Comparative Attribute Access Control Scheme Based on Spatio-temporal Constraints in Cloud

Junling Zhang, Ze Wang, Ping Zhao, Minghua Gao, and Shimin Sun
(Corresponding author: Ze Wang)

School of Computer Science and Technology, Tiangong University
399 Binshui W Rd, Xiqing, Tianjin, China
Email: wangze@tiangong.edu.cn

(Received July 15, 2021; Revised and Accepted Mar. 2, 2022; First Online Apr. 9, 2022)

Abstract

Cloud computing provides an extensible, high-performance solution by entrusting computational tasks and storage to clouds, effectively addressing resource constraints in data storage, data sharing, and computing for users. Attribute-based encryption (ABE) is considered the most potent encryption primitive for achieving fine-grained access control and solves the problem of one-to-many encrypted data sharing. However, as applications evolve, the ABE scheme may not handle particular scenarios such as access policies related to the user's time and space range. In addition, end-users should protect their identity and the associated privacy from attacks by malicious authorized institutions and providers. Attribute encryption scheme based on ciphertext policy (CP-ABE) encrypts ciphertext according to user attribute set, which uses policies and constraints defined by the data owner to provide safe and reliable fine-grained access control. To solve the Spatio-temporal constraints more effectively, this paper introduces the comparison relationship in the comparative attribute encryption (CBE) scheme into the process of attribute-based encryption and realizes that the two attributes of time and location satisfy various constraints on the integer set. Thus, data users can flexibly access data at a multi-dimensional level. Furthermore, our solution is proven secure and efficient through security and performance analysis.

Keywords: Comparison Attribute Encryption; Dynamic Access Control; Spatio-temporal Constraints

1 Introduction

Cloud storage provides us with a new data information storage model, which enables economies of scale and elastic scaling, reducing operating cost and avoiding waste of resources. Therefore, many enterprises are migrating their business to one or more cloud platforms. However, this model of cloud storage poses entirely new challenges to privacy and security, because data is no longer stored in users' trusted domains, but stored in remote cloud servers. In addition, cloud service providers may try to learn user's sensitive data, resulting in privacy leakage. Therefore, achieving access control while ensuring data security is a great challenge.

To implement secure data access control on untrusted cloud servers, traditional solutions encrypt data before storing it in cloud servers, but the encryption results in higher key management overhead and increases the complexity of the system. Attribute-based encryption (ABE) is the most promising solution to private data access control through reasonable configuration of sharing policies. Since [19] was proposed in 2005, the ABE has attracted wide attention from the government, enterprises and academia. The ABE is an extension of identity-based encryption, which implements broadcast encryption for a group of users and provides fine-grained encrypted access control for data down to the attribute level. [10] proposes Key-Policy ABE (KP-ABE), which embeds policies into user keys and embeds attributes into ciphertexts. The design of the KP-ABE has its application in scenarios such as paid video sites, log encryption management, etc. Different from the KP-ABE, [3] proposes Ciphertext-Policy ABE (CP-ABE), which embeds policies into ciphertexts and embeds attributes into user keys. The CP-ABE can be used in many applications including cloud environments, hospitals, government and smart factories.

Due to the high flexibility and scalability of cloud computing, many enterprises and individuals have used cloud servers to store and calculate their data. However, some occasions require high confidentiality, where users' access rights depend not only on their attributes, but also on dynamic factors, such as change of time and location. Taking a PHR system as an example, patients can develop access policies to encrypt their personal medical records and upload them to the cloud server, allowing relevant medical staff to view the patients' medical records under certain conditions and make diagnosis. Assume that in the PHR system only doctors and pharmacists are allowed to view the medical records and examine patients' health status, where doctors can visit the documents at any time and any place, but pharmacists can only visit the documents in the office building of their hospital during working hours (8:00 a.m.-18:00 p.m.). In this case, the fine-grained access control is applied, which considers access time, location and identity attributes, and achieves rapid retrieval and sharing of medical data.

The above case needs to consider both time and space, but the traditional CP-ABE scheme rarely combines dynamic changes of time and space to solve access control problems in cloud storage. In 2014, Androulaki *et al.* [1] described in detail the framework of the LoTAC, which enables spatio-temporal access control of cloud stored data by integrating the cloud provider's operations and infrastructure. According to literature [20], in 2014 Shao *et al.* provided location privacy protection by encrypting the anonymity attribute of ciphertext policy to ensure the confidentiality of service data and access policies for locations. According to literature [22,29,30], in 2013-2014 Zhu *et al.* constructed an encryption scheme based on specific temporal predicates by comparing secure integers. There are some common problems in the existing schemes:

1) Computational cost is fairly high for data owners to enhance access policies and encrypt data for each user;

2) It is easy for malicious users to publish false information and illegally access data and services;

3) When rough locations are not dense enough, it causes the privacy leakage of mobile users' locations.

A malicious location service provider (LSP) may retain the current location, and provide users with previous traveling records and other information. The LSP also can monitor the user's life trajectory, preferences, health status and other aspects of his private life based on the user's spatial and temporal information. Or it may simply use the user ' s spatio-temporal information to identify users. In [5], the singleness test experiment shows that four points with specific time and location are sufficient to correctly identify an individual user with a probability of 95%. The combination of location and time reveals individual characteristics of a user, which is known as the user's standard identifier. Therefore, guaranteeing anonymity of users and protecting their identity requires protection of their spatial and temporal information and their identity. In [17], a new access control scheme based on temporal and spatial constraints is proposed. The TSC-ABAC scheme uses multi-dimensional range derivation function (MRDF) to compare tand uses the token generation algorithm to determine the location. This is the first scheme to handle time and location at the same time, bbut the scheme cannot effectively support continuous location range constraint information in the location dimension.

Therefore, this paper re-designs the ABE encryption scheme based on bilinear mapping to solve the above problems, and adds the two attributes of time and location into the scheme. A user key is associated with static properties, time range and location range. If the user's attributes satisfies given requirements and the temporal and spatial ranges match the access policy, the trapdoor can be released and the ciphertext can be decrypted successfully. The contributions of this paper are as follows:

1) We propose the first CP-ABE scheme that can effectively deal with constraints of both time range and location range;

2) We apply the proxy re-encryption scheme in the proposed scheme, and associate the temporal and spatial attributes owned by the user with the current access time of the user to make the scheme more flexible and efficient;

3) Security analysis and theoretical performance analysis show that the proposed solution is secure and effective.

2 Related Work

At present, there are several ABE schemes to deal with the temporal and spatial factor. However, most schemes consider only one of the factors. Scheme in literature [1] combines both aspects, but ignores the fine-grained access control for users. To overcome the drawbacks of the above schemes, this paper designs a data access control scheme that considers both temporal and spatial ranges. In this paper, by integrating the constraints of user's spatio-temporal attributes into access control, a complete and effective scheme is designed, and the comparative attribute-based encryption and decryption is adopted to define user access rights, thus achieving more flexible fine-grained access control. At present, only a few access control schemes consider dynamic and static attributes, and the computational cost is huge. Considering location-based access control from the perspective of spatio-temporal constraints, and combining common part shared by access control mechanism and location verification, we design the access control scheme that can resist spatio-temporal attacks.

2.1 Encryption Schemes Associated with Time Attributes

In the development process of the ABE scheme, some work has studied the constraint of time. For example, in literature [21], Bethencourt *et al.* proposed a scheme by utilizing the access policy tree which assigns 0 / 1 to the branches of the access policy tree to realize the comparison of integers. However, actual application process faces the problem of low efficiency. Therefore, Hong [11] *et al.* proposed a concept of timed release to encrypt the access policy tree of the CP-ABE, which cannot accurately know user's access time, so it cannot work well when the user make access within a certain time. Through the forward/backward derivation function, Zhu *et al.* [29] first proposed a comparative attribute-based encryption scheme, which performs well in comparing the time range. In literature [28], Zhu *et al.* discussed how to use a mechanism from the proxy re-encryption scheme to control the constraint of time range, but they only presented a general idea without specific solutions to the above problems. In literature [27], Yang *et al.* proposed an access control scheme taking time into consideration, which slices time, refreshes the key in initialization of each time slot and sends that key to the user who meets the policy requirements. However, this scheme needs to update the key and publish it at each time slot, which leads to increased encryption and decryption overhead and reduced efficiency. In literature [23], Wang *et al.* proposed a scheme that handles the constraint of time range with a multidimensional range derivation function, but the range of attributes handled is small and the scheme is not suitable for large scale generalization.In literature [6,18], TRE technology is combined with encryption scheme, and a CP-ABE scheme based on time-release encryption is proposed. TRE relies on a time server to publish trap gates at a specified time, and only when the receiver gets the trap gates can it be decrypted.However, this scheme does not support dynamic change of time range and has poor flexibility.

2.2 Encryption Schemes Associated with Location Attributes

Some ABE schemes are combined with the LBS, as mentioned in literature [14, 25]. However, these schemes only consider the location of users, and use privacy protection technology to achieve access control. Actually, dynamic location can be treated as a common attribute of the ABE scheme. In literature [7], Denisow *et al.* encrypt user's location by the Geohash algorithm, and integrate the location into the ABE scheme. In literature [26], Xue *et al.* improved the CP-ABE scheme by applying the trapdoor mechanism to the access control scheme. In literature [8], the scheme proposed by Ghafghazi *et al.* integrates broadcast encryption with the CP-ABE to handle location attributes.In literature [16], in order to protect user location privacy, combined with OT and CP-ABE schemes, a privacy-protecting LBS query scheme is proposed to protect the privacy of LBS suppliers and vehicles.

3 Preliminaries and Defintions

3.1 Composite Order Bilinear Map

Let $p, q, p^{'}, q^{'}, s_1, s_2$ be large prime numbers, $N = \mathrm{pq}$ is a public RSA model,$s = s_1 s_2$, $n^{'} = p^{'} q^{'}$ and $n = sn^{'}$ be secret,G and G_T be two cyclic bilinear groups of composite order $n = sn^{'}$, α,β be two random exponents in Z, and $e : G \times G \to G_T$ be a bilinear map with the following properties:

1) Bilinearity: for any $\forall g_1, g_2 \in G$ and $\forall a, b \in Z$,$e\left({g_1}^a, {g_2}^b\right) = e\left(g_1, g_2\right)^{ab}$;

2) Non-degeneracy: g_1 and g_2 are the generators of group, $e\left(g_1, g_2\right) \neq 1$;

3) Computability: for $g_1, g_2 \in G$,there exists a valid algorithm to compute $e\left(g_1, g_2\right)$.

3.2 Trapdoor Structure

The access policy tree consists of trapdoors and nodes associated with time and location ranges. Location and time trapdoors can be embedded in any node of the access policy tree, and the access rights of certain users are restricted by the trapdoor TD. We define that a trapdoor has two states, exposed and not exposed.

Not exposed: a user is not allowed to pass through the trapdoor in order to access the corresponding secret.

Exposed: a user can pass through the trapdoor to access the corresponding secret and the trapdoor is exposed.

A user needs to have certain attributes required by the access policy and initiate the access within corresponding temporal and spatial ranges to release the trap door.

With the trapdoor independent of user's set of attributes, the user does not have a private key associated with the time and location. As a result, the trapdoor decreases the workload of revoking and re-distributing private keys. The scheme in this paper allows different access policies to set trapdoors and a ciphertext can be associated with different trapdoors of different spatio-temporal constraints. Therefore, the spatio-temporal information can be flexibly combined with other user attributes, and users have to meet the access policy and release the trapdoor to access the secret data.

3.3 Access Policy Tree

In the process of data encryption, traditional ABE schemes perform access control through the structure of an access policy tree. We optimized the tree by embedding spatio-temporal constraints into the structure, adopting a multi-dimensional distance derivation function, and using trapdoors to help determine the legitimacy of users. The structure of the access policy tree is shown in Figure 1 below:

This paper implements the fine-grained access requirements through an access policy tree, where each leaf node represents a attribute At owned by the user. Non-leaf nodes represent logic gates (AND,OR) and trapdoors (Threshold).num_x represents the number of children of non-leaf node x, and k_x denotes the threshold value. $\mathrm{parent}\left(x\right)$ denotes the parent node of node x. $\mathrm{attr}\left(x\right)$ indicates that the leaf node x is associated with the attribute.

In Figure 1, we consider embedding the spatio-temporal constraints into the non-leaf nodes of the access policy tree,and define $\mathrm{TD}^x_{\{t_a,t_b\}}$ to represent the constraint of time range associated with node x and $\mathrm{LD}^x_{\{l_a,l_b\}}$

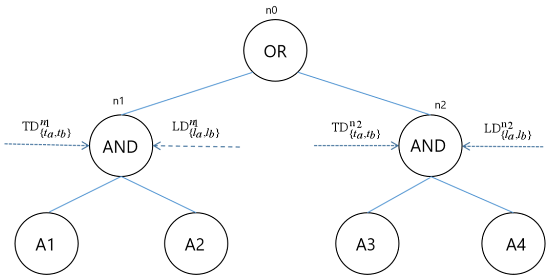
n0
OR
n1
$\mathrm{TD}^{n1}_{\{t_a,t_b\}}$
AND
$\mathrm{LD}^{n1}_{\{l_a,l_b\}}$
n2
$\mathrm{TD}^{n2}_{\{t_a,t_b\}}$
AND
$\mathrm{LD}^{n2}_{\{l_a,l_b\}}$
A1
A2
A3
A4

Figure 1: Access policy tree structure

to represent the constraint of location range associated with node x. In the access policy tree as shown above, the sub-access policy applied to node n1 needs to satisfy $A_1 \wedge A_2$,and the sub-access policy applied to node n2 needs to satisfy $A_3 \wedge A_4$. The user needs to satisfy the requirement of $(A_1 \wedge A_2) \vee (A_3 \wedge A_4)$, and the access time and location should be within the range of $[t_a, t_b], [l_a, l_b]$ in order to successfully access resources from the cloud.

3.4 Introduction of the Multidimensional Range Derivation Function

The MRDF utilizes the "one-way" property to represent the total ordering relation of integers. We choose the MDRF to select the lower-bound and upper-bound integer values $(l_{i,j}, l_{i,k})$,and $\psi \to U$ is a cryptographic mapping regarding the user's U-preserving order, mapping each attribute to a value $v\{l_{i,j}, l_{i,k}\}_{A_i \in A}$ that reflects a cryptographic bound.

We define this mapping function $\psi(\cdot)$ as follows:

$$\begin{aligned} \mathrm{v}\{l_{i,j}, l_{i,k}\}_{\mathrm{A}_i \in \mathrm{A}} &\leftarrow \psi\left(\{l_{i,j}, l_{i,k}\}_{\mathrm{A}_i \in \mathrm{A}}\Big\}\right) \\ &= \left(\varphi^{\prod_{\mathrm{A}_i \in \mathrm{A}} \lambda_1^{l_{i,j}} \mu_i^{z-l_{i,k}}}\right) \in G_{n'}. \end{aligned}$$

Given a function F: $V \to v$ based on a set, it is called the multi-dimensional range derivation function when it satisfies the requirements below:

1) Easy to compute: the function F can be computed in the PPT algorithm, if $l_{i,j} \leq l'_{i,j}$,$l_{i,k} \geq l'_{i,k}$,we have: $\forall A_i \in A$,

$$\mathrm{v}\{l'_{i,j}, l'_{i,k}\} \leftarrow \mathrm{F}\{l_{i,j} \leq l'_{i,j}, l_{i,k} \geq l'_{i,j}\}(\mathrm{v}\{l_{i,j}, l_{i,k}\}).$$

2) Hard to invert: For an attribute $A_i \in A$, it is difficult for any PPT algorithm to derive $\mathrm{v}\left\{l'_{i,j}, l'_{i,k}\right\}$,if $l_{i,j} > l'_{i,j}$ or $l_{i,k} < l'_{i,k}$.

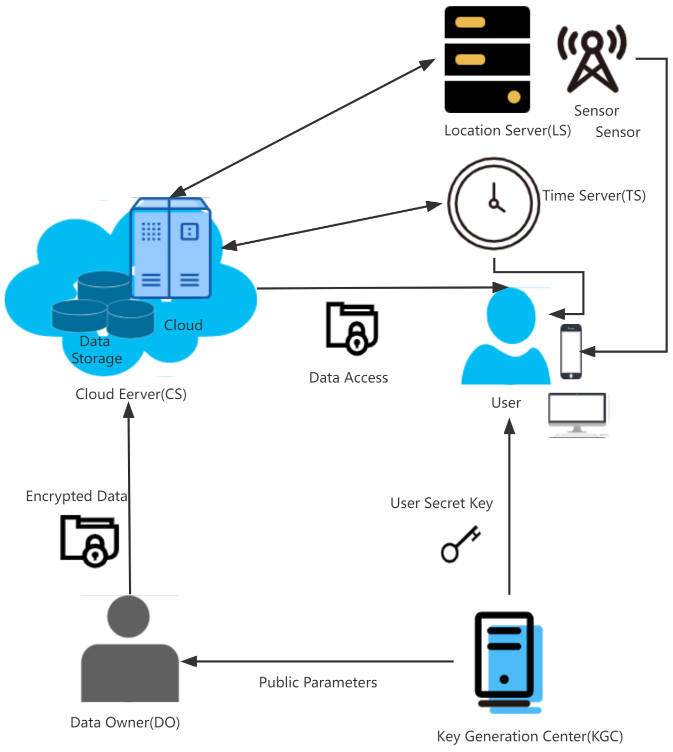

Figure 2: System model diagram

The function F(.) has the following form:

$$\begin{aligned} \mathrm{v}\{l'_{i,j}, l'_{i,k}\} &\leftarrow \mathrm{F}\{l_{i,j} \leq l'_{i,j}, l_{i,k} \geq l'_{i,j}\}(\mathrm{v}\{l_{i,j}, l_{i,k}\}) \\ &= (v\{l_{i,j}, l_{i,k}\}_{A_i \in A})^{\prod_{A_i \in A} \lambda_i^{l'_{i,j}-l_{i,j}} \mu_i^{l'_{i,k}-l_{i,k}}} \\ &= (\phi^{\prod_{A_i \in A} \lambda_i^{l_{i,j}} \mu_i^{z-l_{i,k}}})^{\prod_{A_i \in A} \lambda_i^{l'_{i,j}-l_{i,j}} \mu_i^{l'_{i,k}-l_{i,k}}} \\ &= \phi^{\prod_{A_i \in A} \lambda_i^{l'_{i,j}} \mu_i^{z-l'_{i,k}}} \in G_{n'}. \end{aligned}$$

4 TSC-CABE Scheme

4.1 System Model

Our system mainly contains six entities, Cloud Service Provider (CSP), Key Generation Center (KGC), Location Server (LS), Time Server (TS), Data Owner (DO), Data User (DU), as Figure 2.

Key Generation Center (KGC). The KGC initializes public parameters and distributes keys to users based on their attribute sets. In this system, it is assumed that the KGC is a fully trusted institution.

Cloud service provider (CSP). The cloud service provider stores data and provides access to data for users. Using its powerful computing power to provide data re-encryption services, the cloud service provider is considered to be honest and semi-trustworthy.

Data owner (DO). The data owner specifies the policy for accessing the ciphertext and associates a set of attributes with resources to be accessed. Assume that the data owner is honest and trustworthy.

Data user (DU). Each user has a unique identifier and the KGC issues keys to users based on their attributes. A user, if considered dishonest, wants to decrypt the data without being authorized. It is likely that unauthorized users collude together to obtain more information.

Time Server (TS). The time server is to provide safe and reliable time services including time synchronization. Let F_t represent the time format in the system.

Location Server (LS). The location servers are distributed in some specific areas, which can perform computational operations. For example, the location server for trapdoor decryption can find out the location of the user with the help of sensors. Let F_{loc} represent the location format in the system.

4.2 Framework of TSC-CABE Scheme

The TSC-CABE proposed in this paper mainly includes five phases: system initialization, key generation, encryption, re-encryption, and decryption. Below is introduced the implementation of each phase.

- System initialization $(\kappa) \quad \rightarrow \quad (PKP, MK)$

  The initialization algorithm is operated by the KGC, and the algorithm outputs the public key parameter PKP and the master key MK by entering a security parameter κ.

- Key generation $(PKP, MK, \mathrm{gid}, S_{gid}, T_{gid}, L_{gid}) \quad \rightarrow \quad SK_{gid}$

  Operated by the KGC, the key generation algorithm generates user's private key SK_{gid} by entering the public key parameter PKP, the master key MK, the user global identity gid, the set of attributes of the user, and the time and location constraints $T_{gid} = [t_a, t_b], L_{gid} = [l_a, l_b]$.All elements in $T_{\mathrm{gid}}, L_{\mathrm{gid}}$ are related by specific integers to guarantee the full order like $0 \leq t_a \leq ... \leq t_b$.

- Encryption $(m, \Gamma, PKP) \quad \rightarrow \quad (\mathrm{CT})$

  The data owner operates the encryption algorithm.The algorithm receives the message M, accesses the tree structure and the public key parameter PKP, and final outputs the ciphertext CT with time and location constraints embedded in the ciphertext as defined by the data owner.

- Re-encryption $(PKP, CT, t_c, l_c) \quad \rightarrow \quad (\mathrm{RC})$

  The re-encryption algorithm is run by the CSP. Input the public key PKP, ciphertext CT, current time t_c and current location l_c of user access into the algorithm and the re-encrypted ciphertext RC is output.

- Decryption $(PKP, RC, SK_{gid}) \quad \rightarrow \quad (\mathrm{m}/\perp)$

  The user operates the decryption algorithm by inputting the public key parameter PKP, the ciphertex RC, and the user's private key $\mathrm{SK_{gid}}$.If the attribute of time is within the current time range and the location is within the current location range, the decryption can be done successfully, otherwise the decryption shall fail.

5 Algorithms in the TSC-CABE Scheme

- Algorithm 1. System initialization algorithm

  First of all, the key generation center selects a bilinear mapping system $S_N = \{N = pq, G, G_T, e\}$ with the composite order as $n = s^{'} n^{'}$, and then selects G_s,$\mathrm{G}_{n'}$, two subgroups of G. What follows is that the KGC selects random generators $g \in G_s, \phi \in G_{n'}, \omega \in G$, and two random numbers $\lambda, \mu \in Z_{n'}^*$,where $e(g, \phi) = 1$ but $e(g, \omega) \neq 1$.Next, operate the three hash functions, $H_0, H_1 : \{0,1\}^* \rightarrow G_{S'}, H_2 : G_T \rightarrow Z_n^*$ .Select any two indices $\alpha, \beta \in Z_n^*$ and generate $h = \omega^{\beta}, \eta = g^{1/\beta}, \varsigma = e(g, \omega)^{\alpha}$ In the end, the public key is generated as follows:

$$\mathrm{MK} = \left(p, q, n^{'}, \alpha, \beta\right).$$

- Algorithm 2. key generation algorithm

  Each user has a set of attributes S_{gid}.The key generation center KGC selects $u_j \in Z_n^*$ and a random value $r_i \in Z_n^*$ for each attribute $i \in S_{gid}$, and then computes the corresponding attribute key as follows:

$$\mathrm{SK_{attr}} = \Bigg\{\mathrm{D} = \mathrm{g}^{\frac{\alpha+u_j}{\beta}} H_0(\mathrm{gid})^{\frac{u_j}{\beta}}, \mathrm{D}' = \omega^{\mathrm{u}_j}$$
$$\forall \mathrm{i} \in \mathrm{S_{gid}} : \mathrm{D_i} = (\mathrm{g} H_0(\mathrm{gid}))^{u_j} \mathrm{H_1(i)^{r_i}}, \mathrm{D_i'} = \omega^{\bar{r}_i}\Big)\Big\}.$$

To achieve access control under the constraint of time, we assume that the user is assigned the temporal and spatial access rights $[t_a, t_b], [l_a, l_b]$, where F_t, F_{loc} respectively represent the temporal and spatial formats in the system. $[t_a, t_b], [l_a, l_b]$ represent the boundary values of time and location in the system respectively, and all elements are discrete integers with total ordering. The KGC selects a $r_t \in Z_n^*$ for each temporal attribute and generates a time key as

$$\mathrm{DK}_{[t_a, t_b]} = \{D_t = (g H_0(gid)^{u_j}) \cdot H_1(F_t)^{r_t}, D_t' = \omega^{r_t}$$
$$D_t'' = \left(v_{\{t_a, t_b\}}\right)^{r_t} = \varphi^{r_t} \lambda^{ta} \mu^{z - t_b}\Big\}.$$

Also, the KGC select a $r_l \in Z_n^*$ for each spatial attribute and generate a location key as

$$\mathrm{DK}_{[l_a,l_b]} = \{D_l = (gH_0(gid)^{u_j}) \cdot H_1\left(F_l\right)^{r_l}, D_l' = \omega^{r_l}$$
$$D_l'' = \left(v_{\{l_a,l_b\}}\right)^{r_l} = \varphi^{r_l \lambda^{l_a} \mu^{z-l_b}} \Big\}.$$

Final the user key is generated as

$$\mathrm{SK_{gid}} = \left\{\mathrm{SK_{Attr}}, \mathrm{DK}_{[t_a,t_b]}, \mathrm{DK}_{[l_a,l_b]}\right\}.$$

- Algorithm 3. encryption algorithm

The data owner first encrypts the message by a symmetric key ek, and then encrypts ek according to an access policy defined by himself. Later the DO uploads the whole encrypted data to the cloud server. A ciphertext is generated by the algorithm below.

The algorithm starts by visiting the root node of the tree and generates a polynomial q_x for each node from the top down, and for each node x, set $d_x = k_x - 1$. Beginning with the root node, the algorithm selects a random number $s \in Z_n^*$ and sets $q_R^0 = s$.Each node x has two values q_x^0, q_x^1. If a time trapdoor or a location trapdoor is associated with node x, node x is associated with $t_x^0 \in Z_n^*$ and $l_x^0 \in Z_n^*$. For node x, the value of q_x^1 value is calculated as follows:

$$\begin{cases} q_{\mathrm{x}}^1 = q_x^0 - l_x^0 - t_x^0 & \text{x is associated with both} \\ q_{\mathrm{x}}^1 = q_x^0 - l_x^0 & \text{x is associated with location} \\ q_{\mathrm{x}}^1 = q_x^0 - t_x^0 & \text{x is associated with time} \\ q_{\mathrm{x}}^1 = q_x^0 & \text{There is other cases} \end{cases}$$

For a non-leaf node x, the polynomial q_x can be chosen arbitrarily, provided that $q_x\left(0\right) = q_x^1$ and $d_x = k_x - 1$ are satisfied. For any node x except the root node, $q_x^0 = q_{paraent(x)}\left(index\left(x\right)\right)$. Let χ be the set of leaf nodes in the access policy tree, γ represent the set of attributes associated with the time range $[t_a, t_b]$, and Z represent the set of attributes associated with the location range $[l_a, l_b]$. The ciphertext CT is as follows:

$$CT = \Big\{\mathrm{T}, \tilde{C} = \mathrm{Enc}(\kappa, m), C = \kappa e(g,\omega)^{\alpha s}, C' = h^s$$
$$\forall x \in \chi, C_x = \omega^{q_x^1}, C_x' = H_1(att(x))^{q_x^1}$$
$$\forall y \in \gamma, C_y = \omega^{t_y^0}, C_y' = H_1\left(A_t\right)^{t_y^0}$$
$$C_y'' = \left(v_{\{t_i,t_j\}}\right)^{t_y^0} = \varphi^{t_y^0 \lambda^{t^i} \mu^{z-t_j}}$$
$$\forall z \in Z, C_z = \omega^{l_z^0}, C_z' = H_1\left(A_l\right)^{l_z^0}$$
$$C_z'' = \left(v_{\{l_i,l_j\}}\right)^{l_z^0} = \varphi^{l_z^0 \lambda^{l^i} \mu^{z-l_j}} \Big\}.$$

- Algorithm 4. Re-encryption algorithm

When a user requests access to the cloud server, the cloud server operates a re-encryption algorithm to convert the ciphertext CT to RC, which effectively ensures that the re-encrypted ciphertext is ultimately dependent on location and time. In particular, for each node $y \in \gamma, z \in Z$, the cloud server will examine whether time t_c and location l_c of the node satisfy the constraint of time range $[t_a, t_b]$ and the constraint of position range $[l_a, l_b]$. If not, label $\widetilde{C_y''}, \widetilde{C_z''}$ as the special symbol $\perp$; if the constraints are satisfied, the calculation below is operated:

$$\begin{aligned} \tilde{\mathrm{C}}_y'' &= \mathrm{C}_y \cdot F_{\{t_i \le t_c, t_j \ge t_c\}}\left(C_y''\right) \\ &= \mathrm{C}_y \cdot F_{\{t_i \le t_c, t_j \ge t_c\}}\left(v_{\{t_i,t_j\}}\right)^{t_y^t} \\ &= \mathrm{C}_y \cdot \left(\varphi^{t_y^0 \lambda^{t^i} \mu^{z-t_j}}\right)^{\lambda^{t_c - t_i} \mu^{t_j - t_c}} \\ &= \omega^{t_y^0} \cdot \left(v_{\{t_c,t_c\}}\right)^{t_y^0} \\ &= \left(v_{\{t_c,t_c\}}\omega\right)^{t_y^0}. \end{aligned}$$

$$\begin{aligned} \tilde{\mathrm{C}}_{\mathrm{z}}'' &= \mathrm{C}_z \cdot F_{\{l_i \le l_c, l_j \ge l_c\}}\left(C_z''\right) \\ &= \mathrm{C}_z \cdot F_{\{l_i \le l_c, l_j \ge l_c\}}\left(v_{\{l_i,l_j\}}\right)^{l_z^0} \\ &= \mathrm{C}_z \cdot \left(\varphi^{l_y^0 \lambda^{l_i} \mu^{z-l_j}}\right)^{\lambda^{l_c - l_i} \mu^{l_j - l_c}} \\ &= \omega^{l_z^0} \cdot \left(v_{\{l_c,l_c\}}\right)^{l_z^0} \\ &= \left(v_{\{l_c,l_c\}}\omega\right)^{l_z^0}. \end{aligned}$$

Thus the final re-encrypted ciphertext is

$$RC = \Big\{\mathrm{T}, C, C', \tilde{C}, \{\mathrm{C_x}, \mathrm{C_x'}\}_{\forall \mathrm{x} \in \chi},$$
$$\{\mathrm{C_y}, \mathrm{C_y''}\}_{\forall \mathrm{y} \in \gamma}, \{\mathrm{C}_z, \mathrm{C}_z''\}_{\forall z \in \mathrm{Z}}\Big\}.$$

The cloud server sends the encrypted ciphertext $\mathrm{CT}^{'}$ to the user, as well as the current time t_c and current location l_c.

- Algorithm 5. Decryption algorithm

In the decryption phase, the user uses the private key to decrypt data. First define a recursive algorithm $\mathrm{Dec}rytNode\left(RT, SK_{gid}, x\right)$. Input ciphertext RC, private key $\mathrm{SK_{gid}}$, and node x.

If x is a leaf node, then let $i = attr\left(x\right)$.

If $i \in S_{gid}$, we have

$$\begin{aligned} F_x^{\mathrm{attr}} &= \mathrm{Decryt}\,No\,\mathrm{de}\left(RC, SK_{gid}, x\right) \\ &= \frac{e\left(D_i, C_x\right)}{e\left(D_i', C_x'\right)} \\ &= \frac{e\left(\left(gH_0(gid)\right)^{u_j} H_1(i)^{r_i}, \omega^{q_x^1}\right)}{e\left(\omega^{r_i}, H_1(\mathrm{att}(x))^{q_x^1}\right)} \\ &= e\left(gH_0(gid), \omega\right)^{u_j q_x^1}. \end{aligned}$$

If $i \notin S_{gid}$, define $\mathrm{Dec}rytNode\left(CT^{'}, SK_{gid}, x\right) = \perp$.

Next we consider nodes $\forall y \in \gamma$, which include leaf and non-leaf nodes and require $t_c \in [t_i, t_j]$ and $t_c \in$

$[t_a, t_b]$ while ensuring secure access control within a valid time range. $A_t\left[t_i, t_j\right]$ is the access policy for node y in the access policy tree, and $A_t\left[t_a, t_b\right]$ is the time range when the user is given access right. When the access time is valid, it is calculated as follows:

$$\begin{aligned}
\tilde{\mathrm{D}}_t'' &= F_{\{t_a \leq t_c, t_b \geq t_c\}}\left(\mathrm{D}_{\mathrm{t}}''\right) \\
&= F_{\{t_a \leq t_c, t_b \geq t_c\}}\left(v\left\{t_a, t_b\right\}\right)^{r_t} \\
&= \left(v_{\{t_c, t_c\}}\right)^{r_t} .
\end{aligned}$$

$$\begin{aligned}
F_{\mathrm{y}}^{\text{time}} &= \frac{e\left(D_t, \tilde{C}_y''\right)}{e\left(D_t' \tilde{\mathrm{D}}_t'', C_y'\right)} \\
&= \frac{e\left(\left(gH_0(gid)\right)^{u_j} H_1\left(A_t\right)^{r_t}, \left(v_{\{t_c, t_c\}} \omega\right)^{t_y^0}\right)}{e\left(\omega^{r_t}\left(v_{\{t_c, t_c\}}\right)^{r_t}, H_1\left(A_t\right)^{t_y^0}\right)} \\
&= e\left(gH_0(gid), \omega\right)^{u_j t_y^0} \cdot e\left(gH_0(gid), v_{\{t_c, t_c\}}\right)^{u_j t_y^0} \\
&= e\left(gH_0(gid), \omega\right)^{u_j t_y^0} .
\end{aligned}$$

Similarly, we consider nodes$\forall z \in Z$ in the access policy tree, which include leaf and non-leaf nodes. Requirel $l_c \in \left[l_i, l_j\right]$ and $l_c \in \left[l_a, l_b\right]$ while ensuring secure access control within a valid time range. $A_l\left[t_l, t_l\right]$ is the access policy for node z, and $A_l\left[l_a, l_b\right]$ is the location range where a user is given access rights. When the access location is valid, then the calculation is as follows:

$$\begin{aligned}
\tilde{\mathrm{D}}_l'' &= F_{\{l_a \leq l_c, l_b \geq l_c\}}\left(\mathrm{D}_l''\right) \\
&= F_{\{l_a \leq l_c, l_b \geq l_c\}}\left(v\left\{l_a, l_b\right\}\right)^{r_l} \\
&= \left(v_{\{l_c, l_c\}}\right)^{r_l} .
\end{aligned}$$

$$\begin{aligned}
F_{\mathrm{z}}^{loc} &= \frac{e\left(D_l, \tilde{C}_z''\right)}{e\left(D_l' \tilde{\mathrm{D}}_l'', C_z'\right)} \\
&= \frac{e\left(\left(gH_0(gid)\right)^{u_j} H_1\left(A_l\right)^{r_l}, \left(v_{\{l_c, l_c\}} \omega\right)^{l_z^0}\right)}{e\left(\omega^{r_l}\left(v_{\{l_c, l_c\}}\right)^{r_l}, H_1\left(A_l\right)^{l_z^0}\right)} \\
&= e\left(gH_0(gid), \omega\right)^{u_j l_z^0} \cdot e\left(gH_0(gid), v_{\{l_c, l_c\}}\right)^{u_j l_z^0} \\
&= e\left(gH_0(gid), \omega\right)^{u_j l_z^0} .
\end{aligned}$$

If a node x is independent of any time trapdoor and location trapdoor, namely, the trapdoor is not exposed, then the following equation is obtained:

$$\begin{aligned}
F_x = F_x^{attr} &= e\left(gH_0(gid), \omega\right)^{u_j q_x^q} \\
&= e\left(gH_0(gid), \omega\right)^{u_j q_x^0} .
\end{aligned}$$

If a node x is associated with a time trapdoor, obtain the following equation is obtained:

$$\begin{aligned}
F_x &= F_x^{attr} \cdot F_x^{time} \\
&= e\left(gH_0(gid), \omega\right)^{u_j\left(q_x^1 + t_x^0\right)} \\
&= e\left(gH_0(gid), \omega\right)^{u_j q_x^0} .
\end{aligned}$$

If a node x is associated with a location trapdoor, the following equation is obtained:

$$\begin{aligned}
F_x &= F_x^{attr} \cdot F_x^{loc} \\
&= e\left(gH_0(gid), \omega\right)^{u_j\left(q_x^1 + l_x^0\right)} \\
&= e\left(gH_0(gid), \omega\right)^{u_j q_x^0} .
\end{aligned}$$

If a node x is associated with a time trapdoor and a location trapdoor, the following equation is obtained:

$$\begin{aligned}
F_x &= F_x^{attr} \cdot F_x^{loc} \cdot F_x^{\text{time}} \\
&= e\left(gH_0(gid), \omega\right)^{u_j\left(q_x^1 + l_x^0 + t_x^0\right)} \\
&= e\left(gH_0(gid), \omega\right)^{u_j q_x^0} .
\end{aligned}$$

Finally we need to consider the recursive case when x is a non-leaf node. Child nodes z of node x constitute set S_x and the number of child nodes in the set is k_x. If there is no S_x, the decryption function returns $\perp$. Otherwise, the computation is operated as below:

$$\begin{aligned}
\mathrm{F}_x^{\text{attr}} &= \prod_{z \in S_x} F_z^{\Delta_i, S_x'(0)} \text{ where } \left\{\begin{array}{l} \mathrm{i} = \mathrm{index(z)} \\ \mathrm{S}_{\mathrm{x}}' = \{\mathrm{index}(z) : z \in S_x\} \end{array}\right. \\
&= \prod_{z \in S_x}\left(e\left(gH_0(\text{ gid }), \omega\right)^{u_j q_x^0}\right)^{\Delta_i, S_x'(0)} \\
&= e\left(gH_0(gid), \omega\right)^{u_j q_x^1} .
\end{aligned}$$

Concerning the root node R, if the access control policy is satisfied, we will get

$$F_{\mathrm{R}} = e\left(gH_0(gid), \omega\right)^{u_j q_R^0} = e\left(gH_0(gid), \omega\right)^{u_j s} .$$

So the ciphertext can be decrypted to get the plaintext:

$$\begin{aligned}
\kappa &= \frac{C}{e\left(D, C'\right) \cdot F_R} . \\
\mathrm{m} &= \mathrm{Dec}(\kappa, \tilde{C}).
\end{aligned}$$

6 TSC-CABE Analysis

6.1 Security Analysis

- Security model

  This paper sets up a security simulation by attacker Alice and challenger Bob.

Initialization 1: Alice challenges the access structure Γ.Assume that it knows the relevant access policy A_P consisting of the set of attributes, the constraint T_P consisting of the time range, the constraint L_P consisting of the location range.

Initialization 2: Bob gets the public parameter PKP and the master key MK by operating the system initialization algorithm, and sends PKP to Alice.

Phase 1: Alice submits the private attributes, time interval and location range $T_{gid} = [t_a, t_b], L_{gid} = [l_a, l_b]$ of its own query to Bob. Then Bob gets SK_U based on the above information by operating the key generation algorithm and sends it to Bob.

Challenge: Bob Alice sends Alice Bob two messages m_0, m_1, of the same length as message M. Alice Bob selects any bit $b \in \{0,1\}$ encrypts message M by accessing structure tree T to get m_b, and then sends the ciphertext to Alice.

Phase 2: Bob Alice repeats phase 1, and it is assumed that Bob Alice gets its own location range through AliceBob. However, due to $(S_u \wedge TI_u \wedge LP_u) \notin \Gamma$, the decryption fails.

Guess: Suppose that Bob Alice guesses that b could be b', the probability of the opponent making guesses during the entire game is $Pr\left[b^{'} = b\right] - 1/2$.

Definition 1. *If all opponents with temporal polynomials have a non-negligible advantage to some extent, the proposed TSC-CABE scheme can be effective in defending against selective plaintext attacks.*

- Security for different types of enemy attacks
  We further divide the attackers of the TSC-CABE scheme into two categories.

  1) Attackers who do not satisfy the attributes in the access tree.
  2) Attackers who do not satisfy the location/time range in the access tree.

  If it successfully defends against the two types of attacks, the scheme is resistant to any individual attack.

  The TSC-CABE scheme is further optimized on the basis of the TSC-CABE scheme by embedding the location range constraint into the access policy, but its algorithm does not destroy the structure of the original scheme, therefore, this scheme has the same data confidentiality as the TSC-ABAC scheme when attacked by type 1 attackers.

  We flexibly use the trapdoor in this scheme to embed location range constraint and time range constraint in the access tree. The setting and exposure of the trapdoor enable an identity-based encryption scheme, thus the scheme is secure in the random oracle model. It is impossible to obtain authorized access for attackers who do not satisfy the time and location range constraints.

- Security for different source attack
  The TSC-CABE scheme resists illegal access from many different sources

  1) Resistance to illegal access from the cloud storage platform
     When encrypting data, the data owner first encrypts the message m with a symmetric key $ek \in G_T$ and then encrypts ek according to the defined access policy. The symmetric key is maintained by the cloud storage platform, but to decrypt the data, the root node value s of the access tree needs to be restored, which the cloud storage platform fails to do. As a result, the platform cannot share the secret data, and the scheme can effectively prevent the cloud storage platform from illegally accessing the data.
  2) Resistance to illegal access from time/location servers
     In the TSC-CABE scheme, the time/location servers only play a role in decrypting the trapdoor associated with time/location and have no other privileges to compromise the security of the scheme. If a time server or location server is attacked, it will only affect access to the relevant point, while other part of access control related to attributes or associated with temporal constraints is not affected.
  3) Resistance to illegal access from authorized institutions
     If it obtains the attribute key through illegal channels, to further decrypt the shared data, the authorized organization needs to obtain the symmetric key ek first. According to the bilinear mapping theory, the authorized institution has to collude with the cloud storage platform to obtain ek, but in our model, both are semi-trusted and there is no possibility of collusion. Therefore, this scheme can effectively prevent the authorized organization from accessing the data illegally.

6.2 Comprehensive analysis

- Function characteristics analysis
  Concerning the analysis of functional characteristics, we compare the TSC-CABE scheme with the CP-ABE [21], CBE [29], PPLBAC [2], and TSC-ABAC [17] schemes as shown in Table 1. These schemes give solutions to handling dynamic attributes in the access control based on attribute-based encryption, The CP-ABE scheme achieves fine-grained access control by using an access policy tree that encrypts the user's set of attributes as leaf nodes, but the scheme is too old and more factors need to be considered at present. Seeing the access control is associated with multiple attributes, the CBE [22] scheme integrates comparison between attributes into the access control process, which enables the comparison of multiple attributes of users in the access process and achieves flexible access control. The TSC-ABAC scheme and the PPLBAC scheme consider temporal

and spatial constraints, but time and location are discussed separately. We consider both location and time as normal properties like user name and age, and propose the TSC-CABE scheme, which takes into account the temporal and spatial constraints, and does not require the extra revocation during access, making the whole access process more efficient and flexible.

Table 1: Comparison with other options

Schemes	[21]	[29]	[17]	[2]	Ours
fine-grained	√	√	√	√	√
Support for attribute comparison	×	√	√	√	√
Time frame constraint	×	×	√	×	√
Position range constraint	×	×	×	√	√
Attribute to cancel	Yes	No	No	No	No

- Complexity Analysis
  In this section, the TSC-CABE scheme is compared with the CBE scheme and the TSC-ABAC scheme. Similar to the CBE and TSC-ABAC schemes, our scheme focuses only on the bilinear pair and exponential operations in G and G_T and ignores the hash function operation and the multiplication operation. $|N|$ in the scheme represents the number of leaf nodes in the access policy tree. $|N_{A_t}|$, $|N_{A_l}|$ denote the number of nodes associated with the time range and location range respectively and $|A|$ is the number of attributes involved in encryption and decryption process. l_G, l_{G_T}, l_{Z_n} denote the size of the elements in G, G_T, Z_n respectively.P denotes the overhead consumed by a bilinear pairing, and $E\left(G\right), E\left(G_T\right)$ represents the exponential computation overhead in G, G_T.

  In Table 2, we compare the key size and ciphertext size of the three schemes. In our scheme, the total number of leaf nodes in the access policy tree is much larger than the number of nodes associated with location or time trapdoors, and there is no need for each server to generate parameters during the initialization phase. Therefore the storage cost is significantly reduced. In Table 3 and Table 4, We make comparison in terms of communication overhead and computational complexity. Since the CBE scheme cannot handle non-comparison based attributes, and the TSC-ABAC scheme cannot handle continuous location range, it is assumed that only a time trapdoor and a location trapdoor are embedded in our scheme, which are $|N_{A_t}| = 1$, $|N_{A_l}| = 1$. As shown in Table 3, our scheme handles simultaneously location range and time range without incurring additional expenses.

7 Research Developments

In most existing CP-ABE schemes, such as [4, 9, 21, 24], only one authority is responsible for maintaining the entire set of attributes, which can create a single point of bottleneck. Once the single authority breaks down, the system is paralyzed. Moreover, in real scenarios, attributes vary and require different authorities to distribute keys. Although CP-ABE schemes with multiple authorities have been proposed as in [5, 12, 15], those schemes fail to effectively deal with single-point bottlenecks and improve performance.

Literature [13] proposes a new multi-authority CP-ABE scheme called the TMACS scheme, which is a gated multi-authority access control scheme for public cloud storage. Multiple authorities jointly manage a set of attributes. The TMACS scheme utilizes a (t,n) threshold secret sharing mechanism in order that multiple authorities share the master key. The user needs to interact with t of the authorities to obtain the key. In other words, communication between AAs is not required during the key generation phase, which reduces coupling between attribute institutions. The scheme greatly reduces the communication overhead between AAs, but a problem is ensued that the computational overhead increases since each AA generates the key independently. To solve it, outsourcing the calculation to a cloud server can be considered on the premise of ensuring safety and reliability.

Based on the above analysis, this paper proposes a multi-authority access control scheme based on comparative attributes of spatio-temporal constraints in the cloud.

We regard spatio-temporal factors as normal attributes, and formulate a brief and secure MA-ABE cross-domain data access control scheme. The data owner (DO) defines the access policy tree based on the access policy, encrypts the data based on the tree, and then uploads the ciphertext to the cloud. In addition, DO generates some private key components to prevent joint attacks from several AAs and send them to users through secure channels. Users can freely obtain the ciphertext in the server, but decryption can be successful only when the attributes the user possesses satisfy the requirements of the access policy tree. When a user's attributes change, the cloud needs to re-encrypt the ciphertext and redistribute the private key components. As an attribute generation and authorization institution, AA is responsible for distributing attributes to authorized users and data owners and generating part of attribute-related private key components. CA is a trusted central authentication institution, which is responsible for generating a series of public parameters at the initial stage of authentication. The cloud server is an unreliable storage medium, which is mainly used to store user data.

The solution is as follows:

Table 2: Comparison of storage costs

Schemes	**CBE**	**TSC-ABAC**	**Ours**
key	$(1+4\|A\|)l_G$	$(5+2\|A\|)l_G$	$(7+3\|A\|)l_G$
cipher	$(1+4\|N\|)l_G+1\cdot l_{G_T})$	$(2+2\|N\|+2\|N_{A_t}\|)l_G+1\cdot l_{G_T}+1\cdot l_{Z_n}$	$(2\|N_{A_l}\|+2\|N\|+2\|N_{A_t}\|)l_G$ $+1\cdot l_{G_T}+1\cdot l_{Z_n}$

Table 3: Communication cost comparison

Schemes	**CBE**	**TSC-ABAC**	**Ours**
Setup	$1\cdot p+3\cdot E(G)$	$1\cdot P+(2+\|L\|)\cdot E(G)$	$1\cdot P+2\cdot E(G)$
KenGen	$(1+5\|A\|)\cdot E(G)$	$(7+3\|A\|)\cdot E(G)$	$(9+4\|A\|)\cdot E(G)$
Encrypt	$(1+4\|N\|)\cdot E(G)$ $+1\cdot E(G_T)$	$(2+2\|N\|+3\|N_{A_t}\|)\cdot E(G)$ $+2\cdot E(G_T)+1\cdot P$	$(3\|NAl\|+2\|N\|+3\|N_{A_t}\|)\cdot E(G)$ $+2\cdot E(G_T)+1\cdot P$
ReEncry	–	$\|N_{A_t}\|\cdot E(G)$	$(\|NAt\|+\|NAl\|)\cdot E(G)$
LocToken	–	$1\cdot P+1\cdot E(G_T)+1\cdot E(G)$	–
Delegate	$(1+5\|A\|)\cdot E(G)$	$(5+2\|A\|)\cdot E(G)$	–
DecryptProxy	$(1+4\|A\|)l_G$	$(2\|A\|+4)\cdot P+1\cdot E(G)+\|N\|\cdot E(G)$	–
DecryptUser	$1\cdot P+1\cdot E(G)$	$1\cdot E(G_T)$	$1\cdot E(G_T)$

Table 4: Calculate the cost comparison

scheme	**CBE**	**TSC-ABAC**	**Ours**
Setup	$6\cdot l_G+1\cdot l_{G_T}+2l_{Z_n}$	$(\|L\|+5)\cdot l_G+1\cdot l_{G_T}+2\cdot l_{Z_n}$	$5\cdot l_G+1\cdot l_{G_T}+2\cdot l_{Z_n}$
KenGen	$(1+4\|A\|)l_G$	$(5+2\|A\|)l_G$	$(7+3\|A\|)l_G$
Encrypt	$(1+4\|N\|)l_G+1\cdot l_{G_T}$	$(2+2\|N\|+2\|N_{A_t}\|)\cdot l_G+1\cdot l_{G_T}+1\cdot l_{Z_n}$	$(2\|N_{A_l}\|+2\|N\|+2\|N_{A_t}\|)l_G+1\cdot l_{G_T}+1\cdot l_{Z_n}$
ReEncry	–	–	–
LocToken	–	$2\cdot l_G+l_{Z_n}$	–
Delegate	$3\|A\|\cdot l_G$	$(6+2\|A\|)\cdot lG$	–
DecryptProxy	$1\cdot l_G+1\cdot l_{G_T}$	$1\cdot l_G$	–
DecryptUser	–	–	–

1) Initialization:Select the multiplication group G of prime order p. g is the generator of G. Construct the bilinear map $e : G \times G \rightarrow G_T$,randomly select $\alpha, \eta \in Z_P$, and generate the public key:

$$\mathrm{PK} = (g, G, g^{\eta}, e(g, g)^{a}) .$$
$$MSK = (g^{a}, \eta) .$$

2) Encryption (PK, M, Γ): The data owner encrypts the message M according to the defined access policy. Firstly, the DO formulates an access policy tree according to the attributes distributed by each AA, and randomly selects $s, \rho, \eta \in Z_P$,so the value of the root node is $q_r(0) = s$. The private key value $\frac{q_y(0)}{\rho}$ is assigned to each leaf node in the tree in a top-down manner and the private key value of each leaf node is used for encryption. Let Y be the set of leaf nodes, the ciphertext is as follows:

$$CT = (T, C' = M \cdot e(g, g)^{\alpha s}, C = g^{\eta s},$$
$$\left. \forall y \in Y, C_y = H(\mathrm{att(y)})^{q_y(0)/\rho}, C_{y'} = g^{q_y(0)/\rho} \right) .$$

3) Key generation (MSK, S): Private key generation is completed by the DO and AA.
   - The DO randomly selects $\lambda \in Z_P$ to generate the private key component,$D = g^{(\alpha-\lambda)/\eta}$,and sends D and parameter λp to the user through a secret channel. Since the λp value of each user's private key is different, it can prevent joint attacks launched by a group of users.
   - Each AA randomly selects $r_i \in Z_P$,and generates the corresponding attribute private key component for any attribute $k \in S_j$:
     $\mathrm{SK}_j = (\forall k \in S_j, \mathrm{V}_i = g \cdot H(i)^{r_i}, \mathrm{L}_i = g^{r_i})$
     where $j = 1, \cdot \cdot \cdot, n$,$S_j$ represents the attribute set distributed to users by the jth AA. Let each AA send SK_j to the user via a secure channel.

4) Decryption (CT, SK): The decryption is divided into two parts, decryption by the CSP and decryption by the user. The CSP is only responsible for partial decryption of the data, and the decryption result is sent to the user. Although it can obtain partial result, the CSP cannot obtain the final plaintext, because the key parameter λp is only known by the DO and the user, which ensures the security of the data. The operation is as follows:
   - Decryption by the CSP (DK): After receiving the key sent by the DO and each AA, the user sends the private key component $SK_j \in (1, \cdot \cdot \cdot, n)$ the CSP. The CSP receiving the private key component sent by the user, the decryption algorithm is operated, where the ciphertext as an access policy tree and the private key $SK_j \in (1, \cdot \cdot \cdot, n)$ are entered. The decryption is performed from bottom to top by an recursive algorithm to generate the parameters required for decryption. Let $i = attr(y)$, and $attr(y)$ denotes the attribute value of leaf node y. If x is a leaf node and $x \in S$, then there are

     $$\frac{e(V_{\mathrm{i}}, C'_x)}{e(L_i, C_x)} = \frac{e\left(g \cdot H(i)^{r_i}, g^{\mathrm{q}_y(0)\rho}\right)}{e\left(g^{r_i} \cdot H(i)^{\mathrm{q}_y(0)\rho}\right)} = e(g, g)^{\mathrm{q}_y(0)\rho}.$$

     If x is not a leaf node, for all child nodes z of node x, the result of decryption is denoted as F_Z. Let S_x be the set of child nodes z with the size of K_x. If there is no S_x, then the node x does not satisfy the requirements, the function returns ⊥. Otherwise, perform the calculation below:

     $$\begin{aligned} \mathrm{F}_x &= \prod_{z \in S_x} F_z^{\Delta_i, S'_x(0)} \\ &= \prod_{z \in S_x} \left(e(g, \quad g)^{q_z(0)/\rho}\right)^{\Delta_i, S'_x(0)} \\ &= e(g, \quad g)^{q_x(0)/\rho}. \end{aligned}$$

     $$\text{where } \left\{ \begin{array}{l} \mathrm{i} = \mathrm{index(z)} \\ \mathrm{S'_x} = (\mathrm{index}(z) : z \in S_x) \end{array} \right.$$

     The algorithm calls the Lagrange interpolation function that generates the access policy tree. If the attribute set S satisfies requirements of the access policy tree Γ, then we have

     $$DTK = e(g, g)^{q_R(0)\rho} = e(g, g)^{\frac{s}{\rho}}.$$

     CSP calculates the DK and sends it to the legitimate user.
   - User decryption: After receiving the DK sent by the CSP, the user uses the private key sent by the DO to decrypt again and the calculation below is operated.

     $$\begin{aligned} &\left(e(D, C) \cdot (DTK)^{\lambda\rho}\right) \\ &= e\left(g^{(\alpha-\lambda)/\eta}, g^{\eta s}\right) \cdot e(g, g)^{\lambda\mu s/\rho} \\ &= e(g, g)^{\alpha s}. \end{aligned}$$

   The result is

   $$M = \frac{\mathrm{C'}}{e(g, g)^{\alpha s}}.$$

8 Conclusions

In the encryption of the access control scheme based on spatio-temporal constraints, we optimize the encryption structure of the traditional ABE scheme. The traditional ABE scheme uses the access policy tree to encrypt user's attributes, based on which the structure of the access policy tree is re-designed and the time and location constraints are embedded into the access. The multi-dimensional distance derivation function combined with the trapdoor is used to determine the legitimacy of the user and improve the flexibility of the access process. Finally, through security analysis, function characteristics analysis, comparison of communication overhead and computation overhead, it is shown that the proposed comparative attribute-based encryption scheme based on spatio-temporal constraints is more efficient, flexible and secure than other attribute-based encryption schemes.

In the scheme, we modify the access policy tree to include temporal and spatial constraints. At the same time, we integrate the multidimensional range derivation function and embed it into the process of attribute encryption. The function utilizes the one-way property to represent the total order of integers, thus users who meet the constraints can access resources more flexibly.We also propose a multi-authority access control scheme in cloud environment based on spatio-temporal constraint comparison attributes. In future research, we will start from multi-attribute authorization institutions to study the access control scheme based on attribute encryption in cloud environment, so as to reduce its computing and communication costs. At the same time, we will consider whether users can be unaware in the process of encryption and decryption, and consider whether useless attributes can be eliminated from the perspective of space-time constraints, so as to further reduce the cost of related attributes in the process of encryption and decryption and improve the performance of the algorithm.

Acknowledgments

This study was supported by the Key Project Foundation of Tianjin under Grant 15ZXHLGX003901, Tianjin Natural Science Foundation under Grant 19JCYBJC15800 and National Natural Science Foundation of China under Grant 61802281 and 61702366. The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] E. Androulaki, C. Soriente, L. Malisa, S. Capkun, “Enforcing location and time-based access control on cloud-stored data,” in *IEEE 34th International Conference on Distributed Computing Systems*, pp. 637-648, 2014.

[2] Y. Baseri, A. Hafid, S. Cherkaoui, “Privacy Preserving Fine-grained Location-based Access Control for Mobile Cloud[J],” *Computers and Security*, vol 73,pp. 249-265,2017.

[3] J. Bethencourt, A. Sahai, B. Waters, “Ciphertext-Policy attribute-based encryption,” in *IEEE Symposium on Security and Privacy*, pp. 321-334,2007.

[4] R. Bobba, H. Khurana, M. Prabhakaran, “Attribute-Sets: a practically motivated enhancement to attribute-based encryption,” in *Computer Security - ESORICS 2009, 14th European Symposium on Research in Computer Security, Saint-Malo, France, September*, pp.21-23, 2009.

[5] M. Chase, “Multi-authority attribute based encryption,” in *Proceedings of the 4th conference on Theory of cryptography February 2007*, pp. 515–534,2007.

[6] G. Choi, S. Vaudenay, “Timed-Release Encryption with Master Time Bound Key[M],”Information Security Applications,pp.167-179),2020.

[7] I. Denisow, S. Zickau, F. Beierle, A. Küpper, “Dynamic location information in attribute-based encryption schemes,” in *2015 9th International Conference on Next Generation Mobile Applications, Services and Technologies*, pp. 240-247,2015.

[8] H. Ghafghazi, A. Elmougy, H. T. Mouftah, C. Adams, “Location-aware authorization scheme for emergency response,” in *IEEE Access*, vol. 4, pp. 4590-4608, 2016.

[9] V. Goyal, A. Jain, O. Pandey, A. Sahai, “Bounded Ciphertext Policy Attribute Based Encryption,” in *Proceedings of the 35th international colloquium on Automata, Languages and Programming*, Part II. DBLP,2008.

[10] V. Goyal, O. Pandey, A. Sahai, B. Waters, “Ciphertext-Policy attribute-based encryption,” in *Proceedings of the 13th ACM conference on Computer and communications security*, pp. 89–98,2006.

[11] J. Hong, K. Xue, Y. Xue, W. Chen, D. L. Wei, N. Yu, P. Hong, “TAFC: Time and attribute factors combined access control for time-sensitive data in public cloud,” in *IEEE Transactions on Services Computing*, vol. 13, no. 1, pp. 158-171,2020.

[12] A. Lewko, B. Waters, “Decentralizing attribute-based encryption,” in *Annual International Conference on the Theory and Applications of Cryptographic Techniques Springer, Berlin, Heidelberg*, 2011.

[13] W. Li, K. Xue, Y. Xue, J. Hong, “TMACS: A robust and verifiable threshold multi-Authority access control system in public cloud storage,” in *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 5, pp. 1484-1496, 2016.

[14] X. Li, T. Jung, “Search me if you can: Privacy-preserving location query service,” in *2013 Proceedings IEEE INFOCOM*, pp. 2760-2768, 2013.

[15] H. Lin, Z. Cao, X. Liang, J. Shao, “Secure threshold multi-authority attribute based encryption without a central authority,” *Inf. Sci*, vol.3494, pp.457-473,Springer, 2005vol. 180, no.13, pp. 2618-2632, 2010.

[16] S. Liu, A. Liu, A. Yan, W. Feng, “Efficient LBS queries with mutual privacy preservation in IoV,” *Vehicular Communications*, vol 16,pp. 62-71(10),2019.

[17] Z. Liu, Z. L. Jiang, X. Wang, S. M. Yiu, R. Zhang, Y. Wu, “A temporal and spatial constrained attribute-based access control scheme for cloud storage,” in *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And*

Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), pp. 614-623, 2018.

[18] S. Namasudra, "An improved attribute-based encryption technique towards the data security in cloud computing," *Concurrency and Computation: Practice and Experience*, vol 31,2016.

[19] A. Sahai, B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, vol. 3494, pp. 457-473, Springer, 2005.

[20] J. Shao, R. Lu, X. Lin, "FINE: A fine-grained privacy-preserving location-based service framework for mobile devices," in *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, pp. 244-252, 2014.

[21] E. Shi, J. Bethencourt, T. H. Chan, D. Song, A. Perrig, "Multi-Dimensional range query over encrypted data," in *2007 IEEE Symposium on Security and Privacy (SP '07)*, pp. 350-364, 2007.

[22] S. B. Wang, Y. Zhu, D. Ma, R. Q. Feng, "Lattice-based key exchange on small integer solution problem," *Sci. China Inf. Sci*, vol.57,pp. 1-12,2014.

[23] Z. Wang, D. Huang, Y. Zhu, B. Li, C. J. Chung, "Effificient attribute-based comparable data access control," *IEEE Transactions on Computers*, vol. 64,no. 12, pp. 3430–3443, 2015.

[24] B. Waters, "Ciphertext-Policy attribute-based encryption: an expressive, efficient, and provably secure realization," in *AInternational Workshop on Public Key Cryptography Springer Berlin Heidelberg*, pp. 53-70,2008.

[25] Q. Xie, L. Wang, "Efficient privacy-preserving processing scheme for location-based queries in mobile cloud," in *2016 IEEE First International Conference on Data Science in Cyberspace (DSC)*, pp. 424-429, 2016.

[26] Y. Xue, J. Hong, W. Li, K. Xue, P. Hong, "LABAC: a location-aware attribute-based access control scheme for cloud storage," in *2016 IEEE Global Communications Conference (GLOBECOM)*, pp. 1-6, 2016.

[27] K. Yang, Z. Liu, X. Jia, X. S. Shen, "Time-Domain attribute-Based access control for cloud-based video content sharing: a cryptographic approach," in *IEEE Transactions on Multimedia*, vol. 18, no. 5, pp. 940-950, 2016.

[28] Y. Zhu, G. J. Ahn, H. Hu, S. S. Yau, H. G. An, C. J. Hu, "Dynamic audit Services for outsourced storages in clouds," in *IEEE Transactions on Services Computing*, vol. 6, no. 2, pp. 227-238, 2013.

[29] Y. Zhu, H. Hu, G. J. Ahn, M. Yu, H. Zhao, "Comparison-based encryption for fine-grained access control in clouds," in *Proceedings of the second ACM conference on Data and Application Security and Privacy*, pp. 105–116, ACM, 2012.

[30] Y. Zhu, D. Ma, D. Huang, and C. Hu, "Enabling secure location-based services in mobile cloud computing," in *Proceedings of the second ACM SIGCOMM workshop on Mobile cloud computing*, pp. 27-32, ACM,2013.

Biography

Junling Zhang is a graduate student of Tiangong University with a master's degree in software engineering.She received a bachelor's degree from Shanxi University in 2020. Her research interests include information security, privacy protection, access control and cryptography.

Ze Wang received the B.E. and M.E. degrees from Xi'an Jiaotong University, China, in 1998 and 2001, respectively, and the Ph.D. degree in computer applications technology from Northeastern University, China, in 2004. Since December 2016, he has been a Professor with the School of Computer Science and Technology, Tiangong University, China. His primary research interests include network security, big data modeling and analysis, and privacy computing.

Ping Zhao is a graduate student of Tiangong University with a master's degree in computer science and technology. He received a bachelor's degree from Tiangong University in 2018. His research interests include information security, blockchain data security and privacy protection.

Minghua Gao received a master's degree from Tiangong University in 2021.He received the bachelor degree from Tiangong University in 2018. His research interests include network security and mobile computing.

Shimin Sun received the M.S. and Ph.D. degrees in computer and information communication engineering from the Konkuk University of Korea, in 2009 and 2016, respectively. He is currently an associate professor with Tiangong University, China. His research interests include software-defined networking, the future network architecture, cloud computing, edge computing, and network security.