Intelligent Automation & Soft Computing
DOI: 10.32604/iasc.2023.032267
Article

Tech Science Press

Dynamic Time and Location Information in Ciphertext-Policy Attribute-Based Encryption with Multi-Authorization

P. Prathap Nayudu and Krovi Raja Sekhar*

Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, AP, 522 502, India
*Corresponding Author: Krovi Raja Sekhar. Email: rajasekhar_cse@kluniversity.in
Received: 12 May 2022; Accepted: 16 June 2022

Abstract: Due to the mobility of users in an organization, inclusion of dynamic attributes such as time and location becomes the major challenge in Ciphertext-Policy Attribute-Based Encryption (CP-ABE). By considering this challenge; we focus to present dynamic time and location information in CP-ABE with multi-authorization. At first, along with the set of attributes of the users, their corresponding location is also embedded. Geohash is used to encode the latitude and longitude of the user's position. Then, decrypt time period and access time period of users are defined using the new time tree (NTT) structure. The NTT sets the encrypted duration of the encrypted data and the valid access time of the private key on the data user's private key. Besides, single authorization of attribute authority (AA) is extended as multi authorization for enhancing the effectiveness of key generation. Simulation results depict that the proposed CP-ABE achieves better encryption time, decryption time, security level and memory usage. Namely, encryption time and decryption time of the proposed CP-ABE are reduced to 19% and 16% than that of existing CP-ABE scheme.

Keywords: CP-ABE; geohash; new time tree (NTT); multi authorization; dynamic attribute

1 Introduction

Data on the cloud is vast and growing rapidly [1–3]. These cloud data need to be shared often. Some emerging technologies make it easier to do this. CP-ABE [4–6] is one of the relatively recent developments. This allows the user to specify the access policy within the data encryption process. Access to data can be restricted using accessibility policies. User attributes in their private keys are checked against the access policy in order to determine if they have access to the file. They may include certain characteristics of users, such as date of birth, gender or domain/application specific attributes status within the organization, e.g., employee, manager or external contractor [7].

There are problems with using ABE, however, such as the fact that the attributes assigned to a user are static. The user needs a new private key in order to add or change an attribute. Time-related data: This prohibits the use of frequently changing attributes such as day time or working days. It would be very useful to reveal that specific files can only be decrypted periodically. Another property that changes

frequently is the location of the mobile user. As the number of mobile devices increases, users have the ability to collect information about their location at any time. It is desirable to use this information as part of an encryption plan. Besides, in the cloud environment, when a large number of users have access to a data in a short period of time, it is a big challenge for the Attribute Authentication Center. Therefore, to solve these problems, the following contributions are made in this paper.

- In this approach, location of a user can be considered as an attribute along with the other attributes of the user. Here, latitude and longitude of the user's position is encoded using Geohash.
- A new time tree (NTT) is provided, which sets the encrypted duration of the encrypted data and the valid access time of the private key on the data user's private key.
- To enhance the efficiency of key generation, multi-authorization is extended in this work.
- The performance of the proposed scheme is calculated in terms of decryption time, private key generation time, ciphertext generation time and data verification time.

Remaining sections of the article are organized as follows: Section 2 reviews some recent literatures which did the research on CP-ABE based security enhancement in cloud environment. Section 3 proposes Dynamic Time and Location Information in CP-ABE with multi-authorization. The results of the proposed approach are discussed in Section 4. The conclusion of the research work is described in Section 5.

2 Related Works

This section examines some recent literature that focuses on cloud security using CP-ABE. In these literatures, the authors had presented new technique with CP-ABE to enhance the security level in cloud environment. Namely, DilipVenkata Kumar Vengala, D. Kavitha and A. P. Siva Kumar [8] had the goal to solve the issues of key secrecy expose and key intricacy in prevailing schemes. To attain this goal, the authors had introduced a secure data transmission method along with the data deduplication and a distributed cloud server. At first, they had extracted the features with the support of cloud server. Then, the features have been selected optimally using the Hybrid Meerkat clan algorithm. Data deduplication was performed using the SHA512 algorithm. The input file was compressed by a two-stage lempel–ziv algorithm and was encrypted by optimized CP-ABE elliptical curve cryptography algorithm. As a result of this proposed program, authors increased the level of protection.

Zhao et al. [9] had the objective to revoke the users to access the encrypted data as well as to reduce the computational cost of ABE. To achieve the objective, the authors had proposed an effective and revocable storage CP-ABE with outsourced decryption. Besides, they proposed standard size cyber texts and secret keys. The authors had achieved the revocable storage by using the chinese remainder theorem. In addition, the local functions outsourced ABE framework was reduced to a standard size with cyber texts and secret keys. In the approach, decryptor was used to outsource the computing process to the outsourcing service providers.

ShuminXuea and ChengjuanRen [10] had aimed to increase the security level for data sharing in cloud environment. To achieve this aim, the authors had presented an enhanced, effective data encryption scheme. The encryption scheme was performed based on CP-ABE and cost of controlled time utilizing fixed-length ciphertext. Besides, encryption cost was reduced by combining fixed-length encryption scheme with the CP-ABE. The enhanced algorithm performed better when using multi-users attributes and huge data. By presenting the proposed algorithm, the authors had attained high reliability in cloud computing environment.

Privacy and access control are the major security concerns for big data. Although CP-ABE is an efficient algorithm, encryption time and decryption time are high for big data. So, Praveen Kumar Premkamal, Syam Kumar Pasupuleti and P. J. A. Alphonse [11] had introduced a novel verifiable outsourced CP-ABE to

achieve privacy and access control for big data. Also, heavy computations in the cloud were outsourced to proxy server because of that encryption overhead and decryption overhead were reduced. Using this outsourcing computation, the correctness of the data was verified. The proposed scheme had limited the number of data access. Due to this proposed scheme, computation time to access the encrypted data was decreased.

Conventional CP-ABE algorithms may not consider hierarchical relationships of multiple access structures. It just needs to originate multiple ciphertexts to attain the requirement of hierarchical access because of that computational overhead is increased. So, He et al. [12] had presented an effective hierarchical CP-ABE algorithm to reach fine-grained access control of multiple hierarchical files efficiently. The access structure of the proposed hierarchical CP-ABE algorithm was linear secret sharing method. Besides, the authors had developed an attribute-based hierarchical access control. By presenting this algorithm, the authors had decreased the computational overhead.

Sheng Ding, Chen Li and Hui Li [13] had the goal to reduce the computation overhead of bilinear pairing in ABE. To attain this goal, they had presented CP-ABE with elliptic curve cryptography based a new pairing-free data access control method. They had reduced the computation overhead by presenting effective scalar multiplication on elliptic curves instead of bilinear pairing. Besides, the authors had designed a novel key distribution. During the phase of attribute revocation, an attribute or a user was revoked without updating the keys of other users. In addition, the access policy expressiveness was enhanced by presenting linear secret sharing method. Because of the proposed scheme the efficiency and security of the system were improved.

In recent decades, secure file sharing against the unauthorized or malicious user is the major challenge in the cloud environment. To attain the fine grained access control, attribute-based signcryption algorithm is considered as the efficient cryptographic algorithm. This, attribute-based signcryption algorithm includes ABE and attribute-based signatures for achieving privacy-oriented confidentiality with the authenticity. Nevertheless computational overhead of those algorithms are very high due to the size of ciphertext and key length. So, Mythili et al. [14] had proposed a novel attribute-based signcryption algorithm. Besides, they had presented more expressive hypergraph access structure known as Attribute HyperGraph. Due to the proposed scheme, computation overhead of the system had been reduced.

3 Proposed Methodology

3.1 Overview

Fig. 1 depicts the overall flow diagram of the proposed CP-ABE. In the approach, along with the user's attribute dynamic attribute such as location information is included to create the access policy. To encode the location information, Geohash technique is used. Using Geohash, the longitude and latitude of user's position are encoded into bit string. Besides, access time period or valid decrypt time period of each user is defined with the access policy using the new time tree (NTT) structure. In addition, multiple attribute authorities (MAA) are used in this work to enhance the key generation efficiency. Each AA provides the public key (PK) to the data owner which encrypt the data along with the access policy and valid time period CP-ABE as well as it provides secret key (SK) to the users. Utilizing SK, the user can decrypt the data if the attributes of them satisfies the access policy and valid time period.

3.2 System Model

System model of CP-ABE is illustrated in Fig. 2. As depicted in the figure, the model has four significant parts that are cloud service provider (CSP), data owner (DO), attribute authority (AA) and users. The operation of each user is described as follows,

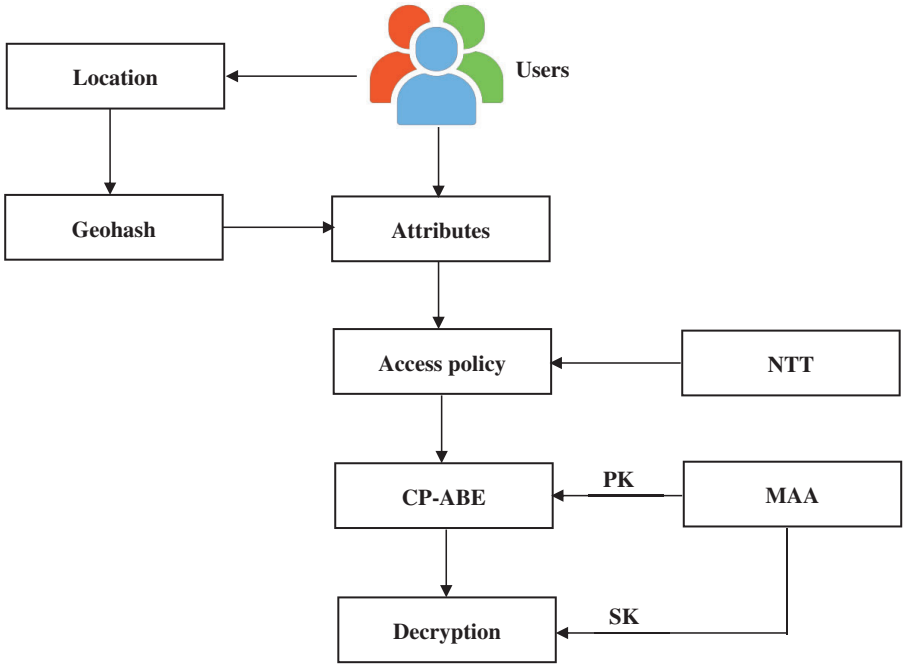

Figure 1: The overall flow diagram of the proposed CP-ABE

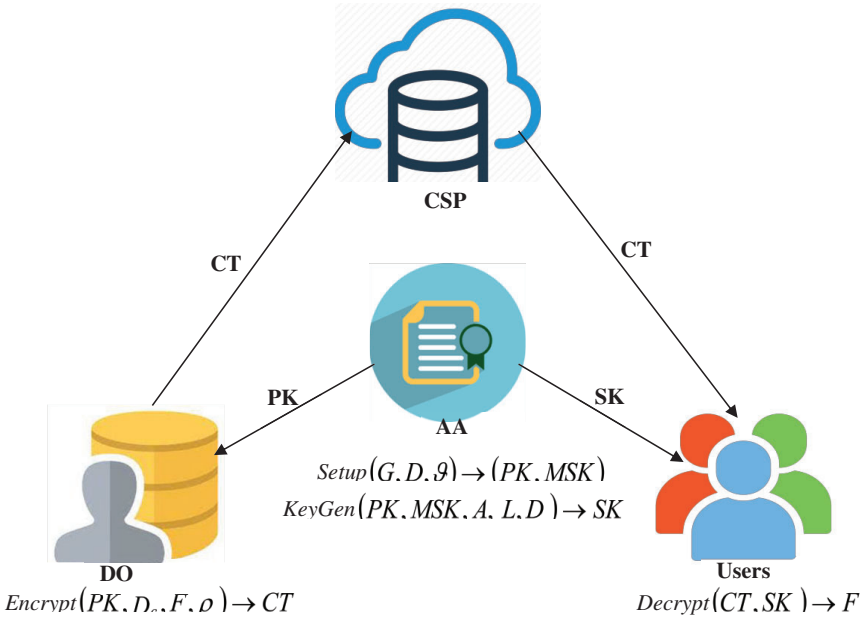

Figure 2: System model

CSP: It is used to store the data from the Dos. The users can retrieve the data from the CSP depend on their requests. However, it is not completely trustworthy.

DO: DOs are known as the resource-constrained entities. They have data files for their users. The data files of DOs are stored in CSP and they can be shared to users via the CSP. The DOs encrypt the data files as well as they can define the access policy. At final, they store the cipher-text (CT) to the CSP.

AA: It is the key centre which originates PK and SK. It assigns PK to the DO as well as is it assigns SK to the users. It provides different access rights to each users depend on their attributes.

Users: users are defined as finite resource entities. The major operation of users is to access the encrypted data from the CSP. If the users' attributes satisfies the encrypted access policy, the can decrypt the data.

3.3 Preliminary

Let 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., $e: G_0 \times G_0 \rightarrow G_1$. The significant properties of bilinear map are described as below:

1. *Bilinearity*: $e(x^a,\ y^b) = e(x,\ y)^{ab}, \qquad x,\ y \in G_0 \ \ and\ a,\ b \in Z_p$
2. *Non-degeneracy*: $e(g,\ g) \neq 1$
3. *Ability for computation*: $e(g_1,\ g_2)$ is able to compute for all $g_1,\ g_2 \in G_0$

3.4 Access Policy Structure

Consider $U_1,\ \ldots,\ U_n$ which denotes the collection of user's attributes. A collection $C \subseteq 2\{U_1, \ldots, U_n\}$ is called as 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, structure of access is converted into a Boolean function.

3.5 Dynamic Location Attribute in CP-ABE

Due to the mobility of the users, new private key can be generated often. By considering this issue, location attributes are included in the proposed CP-ABE. Namely, along with the attributes set, location information of the user also defined in the access policy. To include the location attributes into CP-ABE, location information has to be encoded. So, Geohash framework is used to encode the longitude and latitude of the user's position into another format.

Using Geohash framework, short string is attained from the GPS coordinates. With the alphabet having 26 letters (a-z) and 6 digits from 2–7, Geohashes are encoded in base 32. The locations are specified using internal bits. The Geohash of the location (latitude, longitude) for example (52.51, 13.32) with precision having 6 characters is given as *u336xp*. Each character represents 5 bits. The Geohash represents an area within the shape of rectangle or square.

Encoding function in Geohash: An instance of encoding of location (latitude = 52.51, longitude = 13.32) is explained as follows. Tab. 1 illustrates the encoding of latitude into Geohash. For latitude, the minimum and maximum range is denoted as −90 and 90. As illustrated in the table, if the latitude is greater than the mid range, bit value is set to 1. Else it set to 0. For each iteration or bit position, the range of latitude is adjusted. Namely, the values below the mid range are discarded if the bit value is set to 1 as well as the values greater than the mid range are discarded if the bit value is set to 0. The selected range is highlighted with green colour in Tab. 1. Then, in the next iteration, the position of values discarded in the previous iteration is replaced with the green highlighted mid range. The same encoding operation is followed for longitude. But the difference is the minimum and maximum range of longitude is denoted as −180 and 180.

After attaining the bit string of latitude and longitude, these bits are combined to form the resulting string. Namely, the 0^{th} and remaining bits are in the even position are considered as longitude. As well as the 1th and remaining bits are in the odd position are considered as longitude. At final, by mapping the each 5 bits from the resulting string to the base 32 (26 letters (a-z) and 6 digits from 2–7), Geohash of (latitude = 52.51, longitude = 13.32) is attained as *u336xp*.

Table 1: Encoding of latitude into Geohash

Bit position	Lat_Max	Lat_Mid	Lat_Min	Bit result
0	90	0	−90	1
1	90	45	0	1
2	90	67.5	45	0
3	67.5	56.25	45	0
4	56.25	50.625	45	1
5	56.25	53.4375	45	0
6	53.4375	49.21875	45	1
.....	.....	.....	.....	.....

In CP-ABE, the location information can be included into the policy in the form of following syntax:

<attribute name>:<latitude>:<longitude>:<precision>

In this format, the last part denotes the precision with the number of bits.

3.6 A New Time Tree in CP-ABE

The present study presents a hierarchical identity-based encryption scheme with a new time tree (NTT) approach to prevent data leakage caused by unrestricted access by authorized users. In the proposed NTT, the private key with the valid access time period and encrypted data with decrypt time period are defined in this scheme. Fig. 3 illustrates the structure of NTT. In the structure, root node is set as empty. Rest of the nodes in the NTT denotes time period. Namely, year is denoted in the first layer, the month is denoted in the second layer and the day is denoted in the third layer. The proposed NTT can use additional layers like hours, minutes, seconds and so on. In this work, the layer is described to the day. A data can be encrypted by the DO at a particular time period, which can be day, one month or one year. As well as, the private key of the user can be valid for the time period of a day, one month or one year. In NTT, the DO sets the decrypt time period D_c. The user can decrypt the data when his/her valid time period D matches the time period described by the DO. That is,

$D_c \subseteq D \, or \, D_c = D$

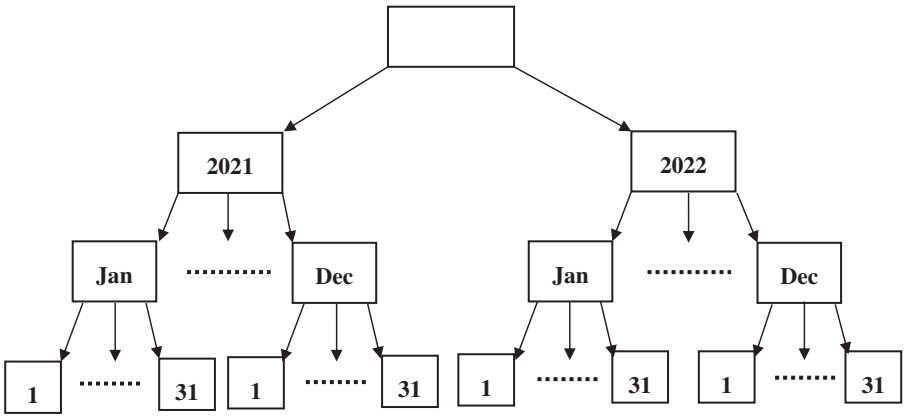

Figure 3: The structure of NTT

For instance, in NTT, April 30, 2022 is considered as the valid time period defined in the private key of the user. As well as, April 2022 is defined as the decrypt time period by the DO. This is not covers complete time period, namely $D_c \not\subset D$. So, the user is not permitted to decrypt the data. If April 2022 is considered as the valid time period of private key and April 30, 2022 is considered as the decrypt time period, this covers complete time period i.e., $D_c \subseteq D$. So, the user can decrypt the data. If the NTT sets the DO's encryption time for a specific month, then the user who has the private key for the specific month can decrypt the CT. An encrypted key can be obtained for a specific day of a particular month or year based on the CT describing that day. For better understanding of description, the depth of the NTT is denoted as D, the child node of each node is denoted as η. Time period can be described as string $s = (s_{\chi_1}, s_{\chi_2}, \ldots, s_{\chi_t})$ of a η elements, here $\chi_1 < D$.

3.7 Multi-Authorization

AA faces difficulty in the number of users accessing the data over a period of time. To solve this problem, single AA is extended into multiple AA. With the support of KeyGen and Delegate functions, the single AA is extended into multiple AA. In this approach, the set of attributes of each user is categorized into subset of attributes A'. For each set of A', SK' is generated. Besides, each AA performs the authorization service individually, i.e., $AA_{ij} \rightarrow A_{ij}$ and $|A_{ij}| \geq |A_{\max}|$. Here, AA_{ij} denotes the AA at ithlevel and jthposition and A_{ij} denotes the jth attribute in ithattribute set. Fig. 4 illustrates the hierarchical structure of multi-authorization. The function of multi-authorization is described as below,

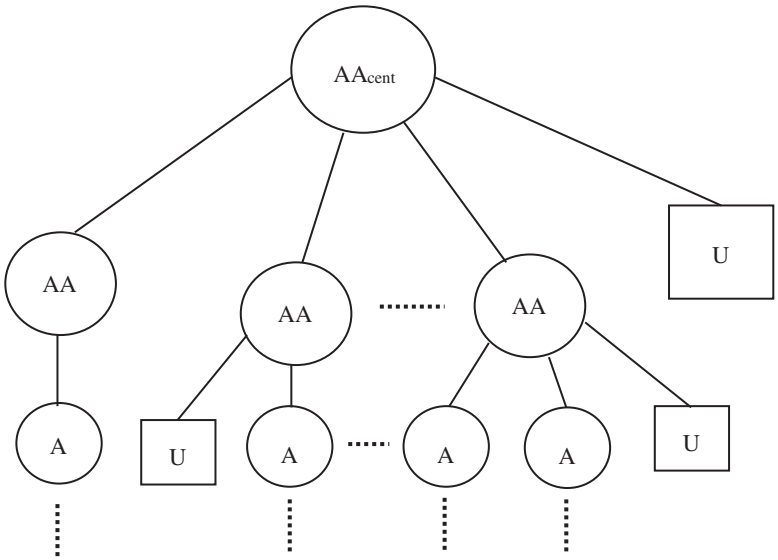

Figure 4: The hierarchical structure of multi-authorization

Case 1: Distribution of SK for secondary $AA_{1(n+1)} \rightarrow A_{1(n+1)}$ from $AA_{center} \rightarrow G$

As depicted in the Fig. 4, AA_{center} contains n number children nodes that are denoted as $AA_{11}, AA_{12}, \ldots\ldots, AA_{1n}$ as well as the corresponding sets of attributes are denoted as $A_{11}, A_{12}, \ldots\ldots, A_{1n}$. If $A_{1(n+1)}$ wants to join to the $AA_{1(n+1)}$, it should satisfy the $|A_{1(n+1)}| \geq |A_{\max}|$ and $A_{11} \cup A_{12} \cup \ldots\ldots\ldots \cup A_{1n} \cup A_{1(n+1)} \subseteq G$.

***Stage 1*:** Legitimacy of $AA_{1(n+1)}$ is to be verified at first.

***Stage 2*:** $MSK_{1(n+1)} = KeyGen(MSK, A_{1(n+1)})$

***Stage 3*:** $MSK_{1(n+1)} = (A_{1(n+1)}, SK)$

***Stage 4*:** AA_{center} forwards the $MSK_{1(n+1)}$ to $AA_{1(n+1)}$

Case 2: Distribution of SK' for ordinary $AA_{i(n+1)} \rightarrow A_{i(n+1)}$ from $AA_{(i-1)j} \neq AA_{center} \rightarrow G$

***Stage 1*:** Legitimacy of $AA_{i(n+1)}$ is to be verified at first.

***Stage 2*:** $MSK_{i(n+1)} = Delegate(MSK_{(i-1)j},\ A_{i(n+1)})$

***Stage 3*:** $MSK_{i(n+1)} = (A_{i(n+1)},\ SK')$

***Stage 4*:** $AA_{(i-1)j}$ forwards $MSK_{i(n+1)}$ to $AA_{i(n+1)}$

3.8 Operations of the Proposed CP-ABE

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

Setup $(G,\ D,\ \vartheta) \rightarrow (PK,\ MSK)$: The AA executes the setup of this algorithm. In this setup, the global attribute *G,* the depth of NTT *D* and security parameter ϑ are considered as input parameters. By using these input parameters, the output parameters PK and master SK (MSK) can be attained.

Algorithm 1: System setup

Input: $G,\ D,\ \vartheta$

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 period as a string s of a η elements.
4. Choose randomly $h_1,\ \ldots\ldots,\ h_G \in G_0$ and $R_1,\ \ldots\ldots,\ R_D \in G_0$
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. $PK \rightarrow \left\{G_0,\ G_1,\ g,\ g^{\beta},\ g^{\frac{1}{\beta}},\ e(g,\ g)^{\alpha},\ h_1,\ \ldots\ldots,\ h_G,\ R_1,\ \ldots\ldots,\ R_D\right\}$
9. **Return** PK, MSK

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

Algorithm 2: Key Generation

Input: $PK,\ MSK,\ A,\ L,\ D$

Output: SK

1. Consider T $\rightarrow$ minimum cover set of D.
2. T has multiple time interval represented as $s = (s_{\chi_1},\ s_{\chi_2},\ \ldots,\ s_{\chi_t})$ of a η elements, here $\chi_1 < D$
3. Choose randomly $r,\ r_j \in Z_p$
4. Estimate
   $E_0 \rightarrow g^r;\ E_1 \rightarrow g^{\frac{(\alpha+r)}{\beta}};\ \{E_{0,j} \rightarrow g^{r_j}\}_{j\in \mathrm{T}};\ \left\{E_{1,j} \rightarrow g^{\alpha} g^{\beta r} g^{\frac{(\alpha+r)}{\beta}} \left(R_0 \prod_{k=1}^{\chi_t} R_k^{kj}\right)\right\}_{j\in \mathrm{T}};\ \{K_v \rightarrow h_v^r\}_{v\in G}$
5. $SK \rightarrow \{E_0,\ E_1,\ \{E_{0,j},\ E_{1,j}\}_{j\in T};\ \{K_v\}_{v\in G}\}$
6. **Return** SK

Encrypt $(PK,\ D_c,\ F,\ \rho) \rightarrow CT$: The DO executes the encryption algorithm on the data to be stored in CSP. By using the input parameters such as PK, decrypt time period D_c, data F and access policy ρ, encryption algorithm is executed. At final, the DO attains encrypted data or cipher text (CT).

Algorithm 3: Encryption

Input: $PK,\ D_c,\ F,\ \rho$

Output: CT

1. Express decrypt time D_c as $s_c = (s_1,\ s_2,\ \ldots,\ s_t)$ for a η elements, here $t < D,\ t \in Z_p$ denotes the depth of D_c in NTT.
2. Generate access tree ρ for the dataset L_w, here L_w denotes set of leaf nodes and ϑ_w denotes the root of ρ.
3. Choose randomly a vector $q = (m,\ l_2,\ \ldots.,\ l_n) \in Z_p^n$
4. Estimate $\gamma_i = Q_i \bullet q$ for $i = 1,\ \ldots.,\ l$
5. Calculate $C_0 \rightarrow Fe(g,\ g)^{\alpha m};\ C_1 \rightarrow g^m;\ C_2 \rightarrow \left(R_0 \prod_{k=1}^{t} R_k^{kj}\right);\ \{C'_i \rightarrow g^{\beta\gamma_i} h_{\rho(i)}^{-m}\}_{i=1,\ldots.,l}$
6. $CT \rightarrow \{C_0,\ C_1,\ C_2,\ \{C'_i\}_{i=1,\ldots.,l}\}$
7. $C_F \rightarrow \{CT,\ \rho\}$
8. Return C_F

Decrypt $(CT,\ SK) \rightarrow F$: The user executes the decryption algorithm. In this phase, the user decrypts the CT using SK. Initially, the valid access time D of the user should matches the D_c in CT as well as attribute set A and location attribute L of the user should matches the defined access policy ρ to decrypt the search data F. Else decryption algorithm will be terminated.

Algorithm 4: Decryption

Input: $CT,\ SK,\ C_F$

Output: F

1. Attain ρ and L_w from C_F.
2. Check the data in L_w and attain the root ϑ'_w
3. Compare ϑ'_w and ϑ_w
4. **If** the attribute set A of user can't satisfy the access policy
   Program is terminated
5. **Else** compute $\phi \rightarrow \prod_i (e(E_0,\ C'_i) \cdot e(K_{\rho(i)},\ C_1))^{\tau_i},\quad \tau_i \in Z_p$
6. **End**
7. **If** $s_c = (s_1,\ s_2,\ \ldots,\ s_t) \in \mathrm{T}$
8. **Then** compute $Dec = \dfrac{C_0 \cdot \phi \cdot e(C_1,\ E_1) \cdot e(C_2,\ E_{0,s_c})}{e(C_1,\ E_{1,s_c})}$
9. **End**
10. $Dec(L_w) \rightarrow F$
11. **Return** F

Delegate $(SK, A') \rightarrow SK'$: For the set of attributes A, 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' satisfied $A' \subseteq A$. This function is used for the extension of AA.

4 Results and Discussion

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

Fig. 5 illustrates the encryption time of different algorithms for varying data size. As illustrated in the graph, encryption time of the suggested CP-ABE is reduced to 20% and 36% than that of existing CP-ABE and ABE respectively at 2000KB of data size. At 10000KB of data size, compared to existing CP-ABE and ABE, the encryption time of the suggested CP-ABE is minimized to 19% and 26% respectively.

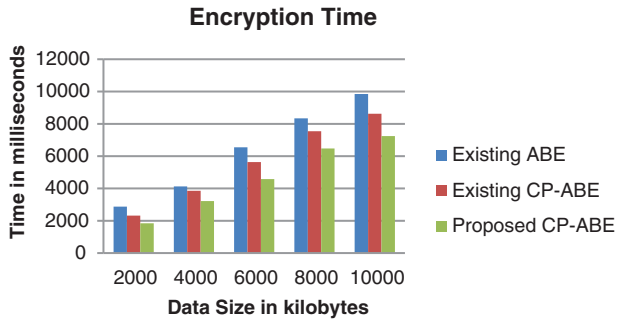

Figure 5: Encryption time of different algorithms

Decryption time of different algorithms is illustrated in Fig. 6. Decryption time increases when the size of data is increased as expressed in the figure. The overall decryption time of the suggested CP-ABE is reduced to 16% and 26% than that of existing CP-ABE and ABE respectively.

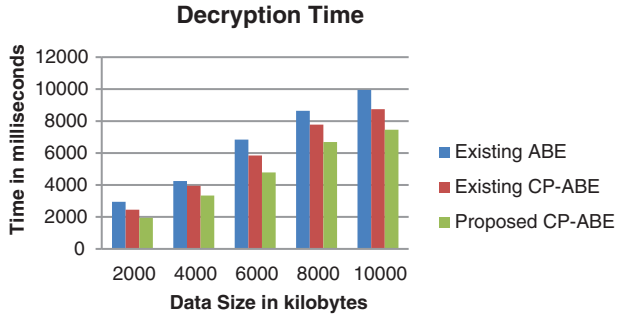

Figure 6: Decryption time of different algorithms

Fig. 7 depicts the key generation time of different algorithms. As depicted in the figure, the proposed CP-ABE achieved key generation within 1034 ms while conventional CP-ABE and ABE achieved key generation within 1214 ms and 1574 ms respectively.

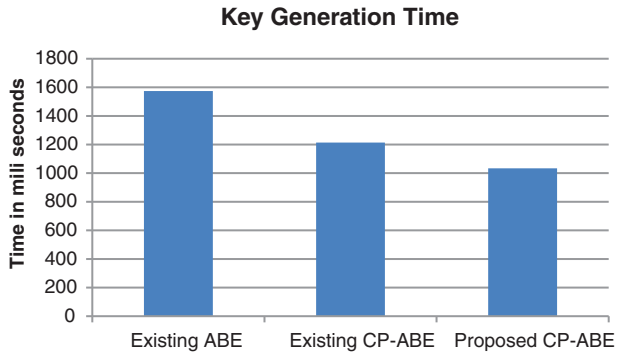

Figure 7: Key generation time of different algorithms

Security level of the different algorithm is analysed in Fig. 8. Although existing CP-ABE increases the security level to 93% than the existing ABE, time and location attributes to be included to it for increasing the security level further. Thus, security level of the system is increased to 96% by the proposed CP-ABE.

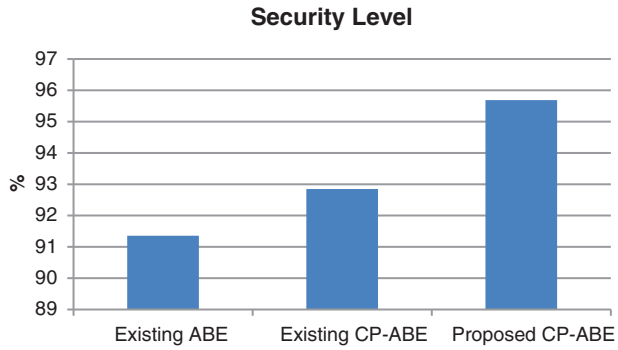

Figure 8: Security level of different algorithms

Fig. 9 illustrates the comparative analysis of memory usages of encryption for varying data size. As depicted in the figure, memory usage of the proposed CP-ABE in encryption is reduced to 8% and 13% than that of existing CP-ABE and ABE in encryption respectively.

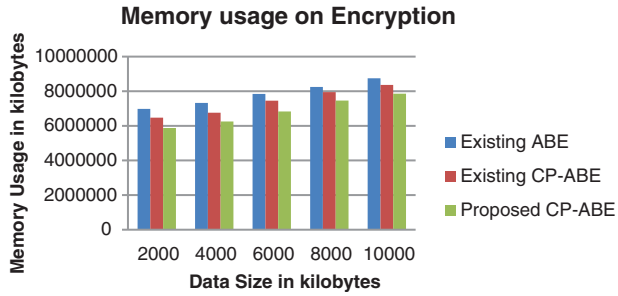

Figure 9: Memory usage on encryption of different algorithms

The comparative analysis of memory usage of decryption is illustrated in Fig. 10. Compared to existing CP-ABE and ABE, memory usage of the proposed CP-ABE in decryption is reduced to 7% and 12% respectively.

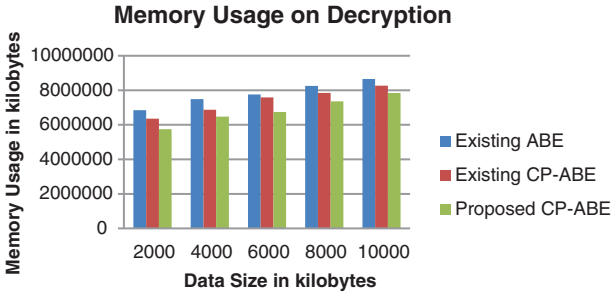

Figure 10: Memory usage on decryption of different algorithms

5 Conclusion

To solve the security issues of CP-ABE due to the mobility of users, dynamic attributes such as time and location have been used in CP-ABE with multi-authorization. The location information, namely latitude and longitude positions of the users have been encoded using Geohash. By presenting NTT structure, decrypt time and access time of users have been defined. It leads to revoke the unauthorized users. Besides, the efficiency of key generation has been improved by extending the single AA to multiple AA. The performance of the proposed CP-ABE has been analysed in terms of security level, memory usage, encryption time and decryption time. Simulation results depicted that the proposed CP-ABE achieved 96% of security level than the existing CP-ABE and ABE. In future, we focus to private the access policy as it is public using an efficient hashing algorithm.

Acknowledgement: The authors 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. P. Lalitha, M. Y. Sagar, S. Sharanappa, S. Hanji and R. Swarup, "Data security in cloud," in *2017 Int. Conf. on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*, Chennai, India, pp. 3604–3608, 2017.

[2] F. Kunz and Z. A. Mann, "Finding risk patterns in cloud system models," in *2019 IEEE 12th Int. Conf. on Cloud Computing (CLOUD)*, Milan, Italy, IEEE, pp. 251–255, 2019.

[3] A. Shaikh and J. Gadge, "Framework for security of shared data in cloud environment," in *2016 Int. Conf. on Computing Communication Control and Automation (ICCUBEA)*, IEEE, Pune, India, pp. 1–6, 2016.

[4] S. Zhou, G. Chen, G. Huang, J. Shi and T. Kong, "Research on multi-authority CP-ABE access control model in multicloud," *China Communications*, vol. 17, no. 8, pp. 220–233, 2020.

[5] T. Lee, H. S. Moon and J. Jang, "Data encryption method using CP-ABE with symmetric key algorithm in blockchain network," in *2021 Int. Conf. on Information and Communication Technology Convergence (ICTC)*, IEEE, Jeju Island, Korea, pp. 1371–1373, 2021.

[6] V. Reshma, S. J. Gladwin and C. Thiruvenkatesan, "Pairing-free CP-ABE based cryptography combined with steganography for multimedia applications," in *2019 Int. Conf. on Communication and Signal Processing (ICCSP)*, IEEE, Chennai, India, pp. 0501–0505, 2019.

[7] L. Touati and Y. Challal, "Efficient CP-ABE attribute/key management for iot applications," in *2015 IEEE Int. Conf. on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, IEEE, Liverpool, UK, pp. 343–350, 2015.

[8] D. V. K. Vengala, D. Kavitha and A. P. Kumar, "Secure data transmission on a distributed cloud server with the help of HMCA and data encryption using optimized CP-ABE-ECC," *Cluster Computing*, vol. 23, no. 3, pp. 1683–1696, 2020.

[9] Y. Zhao, M. Ren, S. Jiang, G. Zhu and H. Xiong, "An efficient and revocable storage CP-ABE scheme in the cloud computing," *Computing*, vol. 101, no. 8, pp. 1041–1065, 2019.

[10] S. Xue and C. Ren, "Security protection of system sharing data with improved CP-ABE encryption algorithm under cloud computing environment," *Automatic Control and Computer Sciences*, vol. 53, no. 4, pp. 342–350, 2019.

[11] P. K. Premkamal, S. K. Pasupuleti and P. J. A. Alphonse, "A new verifiable outsourced ciphertext-policy attribute based encryption for big data privacy and access control in cloud," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 7, pp. 2693–2707, 2019.

[12] H. He, L. Zheng, P. Li, L. Deng, L. Huang *et al.,* "An efficient attribute-based hierarchical data access control scheme in cloud computing," *Human-centric Computing and Information Sciences*, vol. 10, no. 1, pp. 1–19, 2020.

[13] S. Ding, C. Li and H. Li, "A novel efficient pairing-free CP-ABE based on elliptic curve cryptography for IoT," *IEEE Access*, vol. 6, pp. 27336–27345, 2016.

[14] R. Mythili, R. Venkataraman and T. Sai Raj, "An attribute-based lightweight cloud data access control using hypergraph structure," *The Journal of Supercomputing*, vol. 76, no. 8, pp. 6040–6064, 2020.