the original input message (before dumping). The end of those two steps will reach a message with a length of 1024 bits. The expanded message is signified as the sequence of 1024 bits blocks z_1, z_2, z_3,…, z_n, and its total length becomes $a \times 1024$ bits.

- **Hash buffers initialization**

It executes each message block of 1024 bits using the result of the previous block. The intermediary and outcomes of the hash function are kept in the 512 bit buffer. This buffer can be represented by 64-bit records, such as "m, n, o, p, q, m, r, s, t", which are initialized to 64-bit integers (hexadecimal values). The big-endian form, which stores a word's utmost significant byte in the leftmost (low-address) byte position, is utilized for storing these values. These words are obtained by taking the first 64 bits of fractions of the square root values of the first eight prime integers.

- **Message processing**

A volume of 1024 bits is processed at a time. This level consists of 80 rounds, each of which takes up 512 bits of buffer value "m, n, o, p, q, r, s, t" and enhances the buffer contents.

- **Output**

After each block completes the message processing phase, the final 512-bit hash value R''_f is obtained. The behavior of SHA512 is examined as follows,

$$H_0 = F_v \tag{1}$$

$$H_u - SUM_{64}(H_{u-1},\ mnopqrt_u) \tag{2}$$

$$AP = H_P \tag{3}$$

where;

F_v→Initial value of the "m, n, o, p, q, r, s, t" buffer

$mnopqrst_u$→Output of the last round of processing of the u^{th} message block

P→Number of blocks in the R''_f (encompassing padding and length fields)

SUM_{64}→Addition modulo 2^{64} performed individually on each word of the pair of inputs

AP→Final hash value of R''_f, this is a secret key for subsequent processing phases

The above steps are used to generate the hash code of the access policy. This access policy is used for encryption and decryption.

3.3 Data Security Using CP-ABE

To decrease the loss of data, the DO's data is securely stored in the cloud. For security purposes, in this paper HCP-ABE algorithm is utilized.

3.3.1 Preliminary

Let's consider two multiplicative cyclic groups G_0 and G_1 of prime order p. Here, g is a generator of G_0 and e is a bilinear map, i.e., $e\colon G_0 \times G_0 \to G_1$. The significant properties of the bilinear map are described below:

1. *Bi-linearity*: $e\left(x^a,\ y^b\right) = e(x,\ y)^{ab}, \qquad x,\ y \in G_0\ \ and\ a,\ b \in Z_p$
2. *Non-degeneracy*: $e(g,\ g) \neq 1$
3. *The ability for computation*: $e(g_1,\ g_2)$ can compute for all $g_1,\ g_2 \in G_0$

3.3.2 Operations of the Proposed CP-ABE

The proposed CP-APE includes four operations that are set up, key generation, encryption, decryption, and delegate. The basis of these operations is described below:

Set up $(G^P,\ D^P,\ \vartheta^P) \to (PK,\ MSK)$: The AA executes the setup of this algorithm. In this setup, the global attribute G^{P}, the depth of NTT $D^{\mathrm{P},}$ and the security parameter ϑ^P are input parameters. Using these input parameters, the output parameters such as the public key (PK) and master secrete key (MSK) are calculated.

KeyGen $(PK,\ MSK,\ A^P,\ L^P,\ D^P) \to SK$: The AA executes the generation of SK or private key. For SK generation, the following parameters are used: PK, MSK, attribute set $\mathrm{A^P}$ of the user, location attribute $\mathrm{L^{P,}}$ and valid period $\mathrm{D^P}$. After the execution of the key generation algorithm, the SK of the user can be attained.

Encrypt $(PK,\ D^{TP},\ F^P,\ \rho^P) \to CT$: The DO executes the encryption algorithm on the data. Using the input parameters such as PK, decrypt time D^{TP}, data $\mathrm{F^{P,}}$ and access policy ρ^P, the encryption algorithm is executed. Finally, the DO attains encrypted data or cipher text (CT).

Decrypt $(CT,\ SK) \rightarrow F^P$: The user executes the decryption algorithm. In this phase, the user decrypts the CT using SK. Initially, the valid access time D^P of the user should match the D^{TP} of CT, and the attribute set AP and location attribute LP of the user should match the defined access policy ρ^P to decrypt the search data F^P. The decryption algorithm will be terminated if they do not match.

Delegate $(SK,\ A') \rightarrow SK'$: For the set of attributes A^P, secret key, or private key is denoted as SK. For the subset of attribute A', secret key SK' is generated using this Delegate function if A' is satisfied $A' \subseteq A^P$. This function is used for the extension of AA.

```
Algorithm: CP-ABE
Stage 1: System setup
Input: $G^P,\ D^P,\ \vartheta^P$
Output: $PK,\ MSK$
    1. Consider two multiplicative cyclic groups $G_0$ and $G_1$ of prime order p.
    2. Bilinear map, $e: G_0 \times G_0 \rightarrow G_1$, here $G_0 = \langle g \rangle$
    3. Express time as a string s of $\eta$ elements.
    4. Choose randomly $h_1,\ \ldots,\ h_G \in G_0$ and $R_1,\ \ldots,\ R_D \in G_0$
    5. Choose randomly $\alpha,\ \beta \in Z_p$
    6. Estimate $g^\beta,\ g^{\frac{1}{\beta}}\ and\ e(g,\ g)^\alpha$
    7. $MSK \rightarrow \alpha$
    8. $PK \rightarrow \left\{G_0,\ G_1,\ g,\ g^\beta,\ g^{\frac{1}{\beta}},\ e(g,\ g)^\alpha,\ h_1,\ \ldots,\ h_G,\ R_1,\ \ldots,\ R_D\right\}$
Return PK, MSK
Stage 2: Key generation
Input: $PK,\ MSK,\ A^P,\ L^P,\ D^P$
Output: $SK$
        1. Consider T $\rightarrow$ a minimum cover set of $D^P$.
        2. T has multiple time intervals represented as $s = \left(s_{\chi_1},\ s_{\chi_2},\ \ldots,\ s_{\chi_t}\right)$ of $\eta$ elements, here $\chi_1 < D^P$
        3. Choose randomly $r,\ r_j \in Z_p$
        4. Estimate $E_0 \rightarrow g^r;\ \ E_1 \rightarrow g^{\frac{(\alpha+r)}{\beta}};\ \ \{E_{0,j} \rightarrow g^{r_j}\}_{j\in T};\ \ \left\{E_{1,j} \rightarrow g^\alpha g^{\beta r} g^{\frac{(\alpha+r)}{\beta}}\left(R_0 \prod_{k=1}^{\chi_t} R_k^{kj}\right)\right\}_{j\in T};$
           $\{K_v \rightarrow h_v^r\}_{v\in G}$
        5. $SK \rightarrow \left\{E_0,\ E_1,\ \{E_{0,j},\ E_{1,j}\}_{j\in T};\ \{K_v\}_{v\in G}\right\}$
Return SK
Stage 3: Encryption
Input: $PK,\ D^{TP},\ F^P,\ \rho^P$
Output: CT
      1. Express decrypt time $D^{TP}$ as $s_c = (s_1,\ s_2,\ \ldots,\ s_t)$ for $\eta$ elements, here $t < D,\ t \in Z_p$ denotes the depth of $D^{TP}$ in NTT.
      2. Generate access tree $\rho^P$ for the dataset $L_w$, here $L_w$ denotes set of leaf nodes and $\vartheta_w$ denotes the root of $\rho^P$.
```

(Continued)

Algorithm: (continued)

```
    3. Choose a vector $q = (m,\ l_2,\ \ldots,\ l_n) \in Z_p^n$ randomly.
    4. Estimate $\gamma_i = Q_i \bullet q$ for $i = 1,\ \ldots,\ l$
    5. Calculate $C_0 \rightarrow F^P e(g,\ g)^{\alpha m};\ \ C_1 \rightarrow g^m;\ \ C_2 \rightarrow \left(R_0 \prod_{k=1}^{t} R_k^{kj}\right);\ \ \left\{C'_i \rightarrow g^{\beta\gamma_i} h_{\rho(i)}^{-m}\right\}_{i=1,\ldots,l}$
    6. $CT \rightarrow \left\{C_0,\ C_1,\ C_2,\ \{C'_i\}_{i=1,\ldots,l}\right\}$
    7. $C_F \rightarrow \{CT,\ \rho^P\}$
Return $C_F$
Stage 4: Decryption
Input: $CT,\ SK,\ C_F$
Output: F
        1. Attain $\rho$ and $L_w$ from $C_F$.
        2. Check the data in $L_w$ and attain the root $\vartheta'_w$
        3. Compare $\vartheta'_w$ and $\vartheta_w$
        4. If the attribute set A of the user can't satisfy the access policy Program is terminated
        5. Else compute $\varphi \rightarrow \prod_i \left(e\left(E_0,\ C'_i\right) \cdot e\left(K_{\rho(i)},\ C_1\right)\right)^{\tau_i},\quad \tau_i \in Z_p$
        6. End
        7. If $s_c = (s_1,\ s_2,\ \ldots\ s_t) \in \mathrm{T}$
        8. Then compute $Dec = \frac{C_0 \cdot \varphi \cdot e(C_1,\ E_1) \cdot e\left(C_2,\ E_{0,s_c}\right)}{e\left(C_1,\ E_{1,s_c}\right)}$
        9. End
        10. $Dec(L_w) \rightarrow F$
Return F
```

4 Results and Discussion

The experimental results of the proposed approach are analyzed in this section. This proposed scheme is implemented using Java with the Windows 7 operating system on a 2 GHz dual-core PC machine with 4 GB of main memory. The performance of the proposed HCP-ABE is analyzed based on encryption time, decryption time, key build time, security level, memory usage in encryption, and memory usage in decryption. Besides, the performance of the proposed HCP-ABE is compared with that of the conventional CP-ABE and ABE schemes.

Security is one of the most critical parameters in the cloud because the cloud is an unreliable network. For security purposes, in this paper, the HCP-ABE algorithm is utilized. Fig. 4 shows the security level of different methods. When analyzing Fig. 4, the proposed method attained a high-security level compared to the other two methods, namely CP-ABE-based security and ABE-based security. Also, from the figure, ABE-based security attained the worst security level compared to CP-ABE and HCP-ABE.

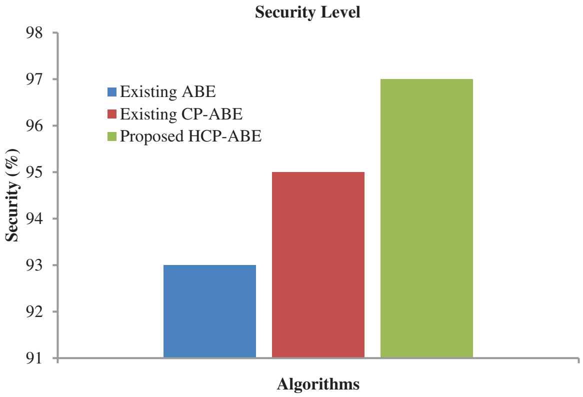

Figure 4: Performance analysis based on the security level

During the data access, the following processes are done that are key generation, cipher text creation, and cipher text verification time. Each process consumes a different time to process. In Fig. 5, the amount of time taken for different processes is analyzed. The figure clearly shows that for the key generation process, 675 ms time is taken, for cipher text creation, 712 ms time is taken, and 824 ms time is taken for the cipher text verification process.

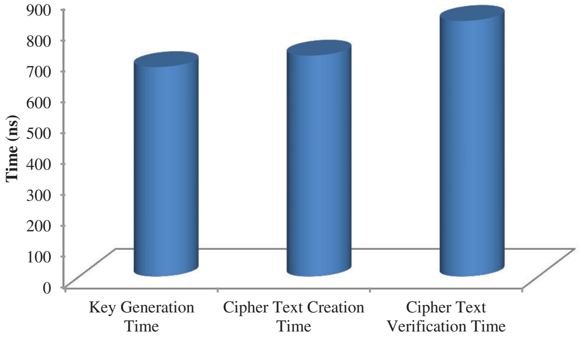

Figure 5: Performance analysis based on time

In Fig. 6, the performance of the proposed approach is analyzed based on the encryption time. For encrypting 1000 data, our proposed approach took 457 ns which is 237 ns for CP-ABE-based data encryption and 683 ns for ABE-based data security. According to the above graph, it is clear that the encryption time also gradually increases as the data sizes increase.

In Fig. 7, the performance of the proposed approach is analyzed based on the decryption time. Analysing Fig. 7, it is clear that the suggested method takes 3867 ns to encrypt 5000 data, which is low compared to traditional CP-ABE and ABE.

In Fig. 8, the effectiveness of the suggested approach is analyzed based on the data uploading time. Analyzing Fig. 8, it is clear that the smaller the data the less time it takes and the larger the time it takes to upload large amounts of data.

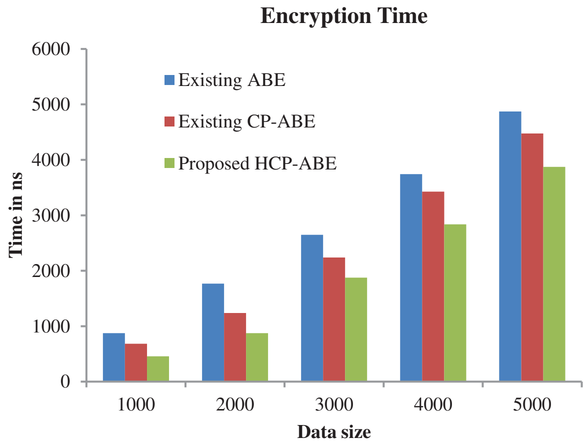

Figure 6: Performance analysis based on encryption time

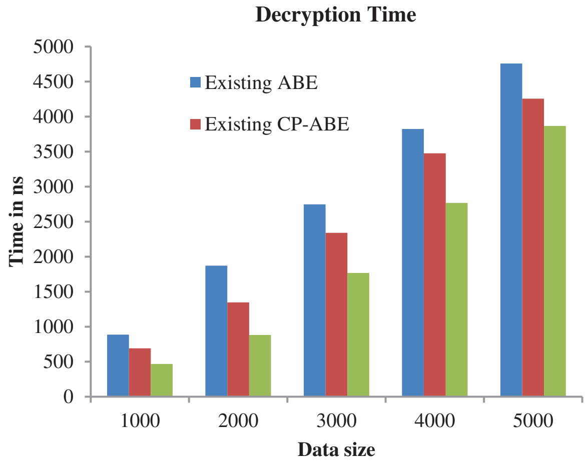

Figure 7: Performance analysis based on decryption time

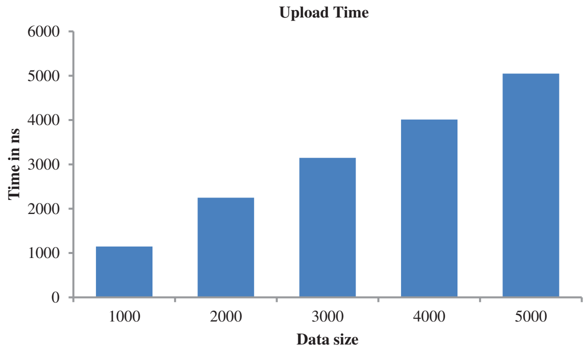

Figure 8: Performance analysis based on upload time

Similarly, the performance of the proposed approach is analyzed based on download time as given in Fig. 9. Here also, the algorithm takes the maximum time when downloading a large file.

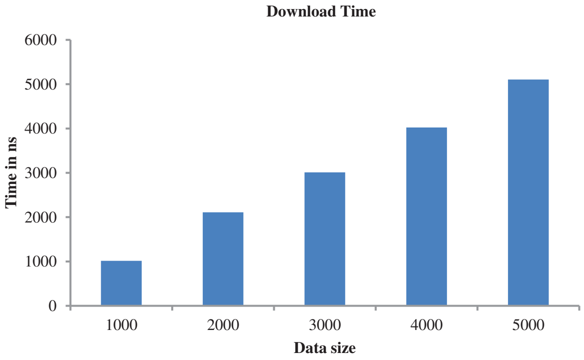

Figure 9: Performance analysis based on download time

Fig. 10 shows the amount of memory usage. To prove the efficiency of the proposed approach, it is compared with different methods. When analyzing Fig. 10, encrypting 5k data, the proposed method utilizes only 76853642-kilo bytes which are 85517741-kilo bytes for CP-ABE-based encryption and 96857874-kilo bytes for ABE-based encryption.

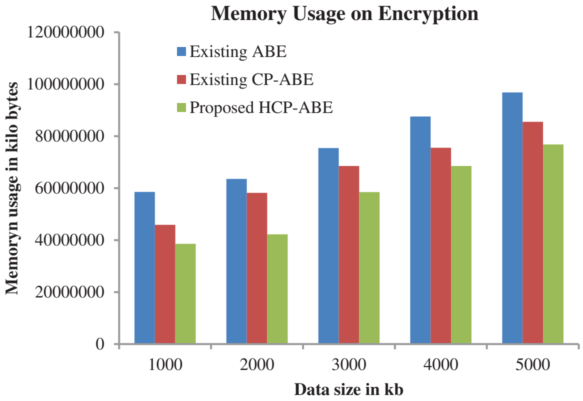

Figure 10: Performance analysis based on memory usage on encryption

Moreover, in Fig. 11, the performance of the proposed approach is analyzed based on memory usage in the decryption process. From the analysis, the proposed method attained better results compared to the traditional CP-ABE and ABE.

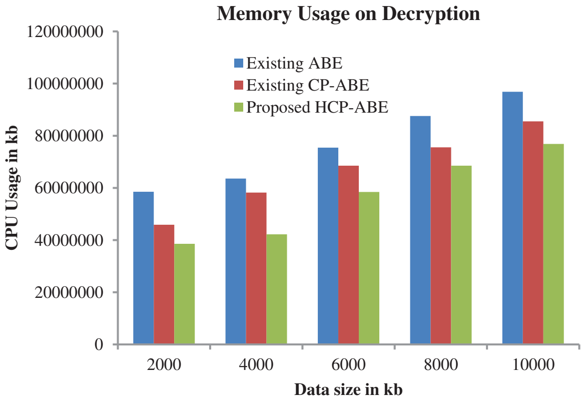

Figure 11: Performance analysis based on memory usage in decryption

5 Conclusion

An efficient HCP-ABE algorithm-based secure data transaction on the cloud has been presented in this work. To prevent the unauthorized login process, an authentication system with a secure access policy has been presented. The access policy has been generated based on the user attributes, mainly location and time attributes. To enhance the system's security, the access policy has been encrypted using the SHA-512 algorithm. The encryption process has been carried out using the HCP-ABE algorithm. The performance of the proposed approach is analyzed based on the different metrics and effectiveness compared with other methods. The proposed HCP-ABE will take 457 ms encryption time and 468 ms encryption time, and those are minimal considering the existing algorithms. Moreover, this proposed work achieves a maximum security level of 97%.

Acknowledgement: The author with a deep sense of gratitude would thank the supervisor for his guidance and constant support rendered during this research.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] V. Ratten, "Cloud computing technology innovation advances: A set of research propositions," *International Journal of Cloud Applications and Computing (IJCAC)*, vol. 5, no. 1, pp. 69–76, 2015.

[2] R. Buyya, V. Christian and S. T. Selvi, "Mastering cloud computing: Foundations and applications programming," *Newnes*, 2013.

[3] W. Wu, Q. Zhang and Y. Wang, "Public cloud security protection research," in *2019 IEEE Int. Conf. on Signal Processing, Communications and Computing (ICSPCC)*, Dalian, China, pp. 1–4. IEEE, 2019.

[4] M. Kallahalla, E. Riedel, R. Swaminathan, Q. Wang and K. Fu, "Plutus: Scalable secure file sharing on untrusted storage," in *2nd USENIX Conf. on File and Storage Technologies (FAST 03)*, pp. 29–42, 2003.

[5] R. Anderson, "*Security Engineering: A Guide to Building Dependable Distributed Systems*," John Wiley & Sons, 2020.

[6] D. E. Bell, "Looking back at the bell-la padula model," in *21st Annual Computer Security Applications Conf. (ACSAC'05)*, Tucson, AZ, USA, IEEE, pp. 15, 2005.

[7] G. Wang, Q. Liu and J. Wu, "Achieving fine-grained access control for secure data sharing on cloud servers," *Concurrency and Computation: Practice and Experience*, vol. 23, no. 12, pp. 1443–1464, 2011.

[8] H. Zhu, L. Wang, H. Ahmad and X. Niu, "Key-policy attribute-based encryption with equality test in cloud computing," *IEEE Access*, vol. 5, pp. 20428–20439, 2017.

[9] X. Huang, W. Susilo, Y. Mu and F. Zhang, "Short designated verifier signature scheme and its identity-based variant," vol. 6, no. 1, pp. 82–93, 2008.

[10] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Annual Int. Conf. on the Theory and Applications of Cryptographic Techniques*, Berlin, Heidelberg, Springer, pp. 457–473, 2005.

[11] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in *Int. Workshop on Public Key Cryptography*, Berlin, Heidelberg, Springer, pp. 53–70, 2011.

[12] P. Chinnasamy, P. Deepalakshmi, A. K. Dutta, J. You and G. P. Joshi, "Ciphertext-policy attribute-based encryption for cloud storage: Toward data privacy and authentication in AI-enabled IoT system," *Mathematics*, vol. 10, no. 1, pp. 68, 2021.

[13] H. El Gafif and A. Toumanari, "Efficient ciphertext-policy attribute-based encryption constructions with outsourced encryption and decryption," *Security and Communication Networks*, vol. 2021, pp. 1–17, 2021.

[14] Q. Wang, L. Peng, H. Xiong, J. Sun and Z. Qin, "Ciphertext-policy attribute-based encryption with delegated equality test in cloud computing," *IEEE Access*, vol. 6, pp. 760–771, 2017.

[15] J. J. D. Raj and J. S. Immanuel, "Simplified ciphertext policy attribute based encryption for multimedia applications," *Procedia Computer Science*, vol. 171, pp. 2713–2719, 2020.

[16] S. Wang, K. Guo and Y. Zhang, "Traceable ciphertext-policy attribute-based encryption scheme with attribute level user revocation for cloud storage," *PloS one*, vol. 13, no. 9, pp. e0203225, 2018.

[17] G. Wang and J. Wang, "Research on ciphertext-policy attribute-based encryption with attribute level user revocation in cloud storage," *Mathematical Problems in Engineering*, vol. 2017, pp. 1–12, 2017.

[18] L. Liu, J. Lai, R. H. Deng and Y. Li, "Ciphertext-policy attribute-based encryption with partially hidden access structure and its application to privacy-preserving electronic medical record system in cloud environment," *Security and Communication Networks*, vol. 9, no. 18, pp. 4897–4913, 2016.

[19] Y. W. Hwang and I. Y. Lee, "CP-ABE access control that block access of withdrawn users in dynamic cloud," *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 14, no. 10, pp. 4136–4156, 2020.