An Efficient and Secure Access Authorization Policy for Cloud Storage Resources Based on Fuzzy Searchable Encryption

Jun Fu
Guangdong Open University, Guangzhou 510091, China

Abstract—**When fuzzy searchable encrypted cloud storage resources are available, keywords are allowed to have a certain range of changes. Even if there are slight differences in the spelling, word order, or spacing between words, the correct data can be matched. Therefore, it does not have the effect of fine-grained access control (FGAC). Consequently, to satisfy the security demands of cloud storage assets and the ease of resource retrieval through fuzzy searchable encryption, CP-ABE employs attribute and policy definitions to introduce a novel, effective security access authorization approach for cloud storage assets utilizing fuzzy searchable encryption technology. Encrypt cloud storage resources after keyword preprocessing through initialization, file encryption and decryption, index generation encryption, search, and other steps; use the wildcard-based method to generate indexes; and use the Bloom filter to generate security traps to achieve Pail lier-based asymmetric fuzzy searchable encryption of resources. In combination with the CP-ABE-based access control method, authorized users are assigned private keys in the authorization center to ensure that unauthorized users cannot obtain cloud storage resources and complete the fuzzy searchable encryption access authorization of cloud storage resources. The experiment shows that the search index generation of this strategy greatly reduces the resource utilization rate and effectively improves the fuzzy search speed. Moreover, the combination of fuzzy searchable encryption and CP-ABE can better ensure full cloud storage resources.**

Keywords*—*Fuzzy search encryption; cloud storage; security access; CP-ABE (Ciphertext-Policy Attribute-Based Encryption); access control; authorization policy

I. INTRODUCTION

With the rapid development of information technology and the wide application of cloud computing, cloud storage has become an important choice to satisfy the data storage needs of enterprises and individuals due to its convenient, dynamic, easy-to-scale, and on-demand low-cost characteristics [1]. However, data security issues in cloud storage are becoming more and more prominent, including the risks of privacy leakage, data tampering, and illegal access [2], [3]. To secure data in cloud storage, a common practice is to encrypt the data locally before uploading. However, this approach may lead to a loss of the ability to access the data [4]. Therefore, researchers have proposed methods of searchable encryption and setting secure access control policies to protect user privacy while ensuring that only authorized users can access the information allowed by the authorization, thus balancing the security and accessibility of data [5], [6]. In cloud storage environments, data searching using traditional single encryption techniques can face huge time complexity challenges due to the massive nature and high dimensionality of resources. To solve this problem, researchers have started to explore more efficient and flexible encrypted search methods.

Sivas elvan N. and others proposed a security-unified authentication strategy based on the ability of the Internet of Things [7]. In this policy, it is a token that can authorize the entity's access rights. This token is used to ensure authorization and control access to the limited resources on the Internet of Things. In this method, lightweight elliptic curve symmetric key encryption and decryption, message authentication code, and encryption hash primitives are used to complete the access protection of data. This method uses tokens or keys to achieve access control over data. However, if an attacker can obtain or forge these tokens or keys, it is possible to bypass the authentication and authorization mechanisms and gain unauthorized access rights. The privacy protection key aggregation proposed by Padhya M. et al. can search for encryption and access control policies [8]. First, an attacker can intercept the aggregation key or query the trap gate from an insecure communication channel involving the ECS and impersonate an authorized user of the server to access data. Secondly, fine-grained multi-delegation is allowed; that is, if the delegated attributes meet the hidden access policy (defined by the data holder), the delegated can delegate the permissions it receives to another user without compromising data privacy. This method involves ECS, but if the communication channel is not secure, an attacker may be able to intercept the communication and obtain the aggregation key or query information. This may lead to an attacker using the identity of an authorized user to access data. Huso I and others proposed a privacy protection data propagation scheme based on searchable encryption, a publish-subscribe model, and edge computing [9]. The customized edge server is deployed at the edge of the network. It completes searchable encryption of data through four steps: (1) collects subscription requests encoded by a searchable encryption trapdoor; (2) receives data publishing; (3) encrypts through an attribute-based searchable encryption scheme; (3) implements keyword search on encrypted data; and (4) only provides encrypted data to authorized requesters. This method uses searchable encryption technology, which may expose sensitive information related to data when decrypting and searching data. This may lead to the disclosure of data privacy. Chaudhari, P., and others proposed a scheme called Key Sea, which is a keyword-based search for attribute-based encrypted data when the receiver is anonymous [10]. When searching for documents related to target keywords, keeping the anonymity of recipients and ensuring data privacy are important functions of applications such as healthcare, bureaucracy, and social engineering. The Key Sea scheme uses hidden access policies for attribute-based searchable encryption. However, in practical applications, this method may have the risk of cross-anonymous data association when dealing with large-scale data sets, which will lead to threats to data privacy. Based on the analysis of existing research, it is known that current research faces multiple challenges in IoT security and cloud service access control, including the security of tokens or keys, the security of communication channels, data privacy leakage, the balance between anonymity and data privacy, the complexity of fine-grained access control, and performance and efficiency issues in practical applications.

Pail lier-based asymmetric fuzzy searchable encryption has strong data confidentiality and privacy protection, ensures the safe storage and transmission of data in the cloud, supports asymmetric encryption, improves data security and encryption efficiency, enables fuzzy keyword search, and facilitates users to quickly find the required data. CP-ABE-based access control can provide FGAC [11], reduce key management costs, facilitate large-scale system deployment, enhance data security and privacy protection, and prevent unauthorized access and data disclosure. According to the above advantages, this paper proposes an efficient and secure access authorization strategy for cloud storage resources based on fuzzy searchable encryption technology, which can quickly and effectively encrypt cloud storage resources and achieve efficient access control to ensure the security of cloud storage resources. The implementation process is summarized as follows: Firstly, the cloud storage resources are encrypted using fuzzy searchable encryption technology to ensure accurate matching of data even in cases of slight differences in keywords, reducing the exposure of sensitive information and thus reducing the risk of data privacy leakage. Next, a wildcard-based method is used to generate indexes, and a Bloom filter is used to generate security trapdoors to prevent attackers from intercepting communication and obtaining aggregation keys or query information, thereby protecting the identity of authorized users from being impersonated. Implement asymmetric fuzzy searchable encryption based on Paillier. In order to solve the access control problem, combined with the CP-ABE (Attribute Based Access Control) access control method, the authorization center allocates private keys to authorized users, ensuring that unauthorized users cannot access cloud storage resources. At the same time, through the definition of attributes and policies in this process, it ensures that even if an attacker obtains or forges a token or key, they cannot bypass authentication and authorization mechanisms, because access permissions depend on the user's attributes rather than simple tokens or keys, avoiding the risk of cross anonymous data association in large-scale dataset processing, and protecting data privacy from threats.

II. Efficient and Secure Access Authorization Policy for Cloud Storage Resources

A. *Secure Access Authorization Policy Based on Fuzzy Searchable Encryption*

Data stored in cloud storage may contain sensitive information, and data privacy and security need to be ensured. In cloud storage environments, there are usually requirements for multiple users and different permission levels. Efficient and secure access authorization needs to be able to flexibly manage the permissions of different users, including read, write, edit, delete, and other operations. Therefore, efficient and secure access authorization to data can be achieved in the cloud storage environment to ensure the privacy and security of data while maintaining flexibility and efficiency to meet the needs of large-scale data storage. This paper combines the CP-ABE algorithm [12] based on fuzzy searchable encryption to achieve the purpose of fuzzy searchable cloud storage resources and secure access authorization. The overall structure of this strategy is shown in Fig. 1.

As can be seen from Fig. 1, the overall strategy of this paper is composed of four parts, namely, the authorized institution, ECS, data holder, and authorized user. The working process of this method is to generate the master key and public key for the authorized authority and distribute the private key to each user. The data holder generates a security index for local storage resources and uploads the ciphertext and security index to the ECS. Generate authorization credentials for each resource and upload them to the authorization authority. The authorized users generate security traps based on the keywords to be searched and upload them to the ECS with their search tokens. The ECS first matches the security index with the security trap door to obtain a highly relevant resource ID and then requests the authorization authority to audit the access rights of the resource. After the audit is passed, the ciphertext and key are returned to the user. The user obtains the key through calculation and decrypts the ciphertext to obtain the plaintext data. In the overall structure, fuzzy searchable encryption technology uses the proposed method to realize the search and encryption of ciphertext [13], and access authorization uses the control based on CP-ABE scope [14].

The implementation of the whole policy can also be divided into four main parts, which are the system management part, resource processing part, ciphertext search part, and access right authentication part.

The system management part is mainly done by the authorization authority, which realizes the overall management operation of the system. As a trusted third party, the authorization authority is the only fully reliable part, i.e., the authorization center in Fig. 1.

The resource processing part is responsible for the encryption and decryption of resources, and the ciphertext search part realizes the fuzzy search of multiple keywords on the ciphertext, which is mainly realized by the fuzzy searchable encryption technology.

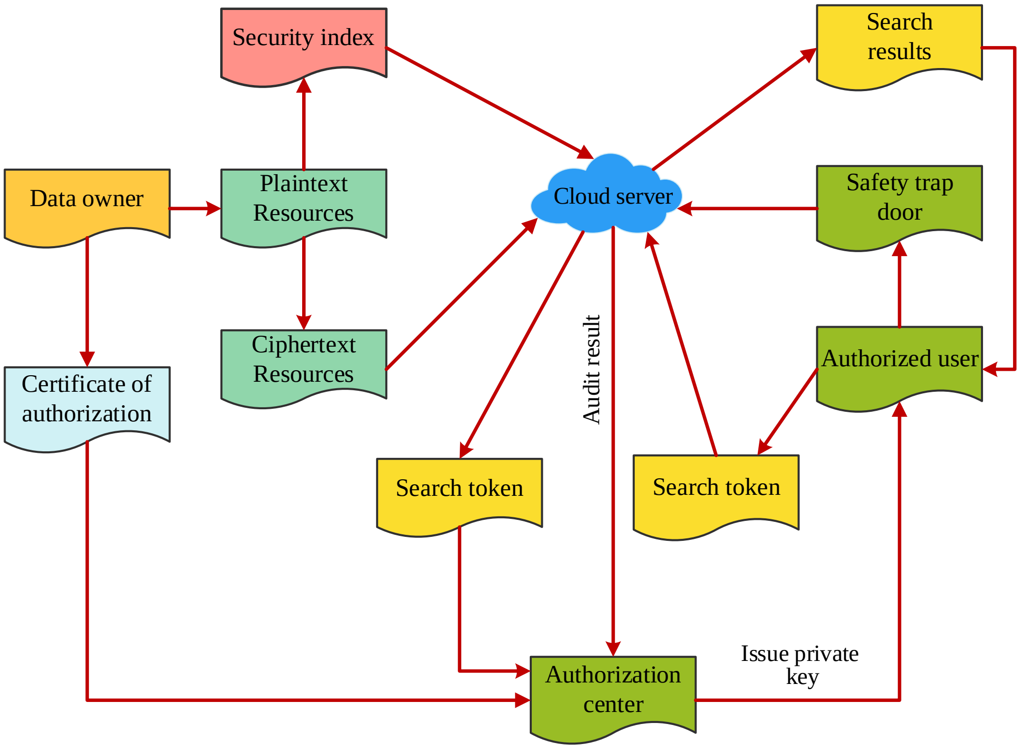


Fig. 1. Overall structure.

Access authority authentication is the core part of authorization search, which is implemented by CP-ABE-based access control.

B. Design of Asymmetric Fuzzy Searchable Encryption Scheme Based on Pail Lier

The access demand for data in cloud storage may change dynamically, and efficient and secure access authorization needs to support dynamic access control policy adjustments to adapt to the change in business requirements. In this context, to safeguard the security and privacy of cloud storage resources and at the same time provide efficient data access control and search functions to adapt to the needs of multi-user, multi-privilege, and dynamic access control, to ensure the security and availability of the system, this paper proposes a secure access authorization strategy based on fuzzy, searchable encryption. The strategy is designed to consider several aspects, including initialization, file encryption and decryption, index generation encryption, and search steps. The key factors in fuzzy searchable encryption are edit distance, fuzzy keyword letting, and keyword trapdoor.

1) Edit distance: Edit distance is used to quantitatively measure the similarity of strings. For a given two keywords, the$\mathrm{w}_1$and w_2, the editorial distanceed $(\mathrm{w}_1, \mathrm{w}_2)$ between them means the minimum number of operations required to change from$\mathrm{w}_1$tow_2. The three basic operations are: (1) Replacement: replacing one letter in a word with another. (2) Delete: delete a letter in the word. (3) Insert: insert a letter into a word. Given a specific keywordw, collection$\mathrm{S}_{\mathrm{w,d}}$represents each word in the set w' that all satisfied the edit distance $\mathrm{ed}(\mathrm{w}, \mathrm{w}' \le \mathrm{d})$betweenw′andw. Heredis a given integer.

2) Fuzzy keyword letting: On the basis of editing distance, fuzzy keyword searching is defined as the following process: Given the collection $C = \{F_1, F_2, \cdots, F_N\}$ containing n encrypted documents stored on the cloud server, a collection of mutually exclusive keywords $W = \{w_1, w_2, \cdots, w_p\}$, a predetermined edit distance$d$, and the search input$(w, k)$of the coded distances$k(k < d)$. The execution result of the fuzzy keyword returns an ID set of all documents that may contain the keyword w isFID_w: if $w = w_i \in W$, return FID_w; if$w_i \notin W$, return$\{FID_w\}$, here$ed(w, w_i) \le k$.

3) Keyword traps: Using bloom filters and locally sensitive hash functions with P-stable distributions to construct index vectors and search trap vectors, keyword traps act as pseudo-random functions, allowing fuzzy keyword-letting schemes to achieve search request privacy and index privacy.

a) Keyword pre-processing: This paper uses a Pail lier-based encryption method to complete fuzzy searchable encryption of cloud storage resources [15]. To Pail lier encrypt the keywords of cloud storage resources, first convert the keywords to an integer. Specific steps: first convert each character in the cloud storage resource keyword to ASC II code, then convert the hexadecimal ASC II code to the decimal integer, and finally accumulate these integers to get a large integer, as shown in Fig. 2.

Pail lier fuzzy searchable encryption also has four parts, namely initialization, file encryption and decryption, index generation encryption, and search. Index generation encryption and search are the core of fuzzy searchable encryption. Different steps will be designed as follows.

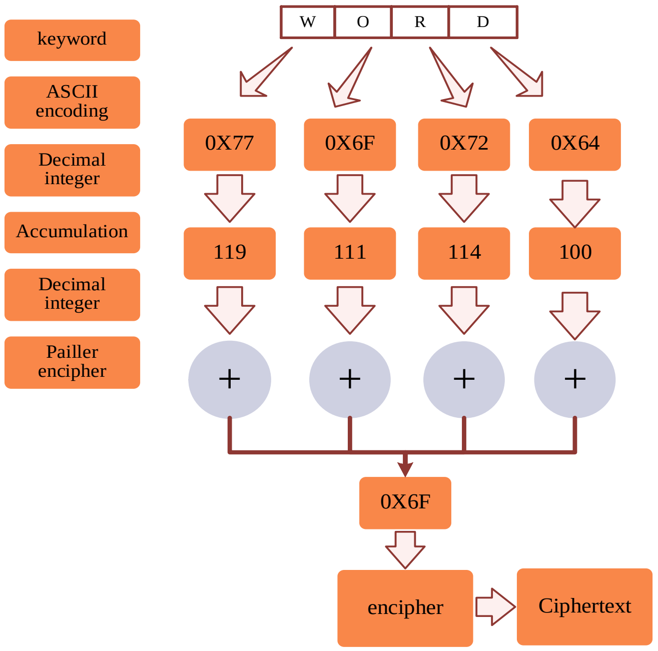


Fig. 2. Keyword conversion process.

b) Fuzzy searchable encryption initialization: For fuzzy searchable encryption, the initialization can be divided into two phases: key initialization and resource initialization [16].

- Key initialization phase: The data holder sets the cloud storage data encryption fuzzy value acc, which is taken by default $acc = 1$ (i.e., the resource edit distance between the two keywords is 1); the data holder enters the security parameter that a, outputs a string of length with a, i.e., the key K, used for AES encryption and decryption; the data holder inputs security parameters λ, and output a pair of public and private keys pk and sk for Paillier encryption, the steps are as follows:
    - ○ Randomly select two large prime numbers $0 < p < 2^{\lambda}$ and $0 < q < 2^{\lambda}$, calculate $n = p \times q$.
    - ○ b. Calculate $\mu = e^{-1}\, mod\, n$, the least common multiple * of p and q, which is e, is expressed as $e = 1cm(p-1, q-1)$.
    - ○ c. To generate the public key pk for (n), private key sk is (e, μ).
- Document initialization phase:
    - ○ a. Extract several keywords from each document in the document collection FS to form a keyword dictionary w'.
    - ○ b. Remove the repeated keywords in the keyword dictionary w', and get the dictionary w containing n keywords.

c) Fuzzy searchable encryption: This section contains two tasks: the resource encryption phase and the resource decryption phase.

- Resource encryption phase.
    - ○ a. The data holder encrypts each resource F_j by AES using the key K, to obtain ciphertext resources CF_j, and generate a ciphertext resource set CFS.
    - ○ b. For each resource CF_j in the ciphertext resource set CFS, setting a unique identifier ID_j.
    - ○ c. Upload the ciphertext resource set CCFS and the corresponding ID_j to a cloud server.
- Resource decryption phase: after the authorized user obtains the key K through the authorization of the data holder, the CFS of the ciphertext resource returned from the search is decrypted by AES to obtain the plaintext resource F_j.

d) Index generation encryption

This section contains two tasks: index generation and index encryption.

- In this paper, a wildcard-based fuzzy set construction is chosen for the index of fuzzy search, which is to say, a keyword fuzzy set. Let the edit distance be d, keywords w based on a wildcard fuzzy set can be expressed as $S_{w,d} = \{S'_{w,0}, S'_{w,1}, \cdots S'_{w,\tau}, \cdots, S'_{w,d}\}$. Here $S'_{w,\tau}$ denotes a keyword w' have number of τ $*$ (wildcard). For example, for the keyword CASTLE, edit the distance $d = 1$, which has a wildcard-based fuzzy set of:

$$S_{CASTLE,1} = \{CASTLE, * CASTLE, * ASTLE, C * ASTLE, C * STLE, \cdots, CASTLE * \{\}\} \tag{1}$$

All the keywords in the set are $13 + 1$iinstead of $13 \times 26 + 1$. Generally, for the keyword w_i of length l, the set $S_{w_i,1}$ constructed by this method has a size of $2l + 1 + 1$, the full keyword of the traditional direct construction method is $(2l + 1) * 26 + 1$. For the direct construction method, the storage capacity will be reduced from 30GB to about 40MB by using the fuzzy set construction based on wildcards.

- Index encryption: Using a public key pk perform bitwise encryption to $S_{w,d} = \{S'_{w,0}, S'_{w,1}, \cdots S'_{w,\tau}, \cdots, S'_{w,d}\}$ to get $CIw_i\{[w_i]_{pk} \| [w_{i1}]_{pk}, [w_{i2}]_{pk}, \cdots, [w_{id}]_{pk}\}$ which ultimately generates a collection of ciphertext indexes $CIS\{CI_{w_1}, CI_{w2}, \cdots, CI_{w_d}\}$. The ciphertext index collection is then sent to the cloud server.

e) Search schemes for fuzzy keywords: Indexed list-based search is a fuzzy keyword search scheme [17], the specific steps of this scheme are as follows:

Step 1: Indexing. In this stage, the data holder encrypts the resources to be stored in the cloud and generates the index of the fuzzy set of keywords, and then uploads both of them to the cloud server, the specific process is as follows: for each keyword $w_i \in W$, the data holder with the key sk computes its gate value for all $w' \in S_{w_i,d}$; the data holder then computes the encrypted address of the corresponding keyword resource store, the

$TEnc\{sk, FID_{w_i}\|w_i\}$, and finally the encrypted resource sets and the list of indexes $\left\{\{T_{w_i'}\}_{w_i' \in S_{w_i,d}}, Enc(sk, FID_{w_i}\|w_i)\right\}$sent together to the cloud server.

Step 2: Request a search. The user wants to search for keywords containing w resources, first at a predefined search distance k under which keyword fuzzy sets $S_{w,k}$are generated; then for each element $w' \in S_{w_i,d}$in the fuzzy set, calculate its trapdoor value and send it to the server.

Step 3: Document search. After the cloud server receives the user's letting request, it performs letting on the index list and returns the corresponding encrypted resource address to the user as the search result. The user decrypts the resource address to obtain the corresponding ciphertext resource, downloads it to the local area for decryption, and ultimately obtains the plaintext resource.

f)Keyword trapdoor: Keywords for querying must be submitted when making a query, and the server cannot obtain this keyword information through the query itself. Authorized users need to generate local security trapdoors for the Chinese keywords to be queried and submit them to the cloud server for querying. The specific steps are as follows:

- Convert keywords into pinyin strings and construct binary vectors. A Bloom filter is constructed, and the LSH function is used to map the binary vector corresponding to each pinyin string to the Bloom filter to obtain the query vector as shown in Fig. 3.

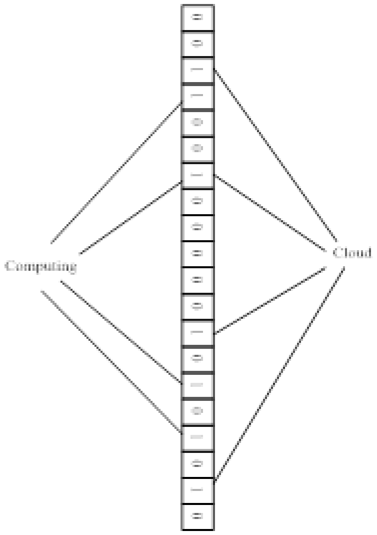


Fig. 3. Example of bloom filter.

- The invertible matrix encrypts the query vector to obtain a secure trapdoor.

Bloom filter: Bloom filter is an efficient data structure that can quickly determine whether an element belongs to the set. It is an array containing m -bits, and initialize each bit to 0 . Usually, Bloom filters are used that r individual hash functions $h_t: \{0,1\}^* \rightarrow [1, n]$, of which, $t \in [1, r]$, and each hash function is mapped to a bit in the array. As shown in Fig. 3, map the words cloud and computing to the Bloom filter, were, $m = 20, r = 4$. The cloud mapped to four positions of $P = \{3,7,12,19\}$by a hash function, computing mapped to 4 positions of $P = \{4,7,15,17\}$by a hash function. So set the value of 7 positions of the Bloom Filter $P = \{3,4,7,13,15,17,19\}$ to 1. When a user submits a search request, the keyword is also hashed. When the value of all hash maps is 1, true is returned. Otherwise, false is returned. The Bloom filter has high time and space efficiency, and when using the Bloom filter as an index to search documents, the required documents will not be missed, which can provide search integrity verification for the system.

C. Access Control of Cloud Storage Resources Based on CP-ABE

The above asymmetric fuzzy searchable encryption scheme based on Pail lier is a form of imprecise keyword searchable encryption, which solves the problem of inconsistent keyword word order and word spacing and improves the fault tolerance and robustness of retrieval conditions. However, it allows keywords to have a certain range of changes. Even if there are slight differences in the spelling, word order, or spacing of keywords, the correct data can be matched. This process tends to reduce the security of cloud storage resource access control. To ensure that only legitimate users can access the corresponding data and ensure the operating efficiency of the system. The CP-ABE algorithm is used to formulate complex access policies, conduct FGAC on data, meet the access needs of different users to data, and ensure the security of data.

The data holder uses the CP-ABE algorithm to connect the cloud storage resources with the access control structure [18], to attain FGAC over the resources saved in CSP, prevent the ECS and unauthorized users from obtaining access rights to cloud storage resources, help manage access rights, and not disclose information about keys or cloud storage resources. In addition, the scheme also meets the following safety requirements:

- Fine-grained access control (FGAC): Authorized users can only access their authorized resources.

- Obfuscation resistance: Authorized users cannot access unauthorized cloud storage resources by sharing keys.

- Privacy protection: The cloud server does not save the user's private information.

CP-ABE algorithm is based on bilinear mapping structure [19], set $G_1$and G_2be two multiplicative cycle groups of the order of prime p the $g$is the generating element of G_1, the $e$is a bilinear mapping, the$e: G_1 \times G_1 \rightarrow G_2$, the bilinear mapping e can be described as below:

- Bilinear: $\forall u, v \in G_1, a, b \in Z_p$, makes $e(u^a, v^b) = e(u, v)^{a,b}$;

- Non-degradation: $e(g, g) \neq 1$;

- Computability: for any $y, z \in G_1$, there exist given polynomial time algorithms to compute $e(y, z) \in G_2$.

CP-ABE data is described by an attribute set [20], which is used to build an access control tree and will then be allocated to cloud storage resources. Authorized users have attribute sets (this task has been assigned by the data holder) and special IDs that describe their access rights. A novel key is allocated to every account associated with a user. If the attribute set held by the authorized user meets the access control tree of the corresponding cloud storage resource, the authorized user can decrypt the cloud storage resource. The scheme is divided into four steps: parameter initialization, encryption operation, generating key, and decrypting operation. In the parameter initialization phase, select the attributes used according to the holder and generate system parameters for each attribute. In the encryption operation phase, the data holder selects an attribute set, utilizes the information to construct a framework for threshold access control, and then uses this structure to encrypt cloud storage resource files. Each authorized user is assigned an attribute set according to the holder. After the authorization center generates the authorized user's private key, the authorized user's public key is used to encrypt the private key, and the encryption result is sent to the ECS. In the decryption operation, authorized users use their private keys to obtain cloud storage resources through the decryption algorithm.

Algorithm for each step:

1) Parameter initialization: The data holder chooses the prime number that P, cyclic groups G_1, G_2, $e: G_1 \times G_1 \rightarrow G_2$ denotes the bilinear mapping, the H represents a hash function. Map the ID of an authorized user asG_1elements, the data holder determines the array of attributes ψ as a secondary consideration, for every attribute $i \in \psi$, the data proprietor produces a pair of arbitrary values $\alpha_i, \beta_i \in Z_p$, then the user's private and public keys are:

$$Sk_u = S_{CASTLE,1}\{\alpha_i, \beta_i, i \in \psi\} \tag{2}$$

$$Pk_u = S_{CASTLE,1}\left\{e(g_1, g_1)^{\alpha_i}, g^{\beta}, i \in \psi\right\} \tag{3}$$

2) Encrypted links: In encrypted messages M, the data holder selects the attribute $i \in \psi$, based on a collection of attributes ψ to define the access structure P. Choosing polynomials that Q_x and P_x, and $Q_x(0) = s$, $P_x(0) = 0$. accessing each leaf node of the access control structure x corresponding to a random number r_x, then calculate:

$$D_{x,1} = e(g_1, g_1)^{Q_x(0)} \cdot e(g_1, g_1)^{\alpha_x r_x} \tag{4}$$
$$D_{x,2} = g_1^{\gamma x}$$
$$D_{x,3} = g_1^{\beta_x \gamma_x} \cdot g_1^{p_x(0)}$$

The ciphertext is encrypted as follows:

$$M_e = Enc_{e(g,g)^x}^{sym}(M) \tag{5}$$

Finally, upload the encrypted cloud storage resource to F Cloud Servers:

$$D = \left\{\forall x, D_{x,1}, D_{x,2}, D_{x,3}, P, M_e\right\} \tag{6}$$

3) Key generation: The data holder receives the authorized user ID_u from the cloud server and selects the attribute collection I_u to allocate the authorized user, the data holder calculates the authorized user key as follows:

$$Sk_D = \left\{g_1^{\alpha_i} H(ID_u)^{\beta_i}, i \in I_u\right\} \tag{7}$$

The encrypted with the authorized public key of the user $Sk_D$is produced to the authorized user through the server of the cloud, and the authorized users are the ones who possess the private key and can decrypt Sk_D.

4) Decryption session: The Authorized users download encrypted cloud storage resources F, $H(ID_u)$, of which $D = \left\{\forall x, D_{x,1}, D_{x,2}, D_{x,3}, P, M_e\right\}$, calculate after the authorized user selects the attributes meeting the access structure P:

$$\prod_x \left(\frac{D_{x,1} \cdot e\big(H(ID), D_{x,3}\big)}{e\big(Sk_D, D_{x,2}\big)}\right)^{\Delta x} \tag{8}$$

$$\prod_x \left(\frac{D_{x,1} \cdot e\big(H(ID), D_{x,3}\big)}{e\big(Sk_D, D_{x,2}\big)}\right)^{\Delta x} = e(g_1, g_1)^s \tag{9}$$

Finally, the user restores the encrypted cloud storage resource to M:

$$M = Dec_{e(g,g)^s}^{sym}(M_e) \tag{10}$$

So far, the efficient and secure access authorization of cloud storage resources that integrates Pail lier-based asymmetric fuzzy searchable encryption and CP-ABE-based access control methods has been completed.

III. Experimental Analysis

To validate the method of this paper, a laboratory computer was chosen to conduct experiments. The parameters of the laboratory computer and the cloud server are shown in Tables I and II, respectively.

Under the above experimental configuration, set up the experimental environment as shown in Fig. 4. The database size is set to 1000 pieces of data, the false alarm rate of the Bloom filter is 0.01, the capacity is 10000 elements, and the average file size is 100KB.

TABLE I. COMPUTER PARAMETERS

Attribute	Parameter
CPU	i5-12400
Internal memory	32GB
Hard disk	1TB
Graphics card	RTX3070
System	windows10
Network bandwidth	10Gbps

TABLE II. CLOUD SERVER PARAMETERS

Attribute	Parameter
Number of CPU cores	32
Memory capacity	10TB
Memory capacity	256GB
Network bandwidth	10Gbps
Storage protocol	NFS, FTP, CIFS Etc.
Data backup	Remote backup to another data center

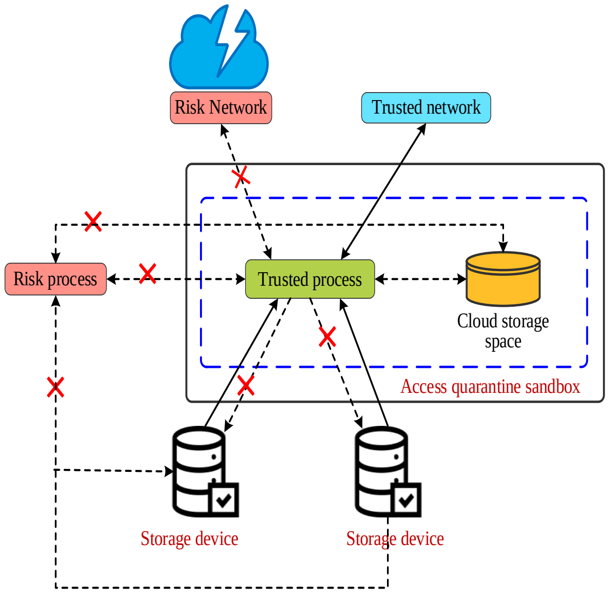


Fig. 4. Experimental environment.

The size of the fuzzy sets constructed based on the wildcard method in the fuzzy search process using the method of this paper at different edit distances is shown in Fig. 5.

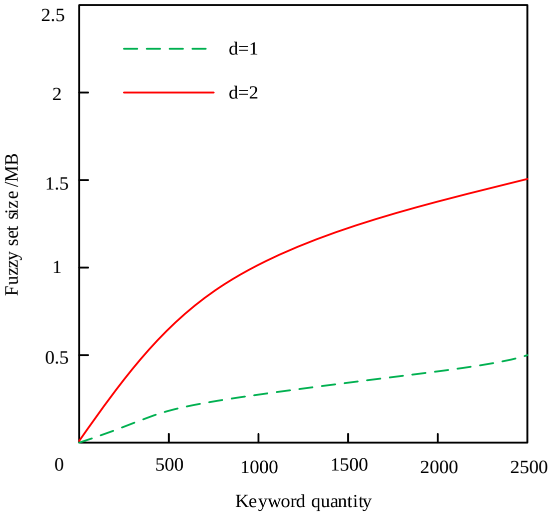


Fig. 5. Fuzzy set size.

By observing Fig. 5, when using this method to build a fuzzy set, the size of the fuzzy set will increase with the increase in editing distance. However, even if the editing distance is 2, the fuzzy set size of 2500 keywords is only 1.5 MB. This shows that the fuzzy set constructed by the method in this paper has the advantages of small data volume, high efficiency, and small space occupation. Compared with the traditional methods, this method has significant advantages in the construction of fuzzy sets. Traditional methods usually need more time and space resources to complete the same task, but this method can generate smaller fuzzy sets in a shorter time, which improves the processing efficiency. This method is efficient and practical in the construction of fuzzy sets and can provide more convenient and fast support for fuzzy search.

The proposed method used in this paper to build the index as well as the time to build the keywords is shown in Fig. 6.

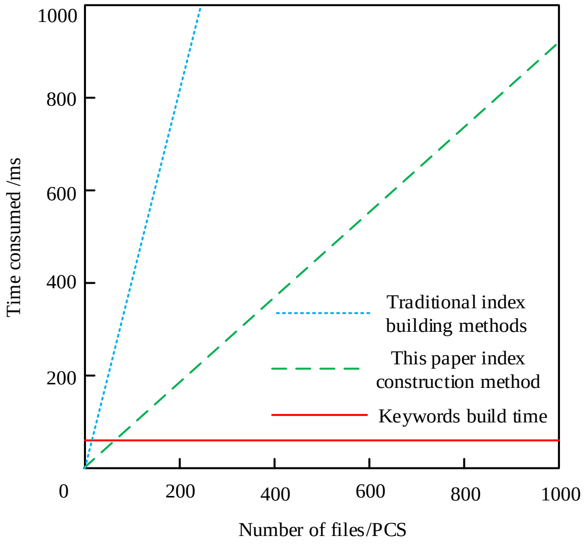


Fig. 6. Index and keyword build time.

By observing Fig. 6, the time overhead increases linearly with the increase in the number of files. This is because the traditional index generation method is less efficient when dealing with a large number of files, and it takes more time to complete the index generation task. In contrast, the index generation method proposed in this paper adopts more efficient algorithms and data structures, which greatly improves index generation speed. In addition, the construction of the keyword verification set and the verification of search integrity operations are executed on the private cloud server. Since these operations need to be performed only once, the time spent is very small. This further proves the efficiency and practicality of the method in this paper. In conclusion, the index generation method proposed in this paper has an obvious time advantage when dealing with a large number of files and can accomplish the index generation task more quickly. This provides strong support for the efficient index generation of private cloud storage systems and helps to improve the overall performance and user experience of private cloud storage systems.

For the fuzzy searchable encryption technique used in this paper, the search time at different numbers of trapdoors is shown in Fig. 7.

By observing Fig. 7, the fuzzy search time will increase as the number of trapdoors increases. This is because the increase in the number of trapdoors will lead to an increase in the amount of data to be searched, thus increasing the search time. However, compared with traditional methods, the search method in this paper has significant advantages in terms of time efficiency. Even if the number of trapdoors reaches 8, with 1000 keywords, the method in this paper can still complete the search in about 35 ms. This advantage in time efficiency benefits from the efficient algorithm and data structure used in this method. By optimizing the search process and reducing redundant operations, this method can locate the target results faster, thus reducing the search time. The search method in this paper has high time efficiency when dealing with large-scale data and can meet the needs of practical applications. Compared with the traditional methods, the advantages of this method in terms of time efficiency make it more suitable for large-scale data fuzzy search scenarios.

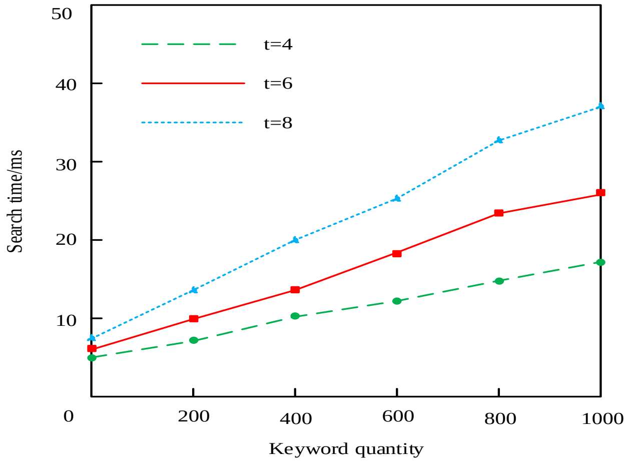


Fig. 7. Fuzzy searchable encryption search time.

In access control, the main reason that affects the access control time is the generation time of the authorization credentials and the generation time of the key, the method of this paper, and the two consumption times as shown in Fig. 8.

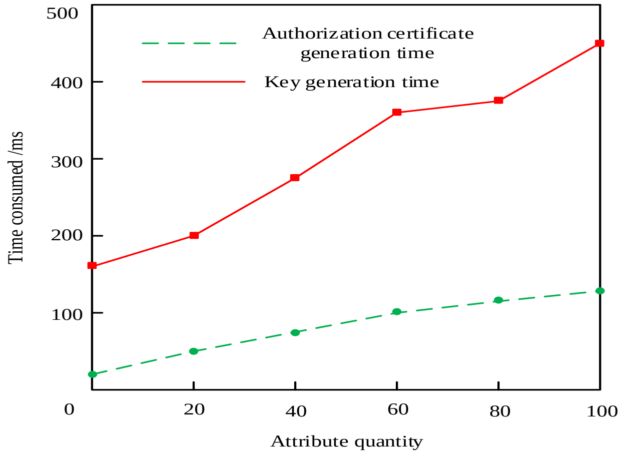
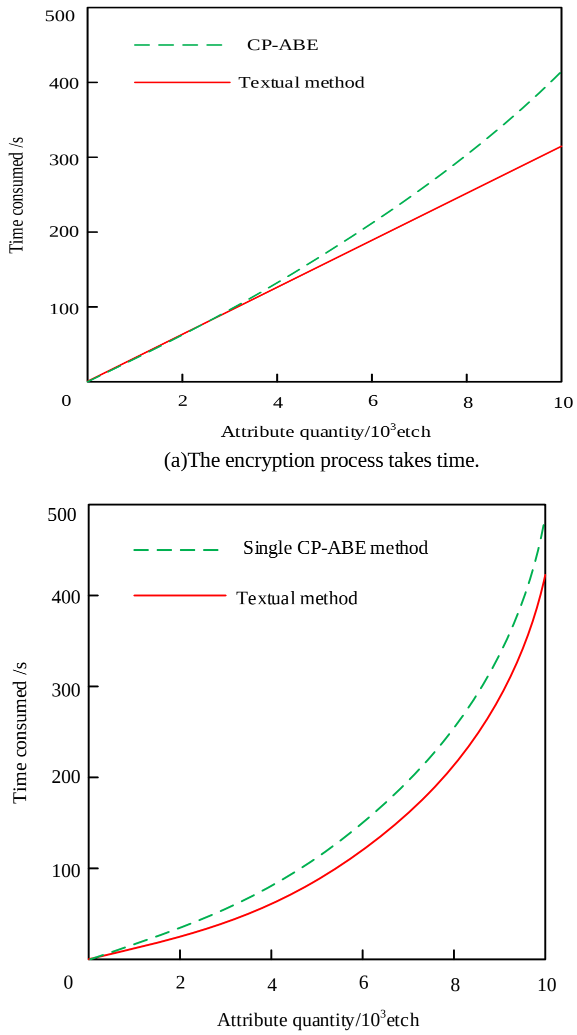


Fig. 8. Influencing factors of access control time.

By observing Fig. 8, among the main factors that affect the access control time, the generation time of authorization credentials accounts for a considerable proportion. This is mainly because the generation of authorization credentials needs to consider the user's attributes, permission levels, and other related factors and verify and calculate them according to the preset access control policies. This process requires a series of computationally intensive operations, such as encryption algorithms and hash functions, which all take a certain amount of time. In addition, with the increase in user attributes, the generation time of authorization credentials will increase accordingly. This is because more attributes mean more calculation and verification steps are required, thus increasing the generation time. When the attribute is 100, it takes 450 ms to generate the authorization certificate.

The file encryption and decryption times of the single CP-ABE access control method and the method in this paper for data encryption security access authorization are shown in Fig. 9.



(a)The encryption process takes time.

(b)Time consuming decryption process.

Fig. 9. The encryption/decryption process takes time.

By observing Fig. 9, the method presented in this article demonstrates significant advantages over a single CP-ABE method in terms of encryption/decryption time for cloud storage resources. By combining fuzzy searchable encryption technology, this method can quickly and accurately complete encryption and decryption operations when dealing with slight changes in keywords, thereby greatly improving the efficiency of the encryption/decryption process. This efficiency improvement not only reduces the waiting time of users but also provides the possibility for large-scale data processing, especially in high concurrency and large data volume cloud storage environments, where our method can maintain stable performance. In addition, the innovation of this method lies in its implementation of fine-grained access control and integration of fuzzy search functions, which is difficult to achieve in traditional CP-ABE methods. By adopting a dual encryption mechanism of CP-ABE encryption and fuzzy searchable encryption, this method has reached new heights in data security and privacy protection. This dual protection mechanism provides a solid protection barrier for data, ensuring that even in complex network environments, data can be protected from unauthorized access and malicious attacks.

Using study [8] method, study [9] method, and study [10] method as comparison methods for our method, we conducted comparative analysis with access delay and throughput as indicators. Among them, access delay is the average time from the user initiating the query to receiving the result, and throughput is the number of queries processed by the server per unit time. The experimental results are shown in Table III.

TABLE III. EXPERIMENTAL RESULTS OF ACCESS LATENCY AND THROUGHPUT

Method	Number of users	Dataset size	Access latency (ms)	Throughput (queries/sec)
Reference [8]	100	1000	150	20
	200	2000	180	18
	300	3000	210	16
	400	4000	240	14
	500	5000	270	12
Reference [9]	100	1000	120	25
	200	2000	140	23
	300	3000	160	21
	400	4000	180	19
	500	5000	200	17
Reference [10]	100	1000	100	30
	200	2000	120	28
	300	3000	140	26
	400	4000	160	24
	500	5000	180	22
Method of this paper	100	1000	80	35
	200	2000	90	33
	300	3000	100	31
	400	4000	110	29
	500	5000	120	27

As shown in Table III, the access latency of our method is lower than that of other methods in all combinations of user numbers and dataset sizes. As the number of users and dataset size increase, the growth rate of access latency is also relatively small, indicating that CP-ABE can still maintain low latency when processing large-scale data. The throughput of this method is higher than other methods in all cases. As the number of users and dataset size increase, the decrease in throughput is also relatively small, indicating that CP-ABE can still maintain high throughput when handling high loads. Overall, it can be seen that the method proposed in this article outperforms other comparison methods in terms of access latency and throughput, especially when dealing with large-scale user and high load datasets. This indicates that CP-ABE has higher efficiency and better performance in fuzzy searchable encrypted access authorization for cloud storage resources. Through fine-grained access control and efficient encryption technology, CP-ABE can better meet the security and query convenience requirements of cloud storage resources.

In order to further verify the universality and applicability of the proposed method, comparative experiments were conducted on different datasets with safety as the indicator. The experimental dataset consists of three different sizes: small (1000 files), medium (5000 files), and large (10000 files), each with a size of 1MB. The experimental results of the methods in this article, study [8], study [9], and study [10] are shown in Table IV.

TABLE IV. EXTENSIVE EXPERIMENTAL RESULTS

Method	Dataset size	Data leakage risk (%)	Access Control Effectiveness (%)
Reference [8]	small-scale	5	95
	medium-sized	7	93
	large	10	90
Reference [9]	small-scale	4	96
	medium-sized	6	94
	large	9	91
Reference [10]	small-scale	3	97
	medium-sized	5	95
	large	8	92
Method of this paper	small-scale	2	98
	medium-sized	4	96
	large	7	93

From Table IV, it can be seen that in small datasets, the data leakage risk of all methods is relatively low. However, the data leakage risk of our method is the lowest, only 2%, and the access control effectiveness is the highest, reaching 98%. This indicates that CP-ABE can provide higher security and more effective access control in small datasets. In medium-sized datasets, as the size of the dataset increases, the data leakage risk of each method increases. However, the growth rate of data leakage risk in our method is relatively small, at 4%, while the effectiveness of access control remains at a high level, at 96%. This indicates that CP-ABE can still provide good security and access control on medium-sized datasets. Under large datasets, the risk of data leakage in various methods further increases. The data leakage risk of this method is 7%, while the effectiveness of access control is 93%, which is still better than other methods. This indicates that even in large datasets, CP-ABE can maintain a lower risk of data leakage and higher effectiveness of access control. Overall, it can be seen that the method proposed in this article exhibits superior security on datasets of different sizes, especially when dealing with large datasets. Through fine-grained access control and efficient encryption technology, CP-ABE can better protect cloud storage resources from the threat of data leakage and ensure the effectiveness of access control. This makes CP-ABE an ideal choice for secure access authorization of cloud storage resources.

In summary, the method in this paper further improves the security and privacy protection of data through double encryption and fuzzy search functions on the basis of realizing access control and providing a more convenient cloud storage service for users. This design idea and method can provide a more secure and efficient solution for the cloud storage system in practical applications.

IV. Results and Discussion

This study proposes a method that combines fuzzy searchable encryption and attribute based encryption based on ciphertext strategy (CP-ABE) to improve the security and search efficiency of data in cloud storage environments. Through a series of experiments and analysis, the following main results and discussions have been obtained.

Firstly, in terms of constructing fuzzy sets, it is observed that as the editing distance increases, the size of the fuzzy set also increases accordingly. However, even with an editing distance of 2, the fuzzy set size of 2500 keywords is only 1.5MB, indicating that the method can effectively support fuzzy search of keywords while maintaining small storage overhead. This result is attributed to the efficient algorithms and data structures used, which greatly improve the speed and efficiency of index generation.

Secondly, in terms of fuzzy search performance, although the search time increases with the number of trapdoors, the search method proposed in this paper has significant advantages in time efficiency compared to traditional methods. Even when the number of trapdoors reaches 8 and the number of keywords reaches 1000, our method can still complete the search in about 35ms. This efficiency is mainly attributed to the optimization algorithms and data structures used, which can quickly locate matching data items.

In terms of access control, it is noted that the generation time of authorization credentials accounts for the main part of the access control time. This is mainly because the generation of authorization credentials involves complex encryption algorithms and computationally intensive operations such as hash functions. However, by optimizing algorithms and reducing unnecessary computational steps, the generation time of authorization credentials can be further reduced. In addition, as user attributes increase, the generation time of authorization credentials will also increase accordingly, but the method can still be completed within a reasonable time.

In terms of encryption/decryption performance, the method proposed in this paper demonstrates significant advantages compared to a single CP-ABE method. By combining fuzzy searchable encryption technology, the method can quickly and accurately complete encryption and decryption operations when dealing with slight changes in keywords, greatly improving the efficiency of the encryption/decryption process. This advantage is particularly important in cloud storage environments as it ensures the security of data during transmission and storage.

In terms of access latency and throughput, our method performs excellently in all combinations of user numbers and dataset sizes. As the number of users and dataset size increase, the increase in access latency is relatively small, while the decrease in throughput is also relatively small. This indicates that CP-ABE can still maintain low latency and high throughput when processing large-scale data, thereby ensuring the availability and performance of cloud storage services.

Finally, in terms of data leakage risk and access control effectiveness, our method exhibits lower data leakage risk and higher access control effectiveness across all dataset sizes. Even on large datasets, the data leakage risk of our method is only 7%, while the effectiveness of access control reaches 93%, which is still better than other methods. This result fully demonstrates the advantages of CP-ABE in providing high security and effective access control.

In summary, the method proposed in this article that combines fuzzy searchable encryption and CP-ABE has demonstrated excellent security and performance in cloud storage environments. By optimizing algorithms and data structures, the speed of index generation and search efficiency can be improved, while reducing the generation time of authorization credentials and the risk of data leakage. In addition, CP-ABE can still maintain low latency and high throughput when processing large-scale data, ensuring the availability and performance of cloud storage services.

V. Conclusion

This paper proposes an efficient and secure access authorization strategy for cloud storage resources based on fuzzy searchable encryption technology. Combining fuzzy searchable encryption and CP-ABE access control, it ensures that data is not accessed illegally and provides flexible and efficient search functions. Fuzzy searchable encryption technology can realize efficient data encryption and retrieval on edge devices, providing a safe and efficient data protection scheme for the development of the Internet of Things and edge computing. In addition, with the wide application of artificial intelligence and machine learning technology, higher requirements are put forward for data privacy protection and secure retrieval. Fuzzy searchable encryption technology can provide a safe and efficient data protection scheme for the training and use of machine learning models and promote the healthy development of artificial intelligence and machine learning. In conclusion, the prospect of an efficient and secure access authorization strategy for cloud storage resources based on fuzzy searchable encryption technology is broad. It will play an important role in big data, the Internet of Things, edge computing, artificial intelligence, and other fields, providing a more secure, efficient, and flexible solution for data security and privacy protection. At the same time, with the continuous development and improvement of technology, this strategy will continue to be optimized and improved to adapt to the changing application needs and market environment.

Although fuzzy searchable encryption technology allows keywords to have a certain range of variation, balancing search flexibility and user privacy protection remains a challenge in practical applications. Therefore, in the future, privacy protection technologies such as differential privacy will be introduced in the process of fuzzy searchable encryption. By adding a certain amount of random noise to the query results, the privacy of users will be protected. Even if attackers obtain the query results, it is difficult to infer specific sensitive information.

REFERENCES

[1] E. S. GSR, R. Ganeshan, I. D. J. Jingle, and J. P. Ananth, "FACVO-DNFN: Deep learning-based feature fusion and Distributed Denial of Service attack detection in cloud computing," Knowl Based Syst, vol. 261, p. 110132, 2023.

[2] N. Chidambaram, P. Raj, K. Thenmozhi, and R. Amirtharajan, "Advanced framework for highly secure and cloud-based storage of colour images," IET Image Process, vol. 14, no. 13, pp. 3143–3153, 2020.

[3] S. Ramasamy and R. K. Gnanamurthy, "Cluster based multi layer user authentication data center storage architecture for big data security in cloud computing," Journal of Internet Technology, vol. 21, no. 1, pp. 159–171, 2020.

[4] S.-M. Chung, M.-D. Shieh, T.-C. Chiueh, C.-C. Liu, and C.-H. Tu, "uFETCH: A Unified Searchable Encryption Scheme and Its Saas-Native to Make DBMS Privacy-Preserving," IEEE Access, vol. 8, pp. 93894–93906, 2020.

[5] B. Alzahrani, N. Fotiou, A. Albeshri, A. Almuhaimeed, and K. Alsubhi, "Distributed access control for information-centric networking architectures using verifiable credentials," Int J Inf Secur, vol. 22, no. 2, pp. 467–478, 2023.

[6] M. Khalid, S. Hameed, A. Qadir, S. A. Shah, and D. Draheim, "Towards SDN-based smart contract solution for IoT access control," Comput Commun, vol. 198, pp. 1–31, 2023.

[7] N. Sivaselvan, K. V. Bhat, M. Rajarajan, A. K. Das, and J. J. P. C. Rodrigues, "SUACC-IoT: Secure unified authentication and access control system based on capability for IoT," Cluster Comput, vol. 26, no. 4, pp. 2409–2428, 2023.

[8] M. Padhya and D. C. Jinwala, "P2 KASE A2—privacy-preserving key aggregate searchable encryption supporting authentication and access control on multi-delegation," IET Inf Secur, vol. 14, no. 6, pp. 704–723, 2020.

[9] I. Huso, D. Sparapano, G. Piro, and G. Boggia, "Privacy-preserving data dissemination scheme based on Searchable Encryption, publish–subscribe model, and edge computing," Comput Commun, vol. 203, pp. 262–275, 2023.

[10] P. Chaudhari and M. L. Das, "Keysea: Keyword-based search with receiver anonymity in attribute-based searchable encryption," IEEE Trans Serv Comput, vol. 15, no. 2, pp. 1036–1044, 2020.

[11] N. C. Rathore and S. Tripathy, "Restricting data-leakage using fine-grained access control on OSN objects," Int J Inf Secur, vol. 22, no. 1, pp. 93–106, 2023.

[12] S. Das and S. Namasudra, "MACPABE: Multi-Authority-based CP-ABE with efficient attribute revocation for IoT-enabled healthcare infrastructure," International journal of network management, vol. 33, no. 3, p. e2200, 2023.

[13] Q. Chen, K. Fan, K. Zhang, H. Wang, H. Li, and Y. Yang, "Privacy-preserving searchable encryption in the intelligent edge computing," Comput Commun, vol. 164, pp. 31–41, 2020.

[14] S. Banerjee, B. Bera, A. K. Das, S. Chattopadhyay, M. K. Khan, and J. J. P. C. Rodrigues, "Private blockchain-envisioned multi-authority CP-ABE-based user access control scheme in IIoT," Comput Commun, vol. 169, pp. 99–113, 2021.

[15] X. Liu, G. Wang, B. Yan, and J. Yu, "KCB-BC-SSE: a keyword complete binary tree searchable symmetric encryption scheme using blockchain," Procedia Comput Sci, vol. 187, pp. 377–382, 2021.

[16] L. Sun, C. Xu, C. Li, and Y. Li, "Server-aided searchable encryption in multi-user setting," Comput Commun, vol. 164, pp. 25–30, 2020.

[17] A. Mortazavi, "Size and layout optimization of truss structures with dynamic constraints using the interactive fuzzy search algorithm," Engineering Optimization, vol. 53, no. 3, pp. 369–391, 2021.

[18] A. Squicciarini, S. Rajtmajer, Y. Gao, J. Semonsen, A. Belmonte, and P. Agarwal, "An extended ultimatum game for multi-party access control in social networks," ACM Transactions on the Web (TWEB), vol. 16, no. 3, pp. 1–23, 2022.

[19] K. Sowjanya, M. Dasgupta, and S. Ray, "A lightweight key management scheme for key-escrow-free ECC-based CP-ABE for IoT healthcare systems," Journal of Systems Architecture, vol. 117, p. 102108, 2021.

[20] P. Sharma, R. Jindal, and M. D. Borah, "Blockchain-based cloud storage system with CP-ABE-based access control and revocation process," J Supercomput, vol. 78, no. 6, pp. 7700–7728, 2022.