IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Secure Encrypted Medical Data Sharing With Identity Based Access Permission

[1] Sudha Devi K, [2] Dhinakaran S, [3] Gokulprasath M,[4] Hariprasath G

[1] Associate Professor, [2] Student, [3] Student, [4] Student
Department of Computer Science and Engineering,
Paavai Engineering College, Namakkal, Tamilnadu, India.

Abstract—Cloud storage services are being used more and more, protecting the security and privacy of data stored in the cloud is becoming increasingly important. Using encryption techniques to encrypt data before uploading it to the cloud is one way to solve this problem. However, because the data must first be decrypted, typical encryption techniques make it more difficult to search for and retrieve certain information from the encrypted data. The effectiveness and utility of cloud storage solutions are hampered by this restriction. This study suggests a novel method for keyword search on encrypted data in cloud storage that makes use of the Advanced Encryption Standard (AES) encryption technique to get around this problem. One popular symmetric encryption technique that is well-known for its reliability and effectiveness is the AES algorithm. The suggested solution guarantees the privacy of data saved in the cloud by utilizing AES encryption, which also makes it possible to conduct effective keyword-based searches without requiring decryption. To facilitate effective data search operations on the encrypted data, the suggested method combines index structures and other cryptographic approaches, such as AES encryption. During the encryption process, a safe index is created that makes it possible to efficiently retrieve encrypted data associated with keywords. Since only authorized users have the decryption keys needed to access the plaintext data, the encrypted data is kept private. The data owner may designate access permissions during data outsourcing according to their user identification. Accessing data from the cloud requires a unique identity and a decryption key for users. Here, the owner of the data can also set a time limit for cloud data access. The findings show that utilizing the AES encryption technique, keyword search on encrypted data in cloud storage strikes a compromise between data privacy and search functionality. The solution provides a practical way to safely store and retrieve data from the cloud while making sure that private data is shielded from unwanted access.

Index Terms *- Component Cognitive Radio (CR), Dynamic Spectrum Access (DSA), Primary User (PU), Secondary User (SU), Software Defined Radio (SDR)*

I. INTRODUCTION

An increasing amount of private data, such as emails, official papers, and personal health records, are being kept in the cloud as cloud computing gains traction. By placing the data in the cloud, the owners can benefit from the high-quality, on-demand data storage service while escaping the burden of maintaining and storing the data. However, the outsourced data may be at risk because the cloud server might no longer be totally trusted because the data owners and the cloud server are not in the same trusted domain. Sensitive data should therefore frequently be encrypted before being outsourced to safeguard data privacy and stop illegal access. [6]. Effective use of data is severely hampered by data encryption, since there may be numerous outsourced data files. Furthermore, cloud computing allows data owners to provide a large user base with access to their outsourced data. Individual users may want to retrieve only specific data files that interest them during a given session. One of the most widely used techniques is to selectively retrieve files using keyword-based search, rather than attempting to recover all encrypted files, which is completely impractical in cloud computing setups. Unfortunately, consumers' ability to utilize keyword searches is restricted by data encryption, making outdated plaintext search methods worthless for cloud computing. Data encryption also requires the privacy of keywords to be preserved because keywords often include important information about the data files. While keyword encryption protects keyword privacy, in this case, traditional plaintext search techniques are useless. "Cloud computing" is a type of Internet-based computing where PCs and other devices can access shared processing resources and data on demand. With the help of this paradigm, a large pool of reconfigurable computing resources—including servers, networks, storage, apps, and services—may be made widely available on demand. These resources can be quickly installed and released with minimum administrative labor required. Thanks to cloud computing and storage choices, users and organisations can store and process their data in privately owned or third-party data centers that may be located far from the user, ranging in distance from across the city to across the world [5]. In order to achieve coherence and scale economies, cloud computing, like a utility (such as the power grid), depends on resource sharing over an electrical network.

Data security and privacy have become major concerns for both individuals and organizations in the modern digital ecosystem. As more and more sensitive data is kept on electronic devices, safeguarding that data against breaches and illegal access has to come first. One of the most reliable methods for protecting data is encryption, and the Advanced Encryption Standard (AES) is a well-known and reliable encryption algorithm that is employed for this purpose. But maintaining access control over encrypted data while guaranteeing safe storage and restricted distribution is a constant issue. The suggested undertaking Since the emergence of cloud computing, data owners are being encouraged to move their intricate data management systems from on-site locations to commercial public clouds in order to take advantage of the increased flexibility and cost benefits. Plaintext keyword search is no longer useful for standard data usage, though, as sensitive data needs to be encrypted before being outsourced to protect data privacy. Enabling an encrypted cloud data search service is therefore essential. Because there are a lot of data users and documents in the cloud, it is necessary to allow multiple keywords in the search request and return documents in the order that they are relevant to these terms. Similar works on searchable encryption mainly concentrate on Boolean or single-word searches, hardly ever sorting the search results [16–18]. This project intends to address these issues by putting forth and putting into practice a novel solution—a comprehensive data and keyword storage system with identity-based data sharing that is encrypted using AES.

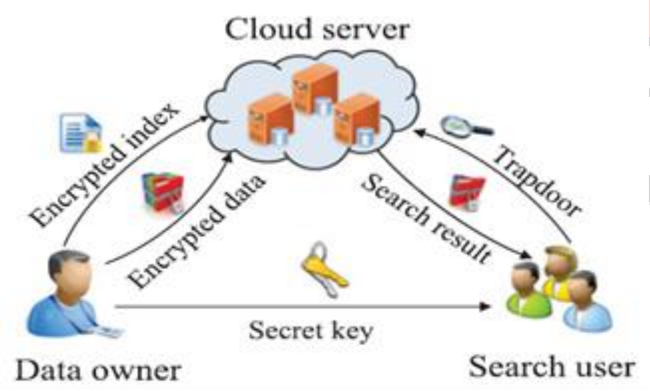

Fig 1. Cloud Data Sharing Model

Fig. 1, describes three general entities that are considered in the cloud data sharing model, namely the data owner and cloud server search users, and the three of them each play the following responsibilities in the ESPD, in that order.

Data Owner: Before sending unprocessed data to the cloud, the data owner must encrypt it. To allow data exchange under the ciphertext, each piece of data is linked to an encrypted index that is constructed using a single keyword and a subset of users. This means that access to this material is restricted to authorized individuals (those who fit into the subset) who possess the target keyword. The data owner then transmits the encrypted indexes and data to the cloud.

Data owner: Encrypting raw data is a requirement for the data owner before uploading it to the cloud. Each piece of data is connected to an encrypted index made up of a single keyword and a subset of users in order to facilitate data sharing under the ciphertext. This indicates that only authorized users—those who fall into the group—holding the target keyword are permitted access to this data. Following that, the data owner sends the encrypted data and encrypted indexes to the cloud .

Cloud server: The primary responsibility of the cloud server is to maintain the integrity of the ciphertext data stored on it, providing a safe environment for multiple users to exchange sensitive information. More specifically, the server is instructed to retrieve the target data that the user is permitted access to and returns ciphertext results that match the keyword being searched when a legitimate user submits a search query.

Users Search: A distinct secret key that is used to generate encrypted search tokens—also referred to as trapdoors—will be provided to each authorized user in ESPD. Then, the authorized user is free to construct the trapdoor, submit it to the server, and have it approved with the help of the cloud server and a keyword. Coded texts.

II. RELATED WORKS

Miao, et.al.,[1] provided a basic ABKS-SM system that preserves privacy and a hidden access policy in a shared multi-owner scenario, as well as a modified ABKS-SM system that supports malicious user tracing. Four categories of entities make up the system model for both the basic and enhanced ABKS-SM systems: various Data Owners (DOs), Data Users (DUs), Cloud Service Providers (CSPs), and Trusted Third-Parties (TTPs). It is noteworthy that traceability is supported by the improved ABKS-SM system, as indicated by the red curve. Numerous services, including data storage, processing, and retrieval, are offered by the cloud server. When a DU submits a search token created based on a keyword they are interested in, the CSP attempts to match it against the indexes and returns to the DU the relevant search results. For cloud clients, including DOs and DUs, TTP is in charge of system initialization and public/secret key pair generation. It may also track the DU, who discloses the private key to unapproved parties. TTP and DOs are thought to be totally reliable. CSP, on the other hand, is thought to be sincere yet inquisitive; it ostensibly adheres to the established protocols while attempting to deduce or collect sensitive information from search or access patterns. Since hostile DUs may purposefully provide partial or altered secret keys for financial gain, DUs are likewise only partially trustworthy.

Qiu, et.al,[2] established a new framework to protect privacy in frequent itemset mining, a process that involves mining data on a public cloud service as it is being collected encrypted. On top of this framework, particularly design three secure frequent itemset mining techniques. Every individual uploads one or more transactions to the CSP. A vast amount of transactions that have been supplied by various users are included in the transaction database that the CSP maintains. In order to help the CSP carry out frequent itemset mining effectively on the transaction database, the data miner can send queries for frequent itemset mining to the CSP and make use of an evaluator. After the executions are complete, the CSP will provide the data miner with a Boolean result (frequent or not) of a mining query. A Miner wants to mine frequently occurring itemsets on the encrypted transaction database that the CSP has stored. Each transaction is represented by a binary vector, while mining queries are represented by another binary vector. If the inner product of a transaction and a mining query equals the number of 1s in the mining query, then the transaction meets the mining query. The support of a mining query on a transaction database can be computed and determined interactively by the CSP and the Evaluator. Now, define three strategies in the next section to execute frequent itemset mining on encrypted data based on distinct privacy criteria. The first approach prioritizes efficiency, while the second one is robust when it comes to privacy protection.

Chaudhari, et.al.[3] developed a searchable encryption system based on a single phrase for applications that allow numerous data owners to upload and access their data. By using attribute-based encryption, the system allows users to view a subset of cloud data without giving the cloud server access credentials. The method is shown to be adaptively secure against chosen-keyword attacks in the random oracle model. The PSE scheme's user-generated trapdoor conceals both the user's attributes and the search term they employed. The index is encrypted by each data owner with the assistance of a reliable authority. Using the master secret key components within the index, the reputable authority secures the index. The adversary, who is able to produce search queries for selected keywords in an adaptive manner, is prevented from learning the keywords from the index by the presence of the master secret key parts. The index is uploaded to the cloud together with the encrypted document once it has been encrypted. The cloud server uses the encrypted index and trapdoor as input when a user delivers it, performing a search operation. For every encrypted index associated with every individual document, the search procedure is repeated. If the user's attributes satisfy the encrypted index's access policy and the term contained in the trapdoor is present in the index, the search operation yields a true result. The cloud server can only determine the search pattern (i.e., whether two search queries were for the same keyword or not) and access pattern (i.e., the number of indexes for which the search process returns true) from the search operation results, even though it has access to encrypted index and trapdoor. Simultaneously, neither the user's qualities nor the search term used are revealed by the trapdoor created by them. This demonstrates that the PSE method is resistant to chosen-keyword attacks in the random oracle model and is adaptively secure. The dataset of terms found in the patient's medical reports is taken into consideration here. Each index pertains to a single report and includes four keywords, which include the report's date, time, kind, and outcome.

Miao, et.al,…[4] two enhanced systems (ABKS-HD-I and ABKS-HD-II) were put into place in order to facilitate user revocation and multi-keyword search, respectively. Unlike state-of-the-art attribute-based keyword search (ABKS) systems, the computing overhead of this approach grows practically linearly with the number of user attributes instead of the number of system attributes. With ABKS-HD, a designated data user can query hierarchical data based on a term of his choice without disclosing the underlying data. Furthermore, fine-grained access privilege control can be accomplished using it. In an effort to enhance the user search experience, ABKS-HD-I allows data users to perform multi-keyword searches in order to prevent returning an excessive number of irrelevant search results. Furthermore, ABKS-HD-II addresses the issue of user revocation, which might result in unwanted access using out-of-date secret keys. The ABKS-HD-I scheme does not take into account the issue of user revocation in dynamic situations, where a particular data user's function may fluctuate over time. In order to address security concerns, the workable plan should stop the individual data user from accessing unauthorized data by utilizing an antiquated secret key. Here, the ABKS-HD-II scheme—which supports both multi-keyword search and user revocation—is described in an attempt to address this issue. For example, every PHR record may be divided into two sections: the medical record M2 and the personal information M1, which would each have the patient's name, gender, social security number, and other details. Later on, a particular physician has to access M1 in order to diagnose a patient, whereas a chemist who specializes in cancer research can only access M2. To securely communicate his PHR data, the patient should ideally encrypt each of his personal data and medical record separately with distinct access policies (T1, T2).

He, Kai et.al,…[5] Added attribute-based access control to a searchable encryption primitive to enable hybrid boolean keyword search over externally encrypted material. There are several desired qualities, including: (1) Data owners can define search limits for encrypted data that is outsourced through access control policies. (2) As long as their attributes are compliant with the access control policy, multiple users are allowed to obtain encrypted data. (3) More in-depth searches, like those requiring Boolean keyword expressions, can only be performed by authorized users. The suggested primitive enables data owners to manage, in accordance with an access control policy, the search authorization for their encrypted data that is outsourced. Any user can do a keyword search as long as his attributes comply with the access control rules. Thus, multiuser search is supported by the suggested primitive. Furthermore, each user possessing a set of attributes has the ability to create a delegated key for a different user with a more limited range of attributes. The suggested architecture offers a more expressive searchable method that allows all authorized users to conduct any desired Boolean keyword expression search, like an access tree structure. The owner of the data outsources his encrypted data to the cloud and manages who has access to it. Trust authority generates private keys for all users in the system. Authorized user whose attributes satisfy the access structure of the keyword ciphertext and who can thus retrieve the data owner's outsourced data. Cloud server that offers computation and storage services, including storing the encrypted data and searching for the encrypted data on behalf of authorized users.

III. EXISTING METHODOLOGIES

When traditional encryption methods fail to recover data, Searchable Encryption (SE) aims to achieve just that. SE techniques typically function by creating an encrypted index. The service provider receives both the encrypted data and the index from the DO. For a given keyword, the Data User (DU) provides the search token; the service provider uses the token and encrypted index to run search algorithms and find matches. Earlier techniques used scanning to return findings, and as the amount of data in the database increased, the efficiency of the scheme declined linearly. By creating an index and extracting keywords, several academics have enhanced SE technology to address the efficiency issue and limit the query complexity to the keywords present in the file set. Unfortunately, the majority of the initial schemes is static and cannot be changed on the fly. Users typically need to update data while keeping it on a cloud server. Dynamic SE technology has evolved to address these demands, increasing the SE scheme's adaptability and accessibility. Nonetheless, the intruder can watch the data updating procedure and identify connections between keywords and files to pilfer or alter the data, adding complexity to the security study. Security designs against a malicious server have not received enough attention, and the majority of current solutions primarily concentrate on an honest yet inquisitive cloud server. The cloud server turns into a malevolent server when an external attack or internal setup error takes place. This might result in changes to the server or the revelation of encrypted data, as well as the possibility of incorrect query results.

Attribute Based Encryption with Secure Data Sharing

Fine-grained access control for encrypted data is made possible via a cryptographic technique known as attribute-based encryption (ABE). ABE can be used for safe data distribution, retrieval, and archiving; this includes using keyword searches and trapdoors. The following describes how to utilize ABE for data distribution, keyword search, and trapdoor-based storage.

Attribute-Based Encryption (ABE):

- Data can be encrypted using ABE so that only users who possess particular traits can decrypt and access the data.
- These attributes can be linked to specific users or the data itself.

Data Storage:

- Data is encrypted and linked to a set of qualities when utilizing ABE for data storage.
- This encrypted data is then kept in a safe location.

Keyword Search:

- A user produces a "search key" or "search trapdoor" with a collection of attributes in order to conduct keyword search on ABE-encrypted data.
- In this instance, the characteristics define the search terms that people are using.

Data Retrieval:

- The person who has the search trapdoor transmits data to the secure data store in order to retrieve it based on keywords.
- After processing the request, the repository provides any data that meets the given attributes.

Data Distribution:

- Data encryption with an access policy that incorporates these features is used when data distribution to users with particular attributes is required; this is how data distribution and data retrieval are intimately associated in ABE.
- The data will only be decrypted and accessible to those who possess the corresponding qualities.

Trapdoor for Data Distribution:

- A data owner or administrator constructs a trapdoor with the required features when distributing data to particular people.

Data Sharing:

- This enables data to be safely disseminated to authorized individuals, who can then decrypt it using their private keys and the trapdoor's properties. Users who possess the trapdoor can use it to access the encrypted data.

IV. Proposed Methodologies

Convenient and scalable data storage options are now available to users thanks to the cloud storage services' explosive rise. However, there are serious privacy and security issues when data is outsourced to unaffiliated cloud providers. In order to mitigate these worries, scholars have created sophisticated encryption methods, like Identity-Based Searchable Encryption (IBSE), which permits safe data retrieval and search features while maintaining data secrecy. To enable effective searching of encrypted data saved in the cloud, IBSE combines the advantages of searchable encryption and indexing approaches. Users can still do keyword-based searches on the outsourced data while storing their data in an encrypted format on the cloud. The Advanced Encryption Standard (AES) and other encryption techniques are used by IBSE to guarantee that the data is kept private and shielded from unwanted access. AES is a popular symmetric encryption technique that has gained popularity due to its effectiveness and security. It utilizes a symmetric key for both encryption and decryption and a block cipher technique. Access control mechanisms based on identity grant user's specific identity access to shared data. Here, a trapdoor might be activated to store encrypted keywords and grant access to the user's identity. Trapdoor verifies the identification of people submitting data access requests before forwarding the requests to the relevant data owner. The data owner can then decide how long data access is allowed to last. IBSE ensures the confidentiality of outsourced data by utilizing identity-based data sharing and AES encryption to stop unwanted parties including cloud service providers from accessing private data. With this method, users may safely store their data on cloud servers and still search for specific information without jeopardizing the secrecy of their data. The suggested plan seeks to balance efficiency and security, guaranteeing a quick and dependable data retrieval process.

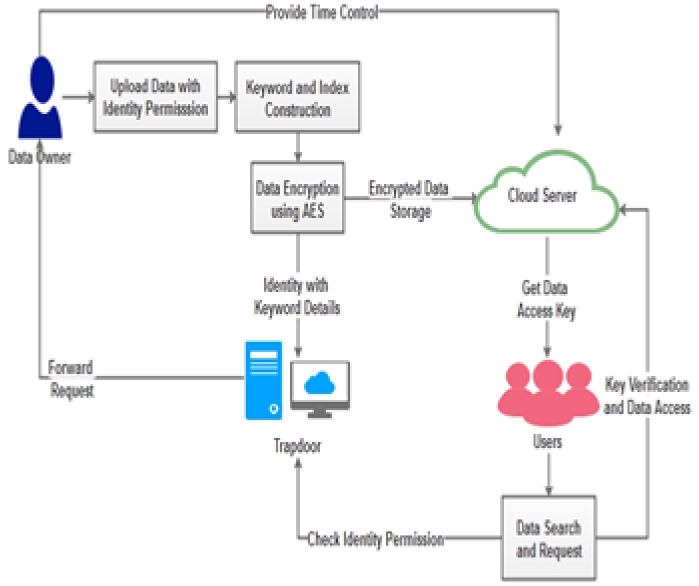
Provide Time Control
Data Owner
Upload Data with Identity Permission
Keyword and Index Construction
Data Encryption using AES
Encrypted Data Storage
Cloud Server
Identity with Keyword Details
Get Data Access Key
Forward Request
Trapdoor
Users
Key Verification and Data Access
Check Identity Permission
Data Search and Request

Fig .2 Architecture for Proposed Work

In fig 2 describes the proposed work of secure data storage and verifiable data distribution process. Entities in the workflow process include the Trapdoor procedure, Cloud Server, Data Owner, and Data User. The data owner may keep encrypted data in the appropriate index together with their pertinent keyword. The encrypted index and identity-based access permission details should be available on Trapdoor. A user can send a request for access to data and do searches. After verifying the user's identity and obtaining permission, Trapdoor will forward the request to the data owner. The owner will provide authorization to access data from the server and define the time limit for data access. The user could then receive decrypted data from the cloud after being validated by the server.

4.1 Proposed Algorithm

4.1.1 AES ALGORITHM

The Advanced Encryption Standard (AES) algorithm is a symmetric encryption algorithm that uses a block cipher to encrypt and decrypt data. The standard defines three key sizes: AES-128, AES-192, and AES-256. The following steps make up the algorithm:

Key Expansion: The 128-bit, encryption key is expanded into a key schedule of 10, 12, or 14 round keys, respectively. The round keys are derived from the original encryption key using a key schedule algorithm.

Initial Round: The plain text is divided into 128-bit blocks and XORed with the first round key.

Rounds: The encryption process consists of a set of rounds (10, 12, or 14) that operate on the state of the cipher. The four transformations that make up a round are SubBytes, ShiftRows, MixColumns, and AddRoundKey.

SubBytes: Each byte of the state is replaced with a corresponding byte from a substitution box (S-box). This step provides confusion and helps to prevent linear cryptanalysis.

ShiftRows: The state is cycled through a set number of steps shifting each row. The second row is shifted one step to the left, the third row is shifted two steps to the left, and the fourth row is shifted three steps to the left.

MixColumns: A fixed polynomial is multiplied by each column of the state. This step provides diffusion and helps to prevent differential cryptanalysis.

AddRoundKey: The round key for the current round is XORed with the state.

Final Round: The final round is the same as the previous rounds except that it does not include the MixColumns transformation.

Output: The resulting cipher text is the final state of the cipher.

The opposite of the encryption process is the decryption procedure. The cipher text is divided into 128-bit blocks and XORed with the last round key. Then, the rounds are performed in reverse order, with the inverse of each transformation used. The final result is the original plain text.

4.1.2 IDENTITY BASED ACCESS CONTROL METHOD

Here explain the process of encrypted data storage with identity permission based data sharing process. The simplified processing steps are given below,

Data Encryption and Keyword Indexing:

- Data is saved in a distributed storage system in the cloud after being encrypted using robust encryption methods.
- For the encrypted data, a keyword index is created, enabling effective and safe keyword-based searches without disclosing the data's content.
- Usually, this index includes details about the keywords connected to every encrypted document.

Identity-Based Access Permission (IBAP):

- Access permissions are established depending on the identities that are assigned to users.
- Similar to Identity-Based Access Control (IBAC), which was discussed in a previous response, access rights can be linked to characteristics or responsibilities that are allocated to users.

Trapdoor Generation:

- A user creates a trapdoor based on their identity and the keyword or keywords they wish to search for in order to gain access to particular data.
- Cryptography techniques are used to generate the trapdoor so that just the information required to make an access control decision is revealed.

Access Request and Trapdoor Verification:

- Users provide the system with their identify and the trapdoor when they wish to access data based on a keyword.
- The system checks to make sure the trapdoor is legitimate for the requested keyword and confirms the user's identity and access rights.
- The system allows access if the access permissions and the trapdoor match.

Data Retrieval:

- The system locates encrypted material that matches the term using the keyword index if access is allowed.
- The data that is encrypted is extracted from storage.

Decryption:

- The data is made available to the authorized user once it has been decrypted using the proper cryptographic keys, which are exclusively in the possession of authorized users.

By using keywords, this procedure makes sure that only authorized people with the right access permissions can look up and access data. Since it conceals the actual keywords and content of the data, the trapdoor serves as a safe and private mechanism for keyword-based searches. Decisions on access control are made in light of the user's identity and permissions, guaranteeing the safe sharing and protection of data.

V. CONCLUSION

A new and all-encompassing approach to protecting sensitive data is the use of the Advanced Encryption Standard (AES) to secure encrypted data along with index construction. Adopting a system that combines time-based controls for data sharing, robust identity management, index generation for rapid data retrieval, and Advanced Encryption Standard (AES) for safe data encryption is a very successful approach to data security and management. This integrated solution helps to accurately retrieve data through keyword searches while also guaranteeing the confidentiality and integrity of sensitive information. Furthermore, identity management strengthens privacy and confidentiality by enabling businesses and individuals to control who has access to the data. This adds an additional layer of security. Users now have even more control over when and for how long data may be viewed thanks to the time-based data sharing function, which also increases security. In a world where data privacy and security are crucial, this suggested technique finds a balance between data protection and usability, making it an essential tool for protecting important information.

References

[1] Miao, Yinbin, Ximeng Liu, Kim-Kwang Raymond Choo, Robert H. Deng, Jiguo Li, Hongwei Li, and Jianfeng Ma. "Privacy-preserving attribute-based keyword search in shared multi-owner setting." IEEE Transactions on Dependable and Secure Computing 18, no. 3 (2021): 1080-1094.

[2] Qiu, Shuo, Boyang Wang, Ming Li, Jiqiang Liu, and Yanfeng Shi. "Toward practical privacy-preserving frequent itemset mining on encrypted cloud data." IEEE Transactions on Cloud Computing 8, no. 1 (2020): 312-323.): 914-927.

[3] Chaudhari, Payal, and Manik Lal Das. "Privacy preserving searchable encryption with fine-grained access control." IEEE Transactions on Cloud Computing 9, no. 2 (2020): 753-762.

[4] Miao, Yinbin, Jianfeng Ma, Ximeng Liu, Xinghua Li, Qi Jiang, and Junwei Zhang. "Attribute-based keyword search over hierarchical data in cloud computing." IEEE Transactions on Services Computing 13, no. 6 (2020): 985-998.

[5] He, Kai, Jun Guo, Jian Weng, Jiasi Weng, Joseph K. Liu, and Xun Yi. "Attribute-based hybrid boolean keyword search over outsourced encrypted data." IEEE Transactions on Dependable and Secure Computing 17, no. 6 (2020): 1207-1217.

[6] Huang, Yaodong, Xintong Song, Fan Ye, Yuanyuan Yang, and Xiaoming Li. "Fair and efficient caching algorithms and strategies for peer data sharing in pervasive edge computing environments." IEEE Transactions on Mobile Computing 19, no. 4 (2019): 852-864.

[7] Wang, Cong, Yuanyuan Yang, and Pengzhan Zhou. "Towards efficient scheduling of federated mobile devices under computational and statistical heterogeneity." IEEE Transactions on Parallel and Distributed Systems 32, no. 2 (2020): 394-410.

[8] Xu, Guowen, Hongwei Li, Hao Ren, Xiaodong Lin, and Xuemin Shen. "DNA similarity search with access control over encrypted cloud data." IEEE Transactions on Cloud Computing 10, no. 2 (2020): 1233-1252.

[9] Li, Zhenhua, and Yuanyuan Yang. "RRect: A novel server-centric data center network with high power efficiency and availability." IEEE Transactions on Cloud Computing 8, no. 3 (2020).

[10] Li, Peng, Song Guo, Shui Yu, and Weihua Zhuang. "Cross-cloud mapreduce for big data." IEEE Transactions on Cloud Computing 8, no. 2 (2020): 375-386.