Survey on Data Storage and Retrieval Techniques over Encrypted Cloud Data

Raghavendra S, Chitra S Reddy, Geeta C M, Rajkumar Buyya, Venugopal K R, S S Iyengar, L M Patnaik

***Abstract*—Cloud Computing is an ever evolving field of technology. In the simplest terms, cloud computing means storing and accessing data and programs over the Internet instead of your computer's hard drive. The cloud offers a variety of services. It reduces the complexity of the networks, makes provision for customization, scalability, efficiency etc. Besides, the information stored on cloud is generally not easily lost. Data stored on cloud is easily susceptible to leak by hackers. In order to prevent this, data is encrypted using Symmetric Searchable Encryption. In such a case, search over the encrypted data becomes difficult and can be executed using various keyword searches as Single Keyword Search, Multi-keyword Search, Fuzzy Keyword Search, Conjunctive Keyword Search, Similarity Search and Synonym Search. In this survey, these keyword searches are explored on the basis of various parameters like security, efficiency, scalability, query effectiveness, architecture and functionality. The overview presented thus compares the different searches on the above mentioned grounds to classify them for various requirements. Sharing of data has become mandatory with the present trending technology overtaking all circumstances. Sharing of data stored in the cloud yields with many advantages. Hence, in this survey we also investigate the various aspects of data sharing on basis of user revocation, competency, encryption techniques, identity privacy and key distribution. Plutus, Sirius, Secure scalable data access scheme, improved proxy encryption and Multi-owner Data Sharing are briefed based on the above mentioned significant parameters.**

***Index Terms*—Cloud Computing, keyword Search, Data Sharing, Information Retrieval, Searchable Encryption.**

I. Introduction

CLOUD computing is witnessing rapid innovations in the recent years. It has two main tasks storing and accessing data and programs by means of Internet rather than usage of a computer's hard drive. The entity cloud presents an extensive range of services. It reduces the complexity of the networks, makes provision for customization, scalability, efficiency etc. Besides, the information stored on cloud is generally not easily lost. Because of its on-demand nature, you could typically buy cloud computing the same way you would buy electricity, telephone services, or Internet access from a utility company. It is so easy with the cloud because one can add extra services (or take them away) at a moment's notice as the business needs change.

As cloud technology is becoming more and more wide-spread, the challenges((like leaking of sensitive data [1], hacking [2], unencrypted data at risk [3–5]involved in maintaining the technology is also increasing. Cloud security, the policies, technologies, controls etc that are used to protect the data, the various applications on the cloud and the associated infrastructure, is becoming an integral field of research in the field of Network Security, and more broadly in Computer Security.The evolution of the Cloud Security policies is equally important to keep up with the cloud issues.

As a kind of emerging business computational prototype, Cloud Computing distributes computation task on the resource pool which consists of a large number of computers and accordingly the application systems gain the computation working strength, the storage space and software service according to its demand. The working of cloud computing can be viewed by two distinctive features One is the cloud infrastructure which is the building block for the upper layer cloud application. The other is the cloud application. Cloud computing has achieved two important goals for the distributed computing by the means of three technical methods. High Scalability the cloud infrastructure can be expanded to very large scale even to thousands of servers and high Availability so that the services are available even when quite a number of servers fail.

The present-day achievements in data, mobile, wireless and Internet technologies cannot be magnified. And hence Cloud computing is an emerging commercial model that promises to eliminate the need for maintaining expensive computing facilities by companies and institutes alike. Cloud computing technology makes it possible develop and host an application design for the internet where information technology (IT) related facilities are provided "as a service"; allowing clients to access technology-enabled services more economically and flexibly on a pay-as-you-use basis. Cloud Computing applications are cloud based services also known as Software as a Service (SaaS). These applications can do everything from keeping track of notes to accounting. Cloud applications give operatives access to their information from anywhere around the globe and must required an Internet connection. This ensures team work, allowing collaborated working as multiple people can view and edit the same information at once. Cloud applications also allow enterprises to push new developments to all users at once, ensuring all round benefit at the same time.

Among the many incentives for using cloud, organizations

Raghavendra S, Chitra S Reddy, Geeta C M, Venugopal K R are with the Department of Computer Science and Engineering, University Visvesvaraya College of Engineering, Bangalore University, Bangalore, India 560001 e-mail: raghush86@gmail.com

Rajkumar Buyya is with Grid Computing and Distributed Systems (GRIDS) Laboratory, Department of Computer Science and Software Engineering, The University of Melbourne, VIC 3053 Melbourne, Victoria, Australia

S S Iyenger is with Department of Computer Science and Engineering, Florida International University, USA.

L M Patnaik is Adjunct Professor and INSA Senior Scientist, National Institute of Advanced Studies, Indian Institute of Science Campus, Bangalore, India.

are looking into ways to assess some of the applications they plan on employing into their environment through the use of a cloud. The adoption of a hybrid cloud approach consents for testing application workloads that can provide a comfortable environment without an initial investment. An organization would seek to have the additional capacity and availability of an environment when needed on a pay-as you-use basis. With cloud computing, there are now readily available environments personalized for your needs often combining automated provisioning of physical and virtualized resources.

Cloud can offer the leeway of storing files and accessing, retrieving and recovering them from any web enabled interface. We have high availability, speed, scalability and security for the environment at all times. There is also the possibility to store the data either on or off premises depending on the regulatory compliance requirements. Yet another benefit resulting from the use of cloud based on the cost effectiveness of a Disaster Recovery (DR) solution that provides for a faster recovery from a network of different physical locations at a much lower cost that the traditional DR site with fixed resources, a much higher cost and rigid procedures. Cloud-based backup can be one of the solutions where we can automatically dispatch data to any location across the wire without any issues of security, availability and capacity.

Libraries are at the brink of accepting the idea of cloud computing because of its both economic and technological advantages. Sharing resources among various academic libraries through Cloud reduces the overall cost and escalate the efficiency. While the list of the above uses of cloud computing is not exhaustive, it certainly gives reasons to use the cloud while considering the traditional alternatives to increase IT infrastructure flexibility, as well as influence on big data analytics and mobile computing.

Storage and Retrieval techniques: Cloud computing is a collection of computing resources used for storing or accessing data from any distant place. Organizations outsource their data on the cloud. Of the many benefits of cloud computing, of which mainly are relief in storage management, global data access and avoidance of capital expenditure on hardware, software and maintenances [6, 7], all these are attributed to the features of on-demand resource availability and pay-as-use concept. The common fact is that many of the organizations encounter obstacles in secure information storage and retrieval on cloud. For resolving these, data comprising of highly confidential data like email, health records, financial transaction and government documents etc has to be encrypted prior to being outsourced to cloud.The possibilities remain that the cloud provider and unauthorized person can breach the security of data stored on the untrusted cloud and obtain it. Data loss and privacy breaches cloud computing systems are reported in [8, 9].

As a result, organizations, health care centres [10] and government are sending the confidential files onto the cloud storage space since they are facing difficulties in maintaining the hardware infrastructure on premises. Many enterprises like Windows Azure, Amazon, IBM etc.. supply cloud services established on basis of IaaS(Infrastructure-as-a-Service). Hence for privacy apprehensions, data intended to be stored is encrypted form; thereafter the owner of the data uploads the data that is encrypted onto the cloud server and later it is retrieved whenever the need arises. Efficient utilization of data stands as a challenge for a enormous number of outsourced data files. An array of data sharing and retreival schemes as shown in Table 1 are available for user accessing data on cloud. Search based on keyword can be titled as one of widespread technique that is made usefor investigating files on encrypted cloud data.Most commonly in plain-text scenarios, keyword search procedures are extensively used and the user is allowed to retrieve chosen files from the storage space.

All the conventional Searchable Symmetric Encryption (SSE)(e.g., [11–15]) paradigms permit a user to examine over cipher text and extract the cipher text securely from the encrypted cloud data by using keywords and not decrypting the stored files.This provisions only techniques like Boolean keyword search devoid of the consideration of any relevance of the document. In a case where enormous number of documents are concerned, Boolean keyword search faces a key disadvantage. It occurs particularly when an user intends to extract matching document for each search request with no prior knowledge of the encrypted cloud data and wishes to examine the entire list of retrieved files, then in such a case it (i) requires huge amount of post-processing in times when examining unrelated files thus producing massive network traffic. (ii) suffers communication overhead.The shortcomings considered above can be resolved with top-k single keyword retrieval techniques [5, 16, 17] and multi-keyword retrieval techniques [18–22].

II. Preliminaries

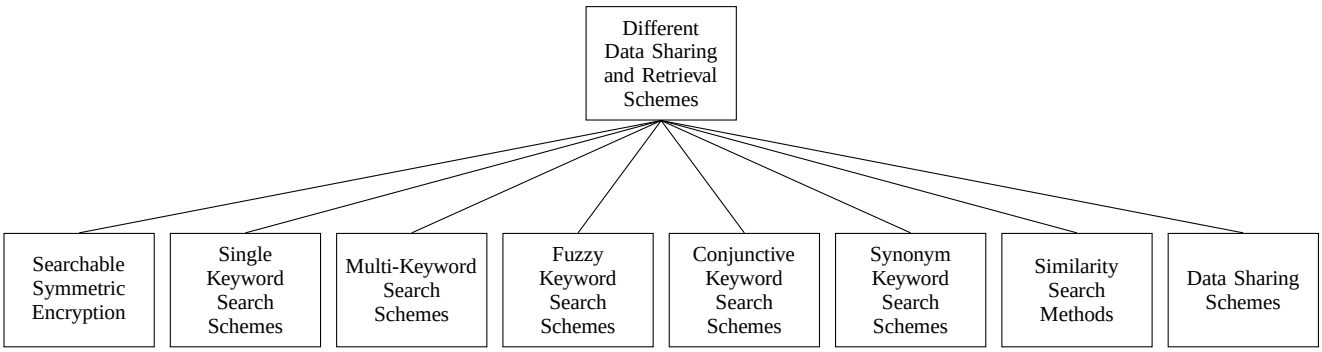
A. Vector Space Model

As mentioned earlier and in [5] for a single keyword search, the ranking is done using the TF-IDF scheme while for multi-keyword search, it is employed using vector space model to score a file. The vector space model [23] is an algebraic model for representing a file as a vector. Every separate term represents a dimension of the vector; for example, in a file when a term occurs, its weight in the vector is non-zero otherwise is zero. Vector Space Model Scheme supports features like allowing an extent of striking similarity between files and queries and then ranks the files based on their relevance and also supporting multi term and non-binary presentation. Based on the weight or score of a file, files are ranked and presented in the top-k ranked order for a given search.

B. Inverted Index

Inverted index is a kind of indexing structure that contains a list of mappings from the set of keywords to the corresponding set of files containing the keyword in the file

TABLE I
DIFFERENT DATA SHARING AND RETRIEVAL SCHEMES



collection uploaded in the cloud server. For Ranked keyword search scheme, the task of determining the files that are most relevant is typically done by assigning a numerical score, which can be pre-compiled, to each file based on some ranking function.

C. Ranking Function

The Ranked Searchable Symmetric Encryption Scheme(RSSE) is based on keyword search. The search generates a ranked order of the files containing a keyword based on the keyword search request sent to the server by a data user; a ranking function or relevance criteria is required to sort the retrieved encrypted files. The most widely used technique for evaluating relevance score is TF x IDF rule, where TF is term frequency which is the number of times a term or keyword is present in a file and IDF is Inverse Document Frequency that is calculated by dividing the number of the files in the entire collection to the number of files containing the particular keyword being searched by the user. Many variations of the TF x IDF Scheme is available, but none of the variant schemes of TF $\times$ IDF rule overshadow the other in terms of outcome [24].

III. SEARCHABLE SYMMETRIC ENCRYPTION

With the concept of data storage in Cloud, Private-key storage is often opted for confidentiality and purpose of data security. In Private-key storage [25–27] when a data user has limited resources, implementation of outsourcing reduces the cost to a minimum for them to store and distribute large amounts of data that is symmetrically encrypted. The reason being that, in regular private-key encryption, data user is not able to retrieve only select segments of their data as search over encrypted data is prevented. In order to solve this issue, proposals have been made for techniques on symmetric encryption with search abilities enabled [11, 28–30]. This paradigm is known as searchable symmetric encryption. The area of searchable encryption has been recognized by DARPA as one of the technical advances that is of significance for security of data for national and private information systems [31].

Another methodology for facilitating symmetric encryption with search abilities is by usage of a secure index [28]. Index can be defined as a data structure where document collections can be stored while supporting efficient keyword search i.e., when provided with a keyword, a pointer to the documents that contain it is given by the index. An index is deemed to be secure if only the data user possessing the trapdoor can perform the search operation for a keyword and also if only the trapdoor can be produced with help of a secret key. If not for the information about trapdoors, no breach of contents by index takes place.

A. Public-key searchable encryption

In the case of searching for data on cloud containing public-key-encrypted data, owner of the decryption key for the encrypted data can be a different person from the other data users who encrypt the data (and direct it to the server).Typically, for a general application a data-user publishes a public key while multiple users send e-mails to the mail server [12, 32]. Any data user with access to the public key can add words to the index, but trapdoors can be generated only by the user who have access to the private key. Trapdoors are used to test for the occurrence of a keyword. The original work on public-key encryption with keyword searches (PEKS) have shown how to build a public-key encryption scheme that hides even the access pattern [12].This construction, however, has an overhead in search time that is proportional to the square root of the database size, which is far less efficient then the best private-key solutions[33].

B. Private-key searchable encryption

In the case of exploring a private-key-encrypted data, the data user encodes the data; so that it can be organized in a random manner way (before encryption) and additional data structures are included so as to provide an efficient access of required data. For allowing an user with the private key to access data, the data and data structures are encrypted and stored on the server. In this method, the rudimentary job of the user of preprocessing data is at least as large as the data,

but the work following that of accessing the data is quite trivial in comparison to the extent of the data for both to the user and the server. Moreover, the entire information about the user's access pattern can be concealed [34, 35].

Song et al., [11] have described different practical techniques for search on encrypted data. The Crypto Systems are secure for encrypted data and untrusted server cannot learn any thing about the plain text based on the search results. Two techniques viz, Hidden queries and query isolation are introduced in this work. The hidden queries searches word without revealing the information to the server and query isolation server learns nothing except the search results. The algorithms are simple, fast, without space and communication overhead. Sequential scan is not efficient and is slow for a large number of documents.

Wang et al., [36] have proposed keyword search encryption technique to resolve the problem of encrypted data through query limitation. The suggested methodology combines the fine-grained access control and keyword search encryption to make available access controls of several users in the cloud setting characterized by encrypted data security. The downside of this scheme is that as the number of access categories of the search files increases, the number of query tokens escalates. The paradigm provides data fortification with high secure strength.

Liu et al., [37] have investigated an efficient privacy preserving keyword search scheme in cloud computing. The cloud server provider does not know any information about specified keywords and encrypted emails. It is able to protect user data and user queried keyword during search process.The construction is based on bilinear maps on elliptic curve to build an efficient Identity-Based Encryption (IBE)[38] and security is based on Bilinear Diffe-Hellman(BDH) assumption. The scheme is semantically secure but the experiment is not performed on the encrypted data.

Jiang et al., [39] have presented a new approach to construct efficient Disjunctively Oblivious Keyword Search (DOKS) protocol which permits fast search and short cipher-text. It provides strong privacy on users side and cloud storage providers. The computation and storage space is less compared to previous Oblivious Keyword Search (OKS) protocols. The privacy and efficiency are better in DOKS protocol. The user submits two search keywords that are not distinguishable and need not know the relation between the cipher-text of the document and search keywords. The matching documents retrieve without revealing statistical information on the search query but the scheme does not support multi-keyword search.

Kumarverma et al., [40] have examined a new Dictionary and Lingual Keyword based Secure Search Scheme for encrypted data stored in the cloud. The technique is to acquire precise encrypted cloud data by using multilingual search queries, phonetic as well as keeping the integrity of data stored in the cloud.The time is reduced to search documents on encrypted cloud data but distribution of dynamic key to enable stronger security and reducing computational cost at user end are not addressed.

Goldreich et al., [35] have proposed oblivious RAM that uses Square-root algorithm and hierarchical solution. $RAMs$ allow clients to completely hide the data access patterns from the cloud server provider. It can be used in conjunction with encryption, to enable stronger privacy guarantees. However, utilising oblivious RAM usually brings exponential number of interactions between the user and the server for each search request.

Goh et al., [28] have outlined a secure index and framed a security paradigm designed for indexes and is known as semantic security against adaptive chosen keyword attack $(ind - cka)$. Secure indexes can be used for examining over encrypted data only in multi-user groups, as the encrypted data files and its' indexes stored at the remote server are regularly apprised. An efficient $ind - cka$ secure index paradigm called $z - idx$ using pseudo-random functions and Bloom filters is developed. in addition, to execute searches over encrypted data that is stored on a remote server accumulated hashing schemes, encrypted and searchable audit logs, database that allows for private queries using a semi-trusted third party, and testing set membership securely are built. This search paradigm has high efficiency, with $O(1)$ search time per file, and takes care of compressed data, variable length words, Boolean and certain regular expression queries. $z-idx$ indexes sacrifices access pattern privacy for efficiency.

Lu et al., [41] designed a novel cryptographic primitive - range predicate encryption - to build a Logarithmic Search over Encrypted Data $(LSED)$ system. This scheme is provably secure with regard to plain-text confidentiality, predicate privacy and supports logarithmic search over encrypted data, query authentication and secure data update. The $LSED$ system reveal the access patterns of cipher texts to the cloud server. Moreover, all database update operation and query authorization relies on the database owner which becomes a single point of failure.

Xia et al., [42] proposed a scheme for basic similarity search over encrypted images based on a secure transformation method that protected the information about features, and do not degrade the result accuracy. The proposed scheme protect the confidentiality of image database, feature vectors, and user's query. Moreover, the image owner could update the encrypted image database as well as the secure index quite easily. This scheme assure the confidentiality of the data, result accuracy and query unlinkability. The time complexity of query on invert index is $O(n)$, which can be further enhanced by using better index to reduce search time.

Pang et al., [43] presented a general framework for multi-user noisy-keyword-based searchable symmetric encryption in a fault-tolerant manner. Existing efforts on multi-user

searchable symmetric encryption (SSE) have focused on exact keyword search, but these results are not applied to the situation where the keywords associated with the files are noisy data. A construction which combines a single-user noisy-keyword-based SSE scheme with a private-key dynamic broadcast encryption scheme is designed. This scheme permits the dataowner to efficiently and dynamically revoke the users. It allowes the authorised users to search the encrypted document set using their chosen noisy keywords with the assistance from an honest-but-curious server. It is secure and correctly realises the goal of multi-user noisy keyword search.

Gu et al., [44] proposed Public Key Encryption with Keyword Search $(PEKS)$ scheme using lattices. $PEKS$ is a method for searching on encrypted data. It enables the user to send a secret value T_w to a server. It enables the server to place all encrypted messages containing the keyword, but without learning anything, but with a probabilistic consistency. The scheme is secure with the hardness of the standard Learning With Errors (LWE). The scheme focuses on security but not on computation cost.

Wang et al., [45] established Static Index (SI) and Dynamic Index (DI) for Public-key Encryption with Keyword Search $(PEKS)$ to make search secure and efficient. SI and DI help $PEKS$ to decrease the load respectively in two parts: If data users are searching queried keyword for the first time, SI is used or else, DI is used, SI and DI are concurrently functional with $PEKS$ and enhanced as Secure Hybrid Indexed Search $(SHIS)$ scheme that uses deterministic encryption (DE) and is convergent. $SHIS$ is improved further for multiple-receiver applications but this extension, support only for one keyword searchable ciphertext.

IV. SINGLE KEYWORD SEARCH

There are many encryption schemes [11–13, 28, 30, 46, 47] available to a data owner to encrypt data for purpose of privacy or security. This leads to impediments for a data user to search the necessary files from a range of encrypted files. If the environment outsourcing is the cloud data is minor, the traditional searchable encryption schemes are compatible with Boolean Keyword Search allowing the user to search the encrypted data of the cloud without a violation to the data privacy. But the case does not stand to be supported the same way in case of outsized cloud environment. In such cases, the user has to review each one of the voluminous number of retrieved files without decrypting the same which only results in post processing overhead. Also this leads to network traffic and undesirable computational cost. To overcome these significant drawbacks and also to warrant file retrieval accuracy, ranked keyword search is preferred. In [48] ranked keyword is defined to enhance system usability by returning the matching files in a ranked order with respect to certain relevance criteria (e.g., keyword frequency), thus making one step closer toward practical deployment of privacy-preserving

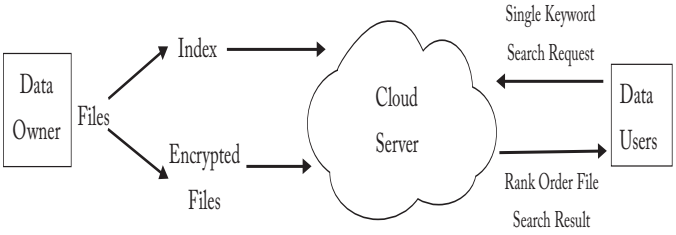


Fig. 1. Single Keyword Search Model

data hosting services in the context of Cloud Computing.

In [5], a proposal is made to integrate Order Preserving Symmetric Encryption (OPSE)[49] in order to prevent privacy breach of the data through knowledge of relevance score that may indicate frequency information. This is implemented by transforming the data to acquire a one-to-many order preserving mapping technique to protect sensitive data whilst also providing the facility of ranked keyword search over the data.

The key components of a basic encrypted data accommodating cloud is: Data Owner, Data User and Cloud Server (as shown in Figure 1). The Data Owner uploads the data content i.e., compilation of a number of files into the cloud whilst maintaining the privacy of the data through secured encryption. Although the data is encrypted the ability to search through the files is retained. To achieve this, the data owner outsources the collection of files and a secure searchable index to the cloud. The searchable index is prepared by the data owner and is a list of discrete keywords extracted from the compilation of files. When an approved user attempts to search for a keyword, a search request in the form of a trapdoor is generated in relation to the keyword. Upon receiving the request, the index stored on the cloud sever is searched and the set of files containing the keyword is returned to the user. As the concept of ranked keyword search goes, the set of files returned to the user are in a ranked order based on particular criteria (e.g. keyword frequency and other ways similarly). This return of files in a ranked manner is carried out in such a way that little or nothing is leaked from the sensitive file information. If the user mentions a value k then efficient file retrieval happens where top-k ranked files are returned to the user thus reducing the overhead cost. The outsourced file collection in cloud is provided with facilities to both access the file and to modify or revise frequently [50–52]

Wang et al., [5] have defined the problem of secure ranked keyword search over encrypted cloud data and provides effective protocol, that fulfils secure ranked search functionality with little relevance of score information leakage against keyword privacy. The secure ranked keyword search scheme is strong security compared to SSE. The Order Preserving Mapping (OPM) technique is used for ranking the searched file over encrypted cloud data. The OPM technique protects the sensitive score information from the cloud provider. This method is highly efficient but they lead to

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

collisions in the network. Computation cost increases when encrypted data is used in secure ranked keyword search.

Zerr et al., [16] have proposed Zerber+R - a ranking paradigm that allows privacy-preserving top-k retrieval from an outsourced inverted index. A suggestion for use of a relevance score transformation function is seen here, that causes relevance scores of various terms identical, so much so that even if data is stored on an untrusted server, information about the indexed data is not revealed. Tests on two real-world data sets concludes that Zerber+R causes cost effective usage of bandwidth and proposes retrieval properties that can be compared with that of an ordinary inverted index. The system paradigm provides support only for a sequence of single term top-k queries.

Cheng et al., [53] have proposed a novel VF-CAN indexing scheme which integrates Content Addressable Network (CAN) based on routing protocol and the Improved Vector Approximation file (VA-file) index in the cloud. VF-CAN scheme reduces the index storage space and improves query performance effectively. The protocol focuses on the establishment of the local index and publishing of global index, but does not concentrate on the index maintenance.

Boldyreva et al., [49] have developed Order-Preserving Symmetric Encryption (OPE), which permits efficient range queries on encrypted data. OPE is a deterministic encryption scheme whose encryption function preserves numerical ordering of the plaintexts. It provides the best-possible security under the order-preserving constraint applications based on pseudo randomness of an underlying block cipher. It leaks more information about the plaintexts than just their ordering.

Buyrukbilen et al.,[54] designed a Privacy-Preserving Ranked Search on Public-Key Encrypted Data. This scheme employs a sample indexing structure, homomorphic encryption and private information retrieval protocols to process queries in a privacy-preserving manner. The query response time reduces by several orders of magnitude but has storage overhead and increased computation cost.

Kuzu et al.,[55] have proposed an efficient scheme for similarity search over encrypted data. The Locality Sensitive Hashing (LSH) algorithm is used for fast near neighbor search in high dimensional spaces. LSH provides fast similarity search in the environment of encrypted data. The experimental datasets are not in large size.

Liu et al.,[56] have investigated the features of cloud storage services and proposed a Secure and Privacy Preserving Keyword Searching $(SPKS)$ scheme, that allows the Cloud Service Provider to return only files containing queried keywords specified by the users. It scales down both the computational and communication overhead in decryption for users, with the condition of preserving user data privacy and user querying privacy. The $SPKS$ scheme incurs storage overhead.

Cong et al., [48] designed a statistical measure approach known as Ranked Searchable Symmetric Encryption (RSSE) was introduced Information Retrieval and text mining to embed the weight information i.e., relevance score of each file while establishing the searchable index before outsourcing the encrypted file collection. A one-to-many Order-Preserving mapping technique integrated crypto primitive and Order-Preserving Symmetric Encryption $(OPSE)$.

Yuan *et al.,* [57] propose privacy-preserving search method over encrypted data in the cloud which is implemented using minhash functions. The main advantage of this method is the capability of multi keyword search in a single query. It uses two algorithms, Index Generation to generate the index and Document Retrieval to limit to retrieve only top documents. The second algorithm i,e Two-Server Secure Search and Document Retrieval searches keywords from the file server and provides adaptive semantic security.

V. MULTI KEYWORD SEARCH

Presently, many companies are embracing cloud computing. One of the major concerns regarding cloud computing has always been security. Encryption in cloud computing are still growing and unstable [58, 59]. There are different kinds of encryption schemes for securing data in the cloud and sometimes integrated within a system. Whenever an enterprise decides to move its applications to the cloud, it considers several aspects regarding security. And the main goal of encryption is to ensure that data accumulated in the cloud is protected against unauthorized access. Access to sensitive user data by other third parties is an interruption of privacy and it should never occur.

Encryption is not a guaranteed method and even if encryption is the most effective way of data protection, it has a certain drawbacks. Even if a cloud service provider provides encryption, the possibility of the keys being accessed by them is certain. The encryption keys should be managed securely in order for it to work effectively. And also when encrypted information is stored in the cloud, the keys for encryption should be kept separately, accessible by the end user. Key management includes creation, use, distribution, and destruction of encrypted keys and the toughest part is to manage in cryptosystems.

Some of the issues with encryption key management in the cloud are the possibilities of insider attack. Keys can be accessed or crept by employees without the knowledge of the end users. Encryption keys are susceptible to data breaches. If the same pair of keys is used on all your machines, then a hacker can gain access to all your cloud data and information just by compromising one of those machines. The keys for all accounts need to be managed properly. It is a challenge to catalog proper accounts with their respective keys in a fast

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

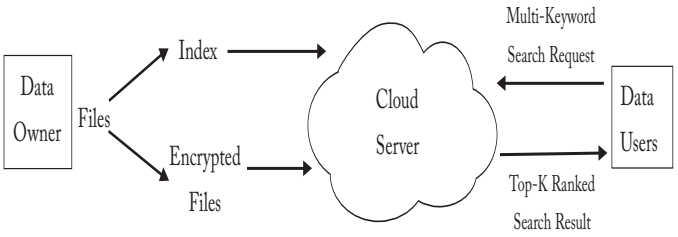


Fig. 2. Multi-Keyword Search Model

and effective manner.

Across the globe, enormous amount of outsourced information are stored and retrieved. In the process, various nuisances regarding data security arise during while providing retrieval and searching procedures. Bearing the various security concerns into consideration, the solution for protection is to upload data into the cloud server after encryption. Retrieval of precise information is challenging over encrypted data stored in cloud. To solve these problems, the paradigm Enhanced Multi-keyword top-k Search and Retrieval (EMTR) scheme is used, and this scheme presents good accuracy and efficiency. On the first note, the document is estimated efficiently using inverted indexing. Thereafter, the data user can send in any number of keyword search queries and for all the keywords that appear in the document, a relevance score is calculated for the document and the relevant document is retrieved from cloud storage as shown in Figure 2 [14]. A new rating method for extracting documents of highest rank from the set of data files that causes not able speedup over inverted subjective indexing.This analysis showcases that the suggested arrangement has successfully achieved high accuracy in developing the quality of search over encrypted data and efficient recovery of data at high speed to an extent but at the same time the efficiency has been negotiated [5, 48]. Searchable symmetric encryption (SSE) allows retrieval of encrypted data from cloud server [11–13, 28, 30, 33, 60–62]. More importantly, data security [63, 64] of searchable symmetric encryption (SSE) in cloud computing is the main focus. Issues in data privacy are withdrawn from the concept of similarity relevance and scheme robustness and later proven that the server side ranking made on the basis of order-preserving encryption (OPE) invariably breaches privacy of data stored in cloud. In order to resolve this issue, a two round searchable encryption (TRSE) strategy [65] that strengthens top-k multi-keyword search is put forward, where unique technologies like homomorphic encryption and vector space model is engaged. Vector space paradigm delivers ample search accuracy, and homomorphic encryption that enables users to be involved in the ranking while a bulk margin of the computing work is done on server-side by performing operations only on ciphertext. In this manner, breach of information is avoided and data security is preserved. Comprehensive security analysis and performance analysis depict that the paradigm put forth ensures high security and high efficiency.

Ning et al., [18] have proposed the model of privacy preserving Multi-keyword Ranked Search over Encrypted cloud data (MRSE) and have developed a set of strict privacy requisites for utilization system of secure cloud data. Searchable encryption technique aids in extracting data files from the cloud data centers. A secure k-Nearest Neighbor (kNN) technique was executed in the MRSE Scheme, where in two threat paradigms - Cipher-text Model and Background Model are examined with reference to factors as privacy and efficiency in multi-keyword ranked search.

Sun et al., [66] have proposed a secure and efficient dynamic multi-keyword ranked search(DMRS) scheme over encrypted data and it also renders aid for dynamic update operations such as deletion and insertion of data files. The index tree is built on the basis of vector space model which delivers flexible update operations. Cosine similarity measure provide precise search result that is ranked. In order to develop search efficiency and security Greedy depth-first traverse strategy algorithm and the known cipher-text threat model are used. Dynamic multi-keyword ranked search scheme have communication and storage overhead.

Orecik et al., [67] developed an efficient privacy-preserving search over encrypted cloud data that makes use of minhash functions in order to develop the accuracy rate. The benefits of this paradigm are multi keyword search in a single query and effective ranking capability established on the basis of term frequency and inverse document frequency. Secure Multi-Keyword Search Method is effective, efficient and privacy-preserving, yet the server computation is more.

Sun et al., [68] proposed a Verifiable Privacy-Preserving Multi-keyword Text Search (MTS) scheme with ranking based on similarity. The term frequency and vector space model with cosine similarity measure are used to build a search index, to obtain precise search results. A tree-based index structure and multi-dimensional (MD) algorithm provides improved search efficiency than linear search. Issues in security for two threat models are addressed i.e, known cipher-text model and known background model. The search process is verifiable in case the user wants to confirm authenticity of the returned search results. Performance is developed in terms of efficiency and privacy but computation complexity is high.

Yu et al., [65] have established a Two-Round Searchable Encryption (TRSE) scheme that provisions top-k multi keyword retrieval from the cloud storage system. Using SSE technique encrypted data is retrieved from the cloud. The vector space paradigm aids in providing sufficient search accuracy. Searched data is ranked by employing homomorphic encryption. The TRSE scheme ensures high security and high efficiency for small datasets. The data user first encrypts and thereafter sends the cipher-text to the cloud server. The scope of the cipher-text is too huge. Hence, the encrypted trapdoor size too is similarly quite large for communication. The computation overhead on server side is dependent on $TF - IDF$ weights to calculate relevance score for each

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

keyword search request.

Li et al., [69] have designed a scalable framework for Authorised Private Keyword Search $(APKS)$ over encrypted data in Cloud Computing based on Hierarchical Predicate Encryption (HPE). In this framework, every user obtained searching capabilities authorisation from Local Trusted Authority (LTA). $AKPS$ enabled multi-keyword search, allowing delegation and revocation of search capabilities. The major disadvantage is that $APKS$ does not prevent keyword attack.

Orencik et al., [70] developed a scheme based on Public Information Retrieval (PIR) that permits multi-keyword queries with ranking facility. Symmetric- key encryption method is used for file encryption rather than public-key encryption. An efficient ranking approach based on term frequency of keywords is utilized that returns highly relevant documents corresponding to submitted search words. This scheme increases the efficiency with the help of the blinded encryption technique in accessing the contents of the retrieved documents without leaking them to other parties.

Chen et al.,[71] developed an Efficient and secure Semantic Multi-Keyword Ranked Search over Encrypted Cloud data. Latent Semantic Analysis (LSA) is used to reveal the relationship between terms and documents. This scheme utilize k-Nearest Neighbor $(k-NN)$ and returns the files containing the terms semantically related to the query keyword. The experimental results of LSA are better than Multi-keyword Ranked Search over Encrypted Cloud Data$(MSRE)$ scheme. The matrix index file utilises large storage space compared to other schemes.

Li et al.,[72] have designed a well-organised multi-keyword ranked retrieval scheme with Johnson-Lindenstrauss (JL) transform over encrypted cloud data. The search technique having problem of low accuracy by directly using JL transform is overcome with Optimized Maximum Query method to build an efficient trapdoor. This scheme significantly reduces the space complexity but has computation overhead.

Zhang et al.,[73] addressed the issue of secure ranked multi-keyword search for multiple data owners and multiple data users in the cloud computing environment. The scheme enables authorised data users to achieve protected, convenient and efficient search over multiple data owner's data that is encrypted with different secret keys to rank the search results and preserve the privacy of relevance scores between keywords and files. A new Additive Order and Privacy Preserving Function family is proposed. This scheme suits for large scale datasets. Additional computation and storage cost is the overhead.

Li et al.,[74] presented a flexible Multi-Keyword Query Scheme, called $MKQE$, that supports partitioned matrices approach where trapdoor generation algorithm is designed to search queried keyword. Keyword weights and user access history is taken into consideration to display the queried keyword results. $MKQE$ scales down both the dictionary reconstruction overhead and the file index re-encryption time as new keywords and files are added.

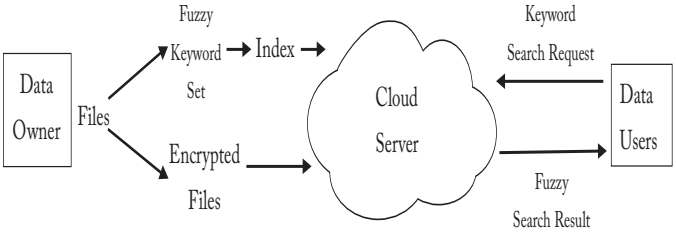


Fig. 3. Fuzzy Keyword Search Model

Sun *et al.,* [75] formulate a privacy-preserving multi-keyword text search (MTS) scheme along with similarity-based ranking. The proposed model develops search index based on TF-IDF and vector space model along with cosine similarity measure that supports multi-keyword search and ranking method and gives accurate search result. By using a tree-based index structure and adapting methods for multi-dimensional (MD) algorithm, it proves that the practical search efficiency is better than linear search. Two secure index schemes have been proposed to meet the privacy requirement.

VI. FUZZY KEYWORD SEARCH

A. Advantages over other models

Other models do not support on-the-fly search of the keyword and cannot tolerate minor inconsistencies and typographical errors in the keyword. Fuzzy keyword search greatly enhances system usability by returning the matching files when users searching inputs exactly match the predefined keywords or the closest possible matching files based on keyword similarity semantics, when exact match fails.

B. System Architecture

The basic fuzzy keyword search system is similar to the systems of traditional keyword search systems and is divided in three parts: Data owner, Cloud server and Data searcher as shown in Figure 3. When a user searches for a keyword, a search request is sent to the cloud server. The cloud server then examines its local index and sends the result to the user. The user upon receiving the query result can perform an array of operations such as download, edit or remove the file from server using a file identifier.

C. Wildcard-based fuzzy

An user may commit an error while typing the keyword i.e., a typographical error at any position index in the keyword. In the Wildcard based fuzzy system, for keyword search request that the user sends to the server, all possible modifications of the keyword are listed and searched. This is done irrespective

of the position where the operation occurs. Based on reviewing the list, in [76] the authors have proposed to use a wildcard to denote edit operations at the same position. The wildcard based fuzzy set edit's distance to solve the problems. For example, for the keyword CASTLE with the pre-set edit distance 1, its wildcard based fuzzy keyword set can be constructed as CASTLE, 1 = CASTLE, *CASTLE,*ASTLE, C*ASTLE, C*STLE, CA*STLE, CA*TLE...CASTLE*.

D. Gram – Based Technique

Gram- Based Technique is another efficient way of performing Fuzzy keyword search. In this search, a fuzzy set is built on basis of grams. Gram can be defined as substring of a given string which is used as a signature for an approximated efficient search. In general, grams is used while constructing an inverted index for approximate string search. In [76] the authors use gram for the matching purpose. It utilizes the fact that any primitive edit operation affects at most one specific character of the keyword, leaving all the remaining characters untouched. In other words, the relative order of the remaining characters after the primitive operations is always kept the same before the operations.
For example, the gram-based fuzzy set CASTLE, 1 for keyword CASTLE can be constructed as CASTLE, CSTLE, CATLE, CASLE, CASTE, CASTL, ASTLE.

Function Variable Meaning [77] Mkey - The master key which is owned by the server
FN - Name of the file
word - Word in the fuzzy keyword set
Key - The key which is used to encrypted the word
Fid - Generated by using the file name
f1(x) - One of the hash functions
Keyword Trapdoor - The encrypted keyword
f2(x) - The other hash function
Index Trapdoor - Generated by using keyword trapdoor and Fid
Bloom filter Index - The index build in bloom filter.

E. Detailed Procedures and System Security

The fuzzy keyword search runs as follows: A fuzzy keyword set is built by the data owner to basis of the wildcard based fuzzy method, the first step to a build a safe index. A Keyword Trapdoor, which is encrypted keyword, is created using the keywords in the fuzzy keyword set that is built by the data owner. Fid is generated by using a file name. The keyword trapdoor is then linked with Fid in order to generate an Index Trapdoor. The last in building a safe index is based on using the Bloom Filter to generate it. Now that the safe index is completely built, the data owner outsources the encrypted collection of files to the cloud server. The safe index and Fid that were generated are not outsourced to the cloud server as the case is in the traditional search techniques like single keyword search but is sent to the search server. When an authorized user searches the cloud server with a keyword and a key, a search request is generated by generating wildcard based fuzzy sets and Keyword trapdoors. Further the search request is sent to the search server. Upon receiving the request by the search server, the cloud server uses the Keyword =Trapdoors and Fid to generate the Index Trapdoor. This Index Trapdoor is then searched in the safe index stored in the search server. When receiving the result, user chooses one of the list and sends the retrieve request to the server [78] . Cloud server use the request information to search the encrypted file content, and sends it to the user. User decrypted the file content by using his own key.
This technique improvises on the existing keyword techniques with two layered security to ensure no sensitive information from being leaked to the server or user while performing the task of keyword search over the encrypted data files.

Sun et al., [79] established a Secure Ranked Semantic Keyword Search (RSS) over encrypted cloud data. A fuzzy solution supports for a search of a semantic keyword over cloud data that is encrypted. The owner of the data produces a piece of metadata for every file first, then the set of metadata that is encrypted and the compilation of data files are uploaded to the cloud. Accurately matched files that are semantically related to the queried keyword is returned by the semantic search. One to many order persevering encryption (OPSE) paradigm is utilized in order to obtain term frequency and relevance score. RSS scheme only works for single keyword query and the semantic keywords are not safe.

Xu et al., [80] formulated the problem with keyword privacy. The Public-key encryption with keyword search (PEKS) enhanced as Public-key encryption with fuzzy keyword search (PEFKS) is a Secure Scheme under Keyword Guessing Attack. The trapdoor is generated for both exact keyword and fuzzy keyword search. A generic transformation which converts any anonymous IBE scheme into a PEFKS scheme has been developed. The PEFKS are more secure compared to PEKS [81]. In both PEKS and PEFKS schemes, the search time is linear and depends on the size of the database. The focus is to reduce search time on encrypted cloud data.

Wang et al.,[82] proposed a novel Fuzzy Keyword Search Scheme $(F2SE)$ that uses fingerprint extraction and secure kNN encryption algorithm to achieve a top-k ranked fuzzy keyword search. It has a low storage overhead and practical searching time cost. The fingerprint extraction algorithm can be optimized to improve Searching Accuracy Rate and match it with other symbols or languages.

Wang et al.,[83] have integrated several innovative schemes to solve Privacy-preserving multi-keyword fuzzy search over encrypted data in the cloud. The fuzzy multi-keyword searches built the file index using LSH function in the Bloom filter. It gives a well-organised solution to the secure fuzzy keyword search. The Euclidean distance is implemented to capture the similarity between the keywords to calculate the similarity score to enable ranked result. It incurs computation and storage overhead.

Chuah *et al.,* [84] design a privacy-aware bedtree which supports fuzzy multi-keyword search. This work uses two algorithms VerifyED and Search tree which also performs fuzzy multi-keyword search. The bedtree index tree construction and storage cost of this method is efficient compared to the symbol-based trie-traversed based [62] and listing based approach [62].

Wang *et al.,* [85] develop a fuzzy keyword search method that considers typos while giving the input over encrypted cloud data utilization service. The encrypted index built using symmetric searchable encryption (SSE) [13] scheme does not support fuzzy keyword search. This work generates fuzzy keyword sets based on wildcard technique that can be applied to asymmetric search which tolerates typos.

Li *et al.,* [62] propose a solution which exploits edit distance quantifying the keyword similarity. The enumeration method used here to construct fuzzy keyword has large storage complexity which affects the data usability. This approach uses wildcard-based fuzzy set construction technique that is based on similarity metric of edit distance. When the exact match fails, this method returns the closest matching files.

Wang *et al.,* [86] proposes a verifiable fuzzy keyword search scheme based on the symbol-tree which supports the fuzzy keyword search and also performs the verifiability of the searching result. This scheme uses two algorithms i.e., Generate Fuzzy Set which generates the Fuzzy keyword set and Searching Tree which generates a set of file ids that helps in security and privacy-preserving.

Wang *et al.,* [83] proposes a scheme for multi keyword fuzzy search which searches by exploiting the locality-sensitive hashing technique. This method does not expand the index file, instead it searches fuzzy keyword matches using algorithmic design where the predefined dictionary is not required and using Symmetric cryptography. The index generation procedure is a single time computation and as the new keywords are inserted, the index generation time increases linearly. The search also happens in the encrypted index. Here, locality-sensitive hashing (LSH) method is used to construct the index of a file and this is efficient to search multiple keywords.

Jie *et al.,* [87] solves the problem of full scale fuzzy keyword set construction as per the keyword given as input that improves the system usability and retrieves right keywords. This scheme achieves completeness in fuzzy keyword set construction as it uses the wildcard as well as dictionary based schemes to remove unwanted keywords and results in relatively small constructed index.

Wang *et al.,* [88] propose a novel fuzzy keyword scheme named F2SE to achieve a top-k ranked fuzzy keyword search which uses finger print extraction as well as secure kNN encryption. This method achieves similarity search for top-k ranked keywords and special string first match. It uses Keyword Fingerprint Extraction that converts a string into a fingerprint vector. The kNN encryption provides two tier of protection for keywords being searched.

Xu *et al.,* [80] propose a method called Public-key Encryption with Fuzzy keyword search (PEFKS) where each keyword refers to an exact keyword and fuzzy keyword search trapdoor. It takes linear time to store the searchable cipher texts as keywords and resists keyword guess attack

Bijral *et al.,* [89] design a method for effective fuzzy keyword search using B-tree and privacy preserving. The B Tree search algorithm uses inverted index and fully inverted index. This method shows that wildcard based search has more keywords than fuzzy keywords in Dictionary based search. It is observed that DFS method is efficient with respect to time.

Shekokar *et al.,* [90] propose a wildcard method for advanced fuzzy keyword search where it returns the matching files in which search keywords match exactly with predefined keywords. If the exact match fails, it returns the closest possible matching files based on similarity keyword semantics. Data is stored securely using Advanced Encryption Standard method and retrieved from the encrypted cloud by performing fuzzy keyword search. It also supports privacy-preserving fuzzy keyword search to achieve effective usage of encrypted data stored remotely in cloud.

Balamuralikrishna *et al.,* [91] propose a fuzzy keyword search on encrypted cloud data along with maintaining privacy of the searched keyword. Wildcard and gram based techniques use string matching algorithm.Trie traverse tree structure has been constructed are transformed from the resulted fuzzy keyword sets.

Zhou *et al.,* [92] propose a scheme to generate fuzzy keyword search over encrypted cloud data which uses K-grams index to return fuzzy results where keywords are searched as wildcard queries on plain-text files. Weighted ranking algorithm has been used to compute the weight of each word in the fuzzy set.

VII. Conjunctive Keyword Search

Nowadays we use a lot of mobile devices. Due to wireless networking, we have gained faster access to a large amount of data. Most of the time we do not store the data on the device, instead we store it online in data servers. If we are using an untrusted server we encrypt our data and then store it.

A user encounting a problem when he wants to find a content of a particular document in a large database of encrypted documents. The encryption makes it hard to search the keyword in every document. If a user is actually interested in documents containing each of several keywords [93], the user must either give the server capabilities for each of the keywords individually or rely on intersection calculation or

the user may store additional information on the server to facilitate such searches [11, 12, 28, 94–100].

Both of the above methods are undesirable, as they incur in large overhead and massive blow up of data. Thus we illustrate two major requirements of conjunctive search protocols: security and efficiency. Conjunctive keyword [97, 100] can be compared to searching all the data containing each of the keyword by application of single keyword search. In short, conjunctive keyword search can be described as an implementation of multiple single keyword searches, overlapping data is obtained and results are returned to user.

The three basic components of conjunctive keyword search scheme are: data owner, storage system and data user. The data owner uploads the encrypted data into the storage system and the data user searches the encrypted data through conjunctive keyword search.
Many schemes are available in a untrusted web-based storage system [25, 26, 101]. A user may store his personal data in encrypted form (searchable encryption e.g., [11, 12, 28, 30, 102] over the server and perform a search on the data using necessary keywords.

Since multiple singe keyword searches need to be performed, the cost associated with conjunctive keyword search is high and due to multiple, duplicate comparisons and searches the server is often redundant. We define a security model which states that the server should learn nothing other than the result of conjunctive query i.e., server should not be able to generate new capabilities from existing capabilities, other than logical extensions.

A. Model

Suppose we have a user who stores 'n' encrypted documents on an untrusted server. We assume there are 'm' keywords/ fields associated with each document.

Ex:If we have encrypted emails, for simplicity we define 4 keywords -"From","to","Date" and "Subject" and make assumptions that:

- Same keyword never appears in two different keyword fields. We associate keywords with the name of field they belong to "From : X" belongs to field "from".
- Every keyword is defined i.e., we fill up missing values of a field with a constant or null. For our discussion, we refer documents with m keywords by $DiWi, j$ is the keyword in document Di in jth keyword field. The conjunctive keyword search consists of 5 algorithms, the first four are randomized. A parameter generation algorithm, which takes as input a security parameter (say k) and outputs public system parameters(p). These parameters facilitate the search operation and keyword match between user and server. (as they have to search based on common parameter.) A key generation algorithm that outputs a set of secret keys K for the user. The keys are generated based on the security parameter of first algorithm.

After agreeing on the parameters (p) and generating keys(K) we choose the document Di with keywords $(Wi, 1, Wi, 2, \ldots\ldots, Wi, m)$ for encryption algorithm. The output of this algorithm is a vector of keywords (encrypted). Now the user end of search operation is over. We need the server to start the search. We define an algorithm for generating capabilities. It takes parameters(p), keys(k), keywords indices(j) and keyword values(Wj). it gives the output value Cap i.e., Capability to search the keywords Wj.

Finally we define a verification algorithm, that takes parameter(p), capability(Cap) and the output of Encryption algorithm. It basically compares the output of Encryption algorithm and capability based on parameter(p). Returns either true or false.

Golle et al., [97] have proposed a security model for Conjunctive Keyword search over Encrypted cloud data and the schemes allow conjunctive keyword queries to conduct secure search. The Decisional Diffe-Hellman (DDH) and standard hardness assumptions are used in communication cost. DDH assumptions have linear communication cost, which incurs the cost before the conjunction query occurs. The new hardness assumptions are having communication cost depending on the number of keyword fields and security. In this scheme, they partially solve the problem for Boolean search and also need a solution for secure disjunctive keyword search on both across and with the keyword fields. The Conjunctive Keyword search is not secure because the server learns about keyword fields.

VIII. Synonym Keyword Search

Cloud computing technology is becoming more mature and increasingly popular making many organizations protect sensitive and private data by encrypting the data before outsourcing making the traditional and efficient plaintext keyword search technique inept. In recent years, consumer-centric cloud computing models have emerged as the development of smart automated devices combined with advance cloud computing technologies. The customers are delivered with a variety of cloud services with the basis that an efficient and effective cloud search service is achieved. Contributions have been made mainly in two aspects, synonym-based search to support synonym query and multi-keyword ranked search to achieve accurate search results.

In a real search scenario, the cloud customers' searching may be synonyms of the predefined keywords with the pile of a search. The cloud servers now have to support a keyword search feature for these encrypted files. In some embodiments, the search tool is employed in the context of a web browser for searching web pages. In other examples, search terms are made noticeable by either highlighting them or by color coding the synonyms on a web page to provide the user with words that have been found through

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

the search. Usually searchable encryption schemes support exact keyword matches only but users sometimes use slightly different formats or make errors thus making fuzzy keyword search an alternate useful feature as the existing multi-keyword search scheme does not provide efficient incremental updates. Even then fuzzy keyword search solution have some limitations as it consumes large storage volume since every fuzzy keyword is inserted as a new leaf node in the index tree and this solution does not support multi-keyword search.

The working is very unique for each synonym keyword. When we use a keyword to describe our search in the library catalog or a database we retrieve records containing those search terms. But a major disadvantage of this keyword search is that it does not take into account the meaning of the keywords used and ambiguous words lead to retrieval of irrelevant records.

Yet another versatile tool is Public-key Encryption with keyword Search (PEKS). This allows a third party which knows the search trapdoor of the keyword used to search for the encrypted documents containing that keyword without decrypting the documents. This can lead to compromised privacy invasion by malicious third party, hence we try solving this by introducing public-key encryption with fuzzy keyword search (PEFKS). Here individual keyword is made to correspond to not just an exact keyword search trapdoor but a fuzzy keyword search trapdoor also and two or more keywords can share the same fuzzy keyword trapdoor. When we search for encrypted documents encompassing that specific keyword the third party is provided with the fuzzy keyword search trapdoor only.

With the constant raise in the number of data users and documents in the cloud, it is necessary that these servers allow multiple keywords in the search request and help retrieve documents in the order of the significance to these keywords. We also try to define and solve problems pertaining privacy by preserving multi-keyword ranked search over encrypted data in cloud computing (MRSE)and to establish a strict set of privacy conditions to ensure a secure cloud data utilization system.

Khan *et al.,* [103] formulate secure rank search of fuzzy multi-keyword which returns the matching files when input keyword exactly matches the predefined keyword. If it fails to match exact files, it returns the possible keywords from the dictionary based on similarity semantics. This method uses two algorithms Build-Index (RFMS) and Fuzzy-search (RFMS). As the encryption files contains special characters, the dictionary attack is not possible. Each element of the trapdoor is unique, it is one-to-many mapping between plaintext keyword and its cipher text which contains special characters that results in enhanced security.

Tuo *et al.,* [104] propose a semantically secure fuzzy keyword search scheme using the bloom filter which translates the keyword into attribute set and uses independent hash functions to bilinear map the elements to some random number. This method extends from fuzzy identity encryption scheme to fuzzy keyword search scheme.

Fu *et al.,* [105] propose a semantic search scheme which returns keyword-based proper match and also a keyword based semantic match and the search result are verifiable. Whenever the keyword is submitted for search, it builds the term similarity tree and then shortest path and similarity between keywords are calculated. The user can cross check the correctness and completeness of the result obtained. It uses the hash function query to achieve index privacy.

Fu *et al.,* [106] propose a synonym-based multi-keyword ranked search which supports synonym query as well as multi-keyword search to achieve exact match. Here Vector space model has been used to construct the index for document. The balanced binary tree-based index structure is used for searching the keyword. Synonyms of predefined keywords are also searched in this method.

Fu *et al.,* [107] propose a semantic keyword-based search scheme and privacy preserving. The stemming algorithm is used to construct the stem set and this reduces the dimension of index. A symbol based trie is adopted for construction of index which improves the search efficiency.

Ko *et al.,* [108] propose a semantical scheme which searches data on mobile devices where a user query is translated into a query graph and then retrieved.This method uses the algorithm for finding answer graphs from a query graph, then the answer graph is translated to SQL statements by traversing answer graphs to obtain the result from the Database. It overcomes the limitations of keyword based full text search.

Chinnasamy *et al.,* [109] propose semantic secure keyword based search scheme(ESSKS) that retrieves exact details needed by the user and ensures that same keyword does not always produce similar results. This method addresses the problem of data integrity while transferring data from the user to cloud and vice-versa. The user has to encrypt a file using secure symmetric encryption that reduces the search time of the keyword and does not allow unauthorized users to access the data.

Moh *et al.,* [110] propose a semantic search with three different schemes like Synonym-based, Wikipedia based and Wikipedia based synonym keyword search. This scheme uses Data encryption standard algorithm for symmetric key encryption as well as decryption. The search results are in the form of similarity score on the data file. This scheme performs better than Wildcard-based Fuzzy Set Construction scheme with respect to storage requirements, performance and the search result in terms of preserving data security and maintaining privacy.

Xu *et al.,* [111] propose a scheme for privacy preserving ranked fuzzy keyword which returns the matching files in ranked order based on semantics of keyword similarity. A

dictionary-based fuzzy set is constructed for fuzzy keywords and uses coordinate matching which returns all the matches as possible so that the data files obtained is relevant. As one-to-many order preserving mapping scheme is used to build the inverted index and for searching.

IX. Similarity Search Methods

The applications where require similarity search is used in the field of medicine Magnetic Resonance Images (MRI) scans; in finance sector, where we can see stocks with similar time behaviour; in digital library with respect to text retrieval, multimedia information retrieval and content based retrieval [112–114]. The standard search techniques lies in the core of the similarity search; there exist an infinite number of (dis)similarity functions that can be used with a wide variety of data types. While searching, the similarity query typically contains an example object (query object) and the search should return the data objects that are the "most similar" to the example according to the specified function. Here we mainly focus on the similarity search based on the metric space model. The metric space is an ordered pair M = (M, d), where M is a domain of data objects and d is a total distance function $d : M \times M \rightarrow R$ satisfying metric postulates of non-negativity, identity, symmetry, and triangle inequality. The set of indexed objects $X \subseteq M$ is typically searched by the query-by-example paradigm, for instance by the range query $Range(q, r) = o \in X \mid d(q, o) \leq r, q \in M$ or by the nearest neighbors query $k - NN(q)$ covering k objects from X with the smallest distances to given $q \in M$ [115].

Shortcomings and solutions of Existing Similarity Search Methods are listed below:

A. Brute force secure solution

The objects are uploaded to the server only after the data owner encrypts them by applying a symmetric key. Actual results are calculated after the client places a query at the query time and the encrypted objects are downloaded from the server. The method is absolutely secure due to the use of encryption, but there is an increase in the communication cost due to the downloading of all the objects, including the data objects not concerned with the query. Hence the method is not suitable for the present day needs.

B. Anonymization-Based Solution

Data privacy for the anonymization-based solution can be achieved by the k-anonymity and not by the encryption. Here we assume that there exists k number of objects and the generalization happens in such a way that every object that is generalized cannot be discriminated from other $k - 1$ objects which are generalized. Hence by following generalization scheme, the transformed objects ranking can be confused. The confusion created help to represent that the $k - 1$ objects has the same rank as the transformed object of the actual nearest neighbor. The clustering-based anonymization technique of Aggarwal et al. [116] can be applied for arbitrary metric space data. Each bucket is represented by Minimum Bounding Sphere (MBS).

The anonymization-based solution has a limitation; the MBRs/MBSs (Maximum/Minimum bounding Rectangle/Sphere) may contain more empty space as they are dealing with multidimensional data, causing them to retrieve a large number of buckets. On the contrary, in the proposed method the anchors are changed to IDs and distance information is changed to numbers and only what is concerned with the query are retrieved.

C. Indexing and NN search on metric space

1) R*-tree: It is capable of accessing multidimensional points and spatial data. Queries and operations, such as map overlay, rectangles and multidimensional points can be easily dealt using R*-tree. Data containing many dimensions can be stored by using X-tree whose indexing structure is based on R-tree. R tree and X tree are renowned disk based indexes for multi dimensional objects. Data objects which are complex ex: time series, cannot properly match up to by the co-ordinate values.
2) Vp tree: Data in the metric space is isolated by choosing a position in the space, called Vantage Point (Vp). The data point are divide into two partitions, those which are at close proximity to the server and those which are farther away from the Vp. Feature vector of fixed dimensions can easily be represented by index objects.
3) Mvp: Objects can be indexed using an abstract data structure called Multi vantage point(Mvp). Similarity query can be applied on large metric spaces using distance based index which is constructed using Mvp-tree. Similarity partitioning strategy are made use in the Mvp-tree and Vp-tree where they do not use pre computed distances and use only one Vantage point.

Similarity search on metric data can be used by various applications in the field of science and business. The confidential data when outsourced must be protected against leaks or attacks at the same time be made available to the authorized client or client groups.

Space program collects scientifically important and extraordinary data. Such data needs to be protected when outsourced to a third party server, to make sure that the funds of the scientific groups do not go in vain. For example, time series data is collected from sensors. They are used to study the density in the atmosphere. These data is made available to the general public only after the scientists involved in setting up the instruments have analyzed them. Access is limited to authorized scientists at an early stage because of the huge amounts of funds invested and the effort which has gone in setting up the instruments, building, testing and verifying the results. The authorized scientists analyze the data by collecting alike patterns in hourly or daily basis which point out interesting events. In this set-up, vectors of values are place in chronological order to represent the time series data. At the query time, client provides an example time series x and would want to retrieve those which are most similar to the

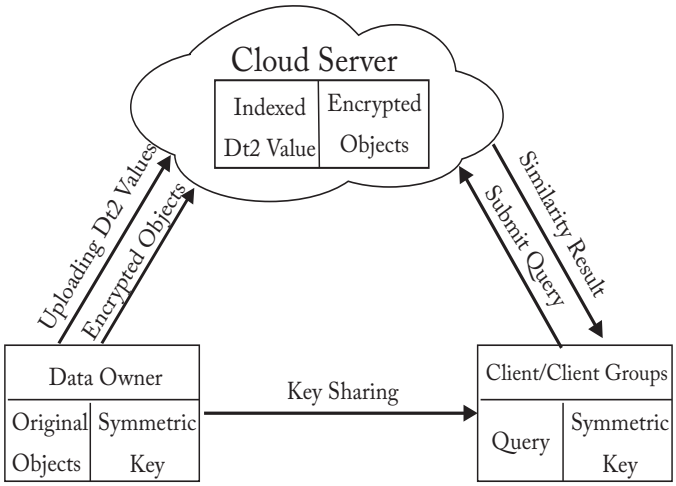


Fig. 4. Similarity Search Model

time series x, the search system responds with y, by providing those time series which are most similar and less distance to x.

Biologists analyze DNA microarray data to analyze the working of genes or gene groups. Gene examples are made to go through various examples and various conditions and results are stored in a matrix format and it is called DNA micro array. Row in the matrix represent gene examples and columns represent conditions. Genes that follow the identical expression pattern have similar characteristics. For a given gene, its expression values appear as a query vector. The scientist put forward query to the database to categorize the genes which are very similar to each other to a particular patterns and therefore can be linked to these genes.

D. SYSTEM MODEL

The system model has 3 elements - data owner, client and server as shown in Figure 4. Data owner desires that the data be made available to the authorized clients, but to do so he has to host his data on the server which he does not trust. Hence the data owner encrypts the data and uploads to the server. He applies a standard encryption method (symmetric key encryption algorithm e.g., AES) on the data set of original objects; this results in encrypted objects. These encrypted objects along with their Ids are uploaded to the server and stored in a relational table (or in the file system).

The original data objects can be anything such as time series, graphs, strings, medical data, and scientific data. Search service is outsourced by the data owner to the server. Data owner shares a secret key with his clients. The clients having the secret key are authorized to use the search service. The client issues a query and must have the key as a proof of its authorization to the server. The server processes the query and returns the similarity result to the client.

The 3 elements of the system model are:

1) Data Owner: Owns the data and allows outsourcing of the search service.
2) Server: Server(s) is a third party similarity cloud used to store the data. The data owner does not trust the server (server can be attacked and data from it leaks to an attacker).
3) Authorized Client: Access the data by using the secret key and retrieves the data needed by using the search service.

E. M-tree

Metric domain can be visualized in a distance based index structure called M-tree. M-tree is more competent than R*-tree in terms of input /output cost and distance optimization. They perform well in high dimensional space. The disadvantage is that the objects are entered randomly; the parent node is located by travelling from the root of the tree until the node itself is found.

F. METRIC SPACE

Indexing a metric space means to provide a well-organized support for dealing with similarity queries. The objective of the query is to get DB objects which are “similar” to a query object, and where the similarity or dissimilarity of the objects is calculated by a particular metric distance function d. In principle, there are three basic types of similarity queries: the range query, k nearest neighbors query, Range k-NN query

1) Range Query: Given a query object Q which is one of the query object of the domain and a search distance which is maximum represented by $r(Q)$, the range query represented as range$(Q, r(Q))$ and selects $d(Oi, Q) \leq r(Q)$ where Oi is indexed object.
2) k-nearest neighbors (k-NN) : Given a query object Q which is one of the query object of the domain and an integer k has a value always greater than 1, k-NN query selects the k indexed objects which have the shortest distance from Q.
3) Range k-NN query: the intersection of the prior two types, i.e Range query and k-NN of queries is called Range k-NN query.

G. General Scheme of Similarity Search

The perception of similarity search can be applied to a wide range of data types together with a number of diverse similarity functions. Metric space is suitable for many applications since it is modelled in such way that data is based on mathematical abstraction. The similarity search is a very tedious and resource demanding process. The technologies that support these processes are very complex. Hence, the inspiration to develop a common method that would provide the similarity search arises. This can be used as a service to make it effortlessly available to the users.

Cloud services have gained huge popularity and therefore we move toward to outsource this task of similarity search to the cloud environment. This service is made available in Software as a Service (SaaS) manner. These services would provide many advantages for the data owners, such as very less initial investments, less storage costs and a very good scalability. A

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

widely studied topic in the context of classic databases is to ensure privacy of outsourced search service. Data owners can encrypt the data in such a way that it is feasible to carry out selective data retrieval (search) over the encrypted collection and at the same time data privacy is ensured.
General scheme of outsourced similarity search can be explained as follows:

1) MS objects are created by the data owner from the raw data.
2) The data owner sends the raw data to storage and the metric space data to be indexed on a similarity cloud.
3) An authoritative client can query the similarity cloud to retrieve IDs of the relevant objects which refers to original data objects. This can be consequently received back from the raw data storage.

Most of the time, security constraints gets tradeoff when compared to the efficiency. Server should have adequate information about the data if it has to be processed and computed. This would help in achieving better task efficiency. Therefore, correct balance needs to be maintained between the security and efficiency such that it can be applied to a specific application.

The three transformation methods of Similarity Search are described below:

H. Encrypted Hierarchical Index Search (EHI)

It is a hierarchical indexing structure which is built on the Metric Space (MS) object data set; these nodes are encrypted using a symmetric key algorithm and address of the root node is made public. Mindistance and Maxdistance functions are used for the data transformation which makes the algorithm secure. The search service is at the client side. The client request for nodes decrypts them and applies the search function on these nodes; a new set of nodes are requested again from the server until the required result is found. There exists a considerable traffic between the server and the client due to multiple communication round trips; a reason for increased communication cost. There is an additional overhead on the client due to the search procedure which takes place on its side. The method suffers from relatively very low search efficiency.
It is a client algorithm which performs NN search technique on data which is indexed, encrypted and arranged in a hierarchical fashion. This scheme offers data privacy for the data owner but at the cost of multiple communication round trips which would occur during processing of the query.

I. Metric Preserving Transformation (MPT)

The method is used for evaluating the NN query. MPT needs only two communication rounds during the phase in which client submits the query, unlike the EHI. The fundamental ideology of MPT method is to select a subset which is small from the data set P. This subset consists of anchor objects; assign each of the objects of data set P to its nearby anchor. Distance $dist(ai, p)$ is calculated from its anchor ai for each object p. The distance obtained by doing so is applied to an order-preserving encryption function (OPE). The server stores these order-preserving encrypted distances which are used to find and process NN queries.

J. Flexible Distance-Based Hashing (FDH)

In the build phase, a set of n anchor objects is chosen and each anchor ai is assigned a range ri. Then, for every object, a bitmap of length n is created; the ith bit of this bitmap equals to zero if $d(ai, o) \leq ri$, otherwise it equals to 1. The objects are encrypted and are stored on the server along with their bitmap representation. The bitmap is used as the data transformation function. The client is given the liberty to select and vary the θ value; the server returns θ number of objects that are closest to the bit map representation of the query. It accomplishes the best communication cost because of the very compact bit map representation; the drawback being it innately supports only approximate search queries.

Flexible Distance-Based Hashing, for finding the NN query. A constant-sized candidate set is returned from the server is the key advantage of this method. The final results are obtained by refining the above candidate set. The final result is very near to actual NN in practice because the FDH method does not give any guarantee to return the exact result. To increasing the accuracy of a query result, the client needs to give a parameter θ which is an integer value. The accuracy of a query result is increased without rebuilding the transformed data stored at the server due to the specification of the parameter θ.

Advantages and Disadvantages

1) Communication cost
   a) BRUTE and ANONY which are the basic methods have more communication cost in comparison to the other methods.
   b) In EHI, the client carry out the search service and hence has a reasonable communication cost
   c) Methods like MPT and FDH carry out search service at the server and therefore the communication cost is diminished greatly.
   d) FDH is the method which accomplishes best in terms of communication cost and thereby the best performer among all the methods.
2) Scalability
   a) FDH is the method which scales exceptionally well in requisites of communication cost.
3) Effects of selection parameter θ
   a) In MPT, when θ is increased, the communication costs is lower.
   b) In FDH, when θ is increased, increases the communication cost due to a single round of communication.

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

c) With more anchors, communication cost is reduced because anchors provide more detail information on the location of the objects.

Binolin et al., [112] formulate similarity search for content based image retrieval involves dynamic similarity querying on metric data from segmented and extracted texture features database. The technique involves segmentation and feature extraction on outsourced lungs images and stored as metric data in database. The proposed similarity search of outsourced lungs images is retrieved with good query efficiency and reduced time on retrieval based on good distance measure. Proficiency and precise processing of the similarity query is rendered by the proposed methods.

Yiu et al., [117] present methods which transform data, only upon the transformed data similarity queries can be presented to the service provider. EHI (Encrypted Hierarchical Index), MPT (Metric Preserving Transformation), FDH (Flexible Distance-based Hashing) algorithms are used for transformation. The techniques were experimented with real data sets which provide Proficiency and precise processing of the similarity query. The proposed solutions have shifted search functionality to the server which reduces the communication rounds. FDH do not guarantee to fetch the exact result but ceases just one round of communication.

Ciaccia et al., [118] Generic metric space consist of large data sets, M tree helps to search and organize such large data sets. The algorithms used for such purpose are similarity range and k-Nearest Neighbor (k-NN), insertion and split operations keep the $M-tree$ balanced. $M-tree$ is more competent than R*-tree in terms of input /output cost and distance optimization. They perform well in high dimensional space; a setback is about choosing the correct policy for the split operation in order to get good performance.

Khoshgozaran et al., [119] approach to map the static and dynamic objects after applying one way transformation to another space and the query can be resolved blindly in the transformed space(Hilbert space) which is used to realize k-NN query. The complexity for calculating the k-NN is reduced compared to other traditional approach. U-anonymity and A-anonymity are two new privacy methods metrics which are efficient and more generalized and perform K-NN query in the original space to get a close approximation of the result. Hilbert curve reduces the dimensions and thus there is a risk of ruling out some potential *NN*(Nearest Neighbour).

Connor et al., [120] fostered *K-NN* graph construction using Morton ordering. Linear list of numbers is achieved as a result of applying Morton ordering on to *N*-Dimensional space. Algorithm performs best on point sets that use integer coordinates and is also viable using floating point coordinates. The algorithm favours faster construction of *K-NN* graphs and uses less space. As more processing power becomes available, a drawback could arise with respect to scalability of cache efficiency.

Wong et al., [121] formulate a new Asymmetric Scalar-Product-Preserving Encryption (ASPE) aims to support k-NN computation on encrypted data by constructing secure schemes. Advanced APSE has higher encryption time and query processing time compared to APSE. APSE profits by giving a very low cost and resist different overhead cost at various level of practical attacks by considering different background knowledge. A downside is encountered due to introduction of additional dimensions which give a higher overhead to advance APSE.

Bozkaya et al., [122] introduce MVP-tree (Multi-Vantage Point tree) to answer similarity based queries efficiently for metric spaces of high-dimensionality. MVP-trees perform well over VP-trees, especially for small query which spans up to 80%. MVP-tree is created in a top down fashion on a given set of data points and hence guarantees a balanced tree. The setback is to have the update operations such as insertions and deletions at a reasonable cost.

Agrawal et al., [123] have come up with Order-preserving encryption scheme. OPSE allows the encrypted data to be directly compared which acts as a new scheme for numeric data. Here the answer tuples are not missed and at the same time query results do not contain any false positive. Encryption of the values need not be changed since OPSE can handle updates gracefully and changes can be done easily. Few issues such as query optimization, key management and impact of encryption on query plans need to be overcome.

Kozak et al., [124] propose two new similarity indexes that are apt for search systems outsourced in a cloud and also guarantee data privacy. The first proposed technique EM-Index is more on the efficiency while the second technique DSH Index shifts this balance more to the privacy. The techniques mentioned provide better privacy guarantees for similarity cloud solution. EM-Index proves profitable by supporting precise evaluation of the range queries and efficient update operations while DSH guarantee higher privacy level.

Kuzu et al., [55] advance towards similarity search over encrypted data which has been accomplished by using the Locality Sensitive Hashing (LSH) algorithm. Functionality for a fast similarity search is provided and at the same time confidentiality of the sensitive data is not sacrificed. Typographical errors both in the queries and the data sources are considered leniently by a real world application which allows keyword search.

Zhu et al., [125] propose similarity search over encrypted images using Locality Sensitive Hashing (LSH) algorithm. The scheme enables content based retrieval service to the cloud without leaking the actual content of the image database to the cloud and also allows the users to outsource their images. LSH achieves reduced computational cost over similarity search. LSH improves search efficiency and reduces the search time. A drawback of LSH scheme is it does not improve precision.

Chen et al., [126] develop a strategy that uses data splitting algorithm that splits data d into k sections. High data security is achieved by simplifying k equation solutions. The increased number of data blocks increases the decoding difficulty by ten times. Data splitting algorithm ensures high data security, guarantees highly reliable data. The secure strategy, however, also has its weakness, such as high data redundancy.

Lu et al., [127] propose an efficient solution, called the Ordered VA-File (OVA-File) based on the VA-File which address the problem of content-based video indexing. A high query result is obtained in the proposed method when compared to two other methods VA-File-based method and $iDistance$. The OVA-File has some merits such as the query response time of OVA-File would be exceptionally low than that of VA-File and any query search algorithm based on VA-File is applicable to OVA-File.

Cui et al., [128] have come up with high-dimensional query in main memory setting which is an indexing structure called $\triangle$-tree. The algorithms used for this purpose are *K-NN* search algorithm for $\triangle$-tree and Range query algorithm for $\triangle$-tree. The proposed methods which lower the cost of computing and cache misses because the search process can reduce the space to be searched efficiently. They capture the feature of the data set and, therefore shrink the search space, advantage of considering top-down clustering scheme. The disadvantage is to decide on the number of dimensions required; because it is difficult to identify the dimensions to be considered to obtain the best performance.

Xia et al., [42] propose a scheme for similar search on encrypted images based on a secure transformation method. Search of similar images are done using the trap door generation method. The transformation on features protects the information about features, and does not mortify the result accuracy and also keeps the confidentiality of the data intact. The demerits of the scheme would be that it suffers from the statistic attacks and hence not an optimal one.

Wang et al., [129] develop a semantic expansion based similar search over encrypted cloud data. The algorithm used for this purpose is One to many order preserving encryption. The proposed scheme performs fast and efficient search for first 100 results. The advantage of the scheme is that it returns the exact matched files; in addition it returns also files that are semantically related to the key word. The short- comings of the system are that efficient crypto techniques still needs to be used to protect the semantic information of the files.

Jang et al., [130] propose the use of network distances among POIs (Point Of Interest) that create a transformed database from the original database. A spatial database encryption scheme is used to do so. MPT algorithm for finding K-NN is used for this purpose. The performance is enhanced by reducing the query processing time and has the benefit of reducing the search range. To overcome the disadvantage of the proposed scheme and to improve the performance of the method we need to reduce the size of the returned candidate set and a study on pruning technique is required.

Hjaltason et al., [131] present a method which works on distances and later the search techniques can be applied over it. They represent algorithms for ordinary category of queries that work on a random search patterns. The algorithms are Range search algorithm and Incremental ranking. They do well by making the hierarchical search easy and using the $M-tree$ for the indexing purpose. The proposed scheme is beneficial by having $M-tree$ that is most suitable for dynamic situation which involves large amount of data.

Tsymbal et al., [132] share their experience gained by translation of a similarity search-based clinical decision support system; Case Reasoner from a standalone desktop application into a mobile Web-based solution. They have come up with advanced similarity search and case retrieval-based solutions with lower computational complexity. The Case Reasoner module is proved advantageous with the intention to help not only with more intuitive navigation within big data, but also with the transparency and explanation of suggested decisions.

Athitsos et al., [133] propose methods to efficiently estimate the nearest neighbor which could be done by indexing spaces with arbitrary distance measures. The Distance Based Hashing algorithm is used to serve the purpose. Their method produces good trade-offs between accuracy and efficiency, and performs better when compared to VP-trees. This has been proved by conducting experiments on several real-world data sets. An upside of using DBH is that it can be constructed in any space because it uses a family of binary hashing functions that is distance-based.

Hajebi et al., [134] introduce a novel algorithm that helps in resolving the nearest neighbor search problem by performing hill-climbing on a K-NN graph. The Graph Nearest Neighbour Search (GNNS) algorithm is used to solve the K-NN Search problems. The GNNS method outperforms both the KD-tree and LSH algorithms. The costly offline construction of the K-NN graph emerges as the drawback of this method.

Popivanov et al., [135] permit proficient similarity search over time-series where data is of high-dimensions and considers the use of wavelet transformations for reducing the dimensions. They perform well when a relatively long (12-16) filter length is considered which gives the highest precision for this application. Similarity search makes use of all wavelets with a steady restoration that is a class of bi-orthonormal wavelets. The disadvantage is that they have not yet fully supported the techniques which reduce the dimension of the time series data.

Amato et al., [136] present an approach to approximate

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

similarity search in metric spaces based on a space transformation that relies on the idea of perspective from a data point. They use inverted files to achieve efficient and effective approximate similarity search. Reference objects should be selected carefully such that the performance increases. Metric spaces can be modelled using similarity search where proposed method can be applied to any such application. It is trivial to optimally choose the reference points.

Gil-Costa et al., [137] propose a method which works on indexed metric space which can be balanced well and query processing can happen in a parallel manner. Scheduling algorithm is applied onto a global index which gets evenly distributed on the processors and hence helps to achieve good performance. The query scheduling algorithm is proved advantageous by solving the problem of performance degradation in global indexing. The same super steps can be imposed in the Sync mode that is trivial to the broker simulation and actual processors and applied on the scheduling algorithm.

Yoon et al., [138] propose a new transformation scheme based on a line symmetric transformation (LST). They use Data Distribution Transformation algorithm. The performance analysis shows that their scheme is protected more strongly against different attack models than the existing ones. It performs both LST-based data distribution and error injection transformation for preventing proximity attack effectively. Transformation scheme can be extended to support more spatial query types, such as skyline spatial queries and spatial path queries.

X. Data Sharing Schemes

Cloud Computing [139] accelerates development of data storage, processing and distribution. Storing data in a public cloud server provides a great deal of benefits to the data owners. When data is shared and stored in a cloud for a group of members, it serves both pros and cons. The advantage being it evades the user from the difficulty of local data storage and maintaining it and also provides an easier platform for sharing of the data amongst the members. This reduces the cost and time required to manually exchange data. However, the considerable threat that arises is the breach of confidentiality of the data shared in the cloud. Major feature of cloud is that the user's data is usually processed remotely in unknown machines i.e., the servers that users do not own or operate that cause threat to confidentiality of sensitive data. The privacy of the sensitive data can be preserved by first encrypting the data before it is uploaded on to the cloud [140–144].

In the concept of Multi Owner data sharing, the challenging issues confronted by secure data sharing schemes are: Identity privacy, Revocation of Membership, Multiple owner manner, Access control, optimization of key generation methods and reduction of key storage spaces (as shown in Table 2).

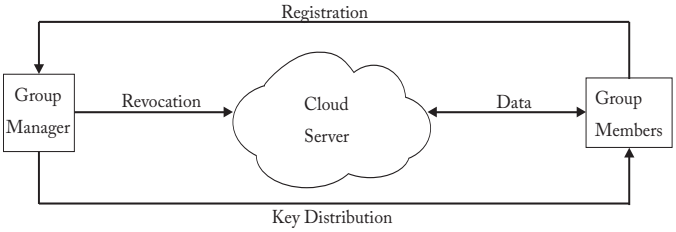


Fig. 5. Data Sharing Model

A. Key generation methods

Group Signature : The basic concept of group signature scheme is it permits any member of the group to sign messages while keeping their identity secure from the verifiers. In situations of a dispute, the group manager can reveal the identity of the originator of the signature.This concept is known as traceability [145, 146].

Dynamic Broadcast encryption : Broadcast Encryption [147] allows a broadcaster to transmit encrypted data to users of a group in such a way that only a restricted subgroup of users can decrypt the data. Along with the above feature, dynamic broadcast encryption enables the group manager to add new users while preserving the previously computed data i.e., without the need to modify the user decryption keys, the morphology and size of cipher texts are unchanged and the group encryption key requires no modification. Dynamic Broadcast Encryption is based on bilinear pairing technique [148].

B. System Model

System Model consists of three different entities: Cloud, Group Manager, Group member as shown in Figure 5.
Cloud: Cloud is operated by Cloud Service providers and provides the facility to store large amounts of data [63, 64, 149, 150].
Group Manager: The group manager is the administrator of the group i.e., duty is to handle the revocation of users, user registration, revealing real identity of data owner in case of a dispute. A group manager is fully trusted by the users of the group.
Group Members: Group members are a set of registered users who can store their private data in the cloud server and also share it along with other members of the group. Group membership is dynamically changed when new users are added or existing users are removed.

Several security schemes [63, 140–142, 149, 151] for sharing data on malicious servers have been proposed. In such proposed security approaches, the owners of the data upload the data files which are encrypted onto an untrusted storage server and decryption keys corresponding to each file is distributed only to users who are authorized by data owners. In this way, content of data files are not breached and leaked to users who are unauthorized and untrusted storage servers because they have no knowledge of the decryption

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

keys.
However, as the number of data owners and the number of revoked users increase, the complexities of user participation and revocation in these schemes are increasing linearly.

MONA implies that any user in the group can securely share data with others by the untrusted cloud. Multi Owner Data Sharing employs the features of Group Signature and Dynamic Encryption, through which it allows users to use the cloud resources anonymously and also securely share the data files all others including newly joined members. Group Signature allows a user to use the cloud anonymously while dynamic broadcast encryption allows a user to securely share their data with other members. In data sharing schemes, to provide data confidentiality for dynamic broadcast encryption each user has to process the revocation parameters which helps in keeping the data preserved from the revoked users. In such a case, the revocation results in computation overhead of encryption and as the number of revoked users increase, the size of cipher text also increases. Thus the heavy overhead and large cipher text impede the adoption of dynamic broadcast scheme.

To overcome such a drawback i.e., decrease the computation overhead of users and cipher text, the group manager computes the revocation parameters and it is transferred into the cloud where the revocation result is made public. This kind of process not only eliminates the previous drawbacks over computation overhead and cipher text but also makes them constant and independent of the number of revoked users.

C. Main design goals of Data sharing system

Main Design goals of data sharing system are access control, data confidentiality, anonymity and traceability, and efficiency.

Access control for data sharing system is defined for using the cloud for data operations. In Mona, unauthorized users are not allowed to access data at any time and a user can access data only at the revoked time allotted to him and is incapable of accessing data otherwise.

Data confidentiality means to store data in such a way that unauthorized users and cloud are not able to learn the contents of the file stored. Nevertheless, an obstacle stands in the way of data confidentiality which is its availability in dynamic groups. Therefore, new users have to decrypt the data before they take part and revoked users are unable to decrypt the data after revocation.

Cloud is managed by CSPs which are not known to the users and hence to assure the user no breach of identity, data sharing system allows any user to share data with others anonymously. This feature is known as anonymity. While anonymity provides efficient protection to the user, it also creates a vulnerable platform for an attack to the system. To resolve such instance another feature of data sharing system, traceability is introduced where in the group manager reveals the identity of a data owner in case of a dispute.

In the group any user can store data or share the stored files with the remaining users in the group. When revocation list is modified only for some users, the remaining users do not have to update their private keys or reencryption operations. Hence any user added to the group can acknowledge all the data files stored before his addition without contacting the data owner. This feature makes MONA an efficient scheme [152].

Boneh et al,. [12] have defined the mechanism of a public key encryption technique (PKES) with keyword search. There are two constructions for PKES; Decision Diffie-Hellman assumption and Trapdoor Permutation but are less efficient.The Identity Based Encryption is implemented in PKES but the converse is an open problem. The user privacy is not violated and gateway does not learn anything about the encrypted mail. It is used for a single user only and keyword may not be relevant to the message.

Bao et al., [153] have formulated a system model as well as security requirements for searchable encryption in a practical multi-user database groups. The construction for multi-user encrypted database system (MuED) uses a Bilinear map function. It provides query privacy, query unforgeability and addresses the problem of user revocation in a multi-user application. This construction is efficient achieving similar performance as most of the existing single-user schemes. This scheme has computation overhead

Shi et al.,[154] have designed an encryption scheme called Multi-dimensional Range Query over Encrypted Data $(MRQED)$, that supports privacy concerns related to the sharing of network audit logs using decision bilinear Diffie-Hellman key exchange. $MRQED$ can be useful for various other applications like financial audit logs, untrusted email servers and medical privacy. This scheme does not provide priority results over audit logs.

Xia *et al.*, [155] propose a public key encryption scheme in mobile cloud storage for a group of users for data sharing. A asymmetric group key agreement protocol and proxy re-signature is used for updating the searchable keywords which are encrypted. It ensures that mobile users in some group have to share the common secret key and update it when group members change.

D. PLUTUS

On connecting a storage system to the network, there is a high risk of privacy beach of data when it is navigates through an untrusted, public network. This forms an obstacle to the concept of easy sharing of data which is the main goal of storing data in network.

The solution to secure data traversed between clients and servers or between two data users by using traditional network security schemes does not suffice. In a data sharing network, the data stored is accessible for reading by users of that group, the data is placed by data owner in the storage system but the end user or recipient of the data cannot be determined a priors. Any modification or updation of data in

TABLE II
COMPARISON OF VARIOUS DATA SHARING SCHEMES

Parameters	PLUTUS[140]	SIRIUS [141]	IMPROVED PROXY ENCRYPTION[142]	SECURE SCALELABLE DATA ACCESS SCHEME [63]	MONA [156]
Encryption Technique	File-block key and lockbox –key	Public Key Cryptography	Proxy Cryptography	KP-ABE Technique	Broadcast Encryption
Identity Privacy	Satisfactory	Satisfactory	Less	Less	High
Revocation Mechanism	Inefficient	Inefficient	Inefficient	Inefficient	Efficient
Key Distribution	Heavy	Heavy	Medium	Independent of number of revoked users	Independent of number of revoked users

the storage system can be done by any of the users and a third user has to update the changes before the data reaches the end user. In [157], it proposes Plutus, a security scheme to prevent privacy of data when shared or stored on a network with an unsecure server [26, 158–160]. The concept on which the proposed Plutus scheme for secure data handling works is all the data is encrypted and stored, the key distribution is handled in a decentralized manner. Furthermore, the cryptographic encryption and key management operations are client-based and hence the server experiences a small amount of cryptographic overhead [140].

Plutus is an encrypt-on-disk system against the existing encrypt-on-wire systems and the advantages of this is protection against data out-pour when the device is attacked by a malicious server, for key distribution, policies can be established by users, scalability of server is possible as exhaustive cryptographic operations are performed at the end systems and not on centralized servers.

Plutus allows the data user to customize the security policies as the key distribution is performed by the clients. Compared to the existing security systems i.e., encrypt-on-wire systems, in Plutus as the key distribution is performed by the client, they experience a higher overhead while they have the same aggregate work within a system. Also, while in encrypt on wire system, files have to encrypted or decrypted every time an exchange of data over the network takes place, while on Plutus it calculates beforehand only the encryption when data is modified; this reduces the cost of encryption and decryption that takes place every time in the existing system. The concepts that Plutus works to provide security for data sharing is – identify any data modifications that are unauthorized and avert them from taking place, to distinguish between read and write access to files, to modify data users' access privileges.

1) Lazy Revocation: Plutus works on a main feature where the access privileges of the data users' can be modified or revoked by data owners. After a revocation, a revoked reader can read files that are not modified or the cached files. But a revoked reader cannot access (write or read) files that have been updated after modifications by the data owner. Lazy revocation (first introduced in [161]) results in a advantage of increased security and disadvantage of re-encryption cost.

XI. CONCLUSIONS

The research methods and schemes discussed are shown in Table 3 and Table 4. Efforts have been made to address the issues of storing the data on cloud and the drawbacks or problems that follow it. Noteworthy progress has been made in three main directions, mainly dealing with query effectiveness, security and efficiency. we present our conclusions over the review of various data retrieval and data sharing techniques.

Query Expressiveness: The existing schemes for data retrieval have improved due to several search features that are employed in various applications. Public Key Encryption with Keyword Search schemes are those which show a wide variety in query expressiveness due to the existence of comprehensive public key encryption schemes in this expanse. Searchable encryption in multi user sharing cloud setting is more indicative towards access control. With the usage of access control, a simple search that consists of a test to determine whether a certain trapdoor can be used for a cipher text is to be decrypted is conducted. If the decryption is successful then it indicates a match.

Efficiency: As research in data retrieval techniques using searchable encryption continues in all directions, efficiency gains importance especially in settings with multiple users. This issue for a requirement of efficiency has to be resolved for permitting the extensive use of searchable encryption in data retrieval techniques. If the proposed schemes in the data retrieval techniques are employed, they cause high latency and permit the server containing the database to be utilized by a restricted number of clients. Nevertheless, increased utilization of cloud and critical need of encryption in highly sensitive data cases has made it more important and urgent to build practical schemes.

Security: All schemes for data retrieval have attained verifiable security. In some cases, encryption is terminated if the data retrieval scheme is vulnerable. For the data retrieval schemes which are protected, the vital point is that most schemes are difficult to be evaluated based on security the reason being that these schemes are assessed on grounds of various computational assumptions. Therefore, it becomes impossible to assess existing schemes by comparing on basis of security. Several schemes permit the search pattern to be leaked while some make sure most of the information is

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

TABLE III
COMPARISON OF DIFFERENT MODELS

	Cloud security	Data sharing	Threats	Defense strategies	Requirements	Impact on society
Xiao *et al.,* [162]	Y	N	Y	Y	N	Y
Chen *et al.,* [163]	Y	Y	Y	N	Y	Y
Zhou *et al.*,[164]	Y	N	Y	Y	Y	Y
Wang *et al.,* [165]	Y	N	N	Y	Y	Y
Wang *et al.,* [166]	Y	N	N	Y	Y	N
Oza *et al.*,[167]	Y	N	Y	N	Y	Y
Saradhy *et al.,* [168]	N	Y	Y	Y	Y	Y
Butler *et al.,* [169]	N	Y	Y	N	Y	Y
Feldman *et al.,* [170]	N	Y	N	N	N	Y
Sahafizadeh *et al.,* [171]	N	Y	N	N	Y	Y

TABLE IV
SUMMARY OF VARIOUS SCHEMES

SSE	Searching in the encrypted data . the algorithm used are Symmetric searchable encryption for private and public encryption i.e., SSE1 ,SSE2. The performance of both the algorithms are O(n). Despite being more secure and more efficient, SSE schemes are remarkably simple. It needs more overhead calculation for searching.
Single Keyword Search	The betterment of the performance is achieved ensuring privacy to the keyword of the search. Dramatically increase the storage overhead. The search is augmented by providing results that are ranked and ordering them as top-k
Multi-Keyword Search	Effective approach to solve the problem of multi keyword ranked search over encrypted cloud data supporting synonym queries. The linear increase in the time and storage space for the increase in the length of the keyword. Since it is ranked keyword search so the search quality is enhanced significantly. Since its ranked based so to give rank we need to perform over head operations which is time consuming
Fuzzy Keyword Search	Keyword privacy enhanced variant of PEKS referred to as public-key encryption with fuzzy keyword search (PEFKS). The betterment of the performance is achieved by providing the security to the keyword. More secure and provides solution to keyword guess attack (KGA). Since it does encryption and decryption of the keyword so it is relatively time consuming process
Conjunctive Keyword Search	Building of a public-key system that supports a rich set of query predicates The algorithm used is deffie-helman algorithm. The performance varies as the size of the n varies . the betterment of the performance achieved by improving the security of the data. Secure and still fast enough. It needs lots of over-head calculation is needed
Data Sharing	A data owner authorizes every user by providing encryption keys. Each data user who have the encryption key are authorized to access the data that is encrypted in cloud using that key. Doing so, the user who is unauthorized, not having the authorization key does not get access to the encrypted data stored on cloud. Even if the unauthorized user gets access and downloads the ciphertext, the user will not be able to decipher the content as it is encrypted and the unauthorized user does not possess the decryption key.

hidden. This is acceptable in Several scenarios, leakage of search patterns is equitable while in some cases as sensitive Government organization databases, search and access pattern protection is obligatory.

A. Future Work

Future work in Data retrieval Technique should concentrate on improving the query articulateness, increasing the efficiency and enhancing the security of the present keyword search schemes. An interesting approach for further research on query effectiveness would be to bring the disparity between data retrieval techniques and plaintext searches.

Major stress has to be laid on efficiency of data retrieval techniques. One of the ways of doing that is by reducing the overhead computational complexity. Efficient and scalable data retrieval schemes have to be focussed in order to make widespread use of searchable encryption. another possible way the search process can be made efficient would be by investigating use of two or more collaborating servers. Another parameter for working towards scalability and efficiency of data retrieval technique is to loan the computation to third party entities

Enhancing security of Data retrieval Technique is important. Reason being the data stored on cloud may be of high confidentiality, like in data related to healthcare where patients' detail stored in their records, data related to Government wherein detail regarding the citizens or classified government schemes is stored, data related to personal profile detail on social networks and data in education or corporate sectors. Data stored on cloud are prone to data breach and hence more emphasis has to be laid on data retrieval schemes with high privacy and confidentiality.

Sharing information in today's technology accelerated living becomes important. In recent times, data sharing on cloud has also increased concerns. The main issues are dealing with privacy and security, user revocation, scalability and efficiency, collusion between entities.

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

REFERENCES

[1] Z. Slocum, "Your Google Docs: Soon in Search Results?" http://www.cnet.com/news/your-google-docs-soon-in-search-results/, 2009.

[2] B. Krebs, "Payment Processor Breach may be Largest Ever," http://voices. washingtonpost. com/securityfix/2009/01/payment processor breach may b.html, 2009.

[3] G. Brunette, R. Mogull *et al.*, "Security Guidance for Critical Areas of Focus in Cloud Computing v2. 1," *Cloud Security Alliance*, pp. 1–76, 2009.

[4] K. Ren, C. Wang, Q. Wang *et al.*, "Security Challenges for the Public Cloud," *in IEEE Internet Computing*, vol. 16, no. 1, pp. 69–73, 2012.

[5] C. Wang, N. Cao, J. Li, K. Ren, and W. Lou, "Secure Ranked Keyword Search over Encrypted Cloud Data," *in Proceedings of the IEEE 30th International Conference on Distributed Computing Systems (ICDCS)*, pp. 253–262, 2010.

[6] A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, and I. Stoica, "Above the Clouds: A Berkeley View of Cloud Computing," *Dept. Electrical Eng. and Comput. Sciences, University of California, Berkeley, Rep. UCB/EECS*, vol. 28, pp. 1–23, 2009.

[7] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility," *Future Generation Computer Systems*, vol. 25, no. 6, pp. 599–616, 2009.

[8] M. Arrington, "Gmail Disaster: Reports of Mass Email Deletions," *Online at http://www. techcrunch. com/2006/12/28/gmail-disasterreports-of-mass-email-deletions*, 2006.

[9] A. S. Team, "Amazon S3 Availability Event: July 20, 2008," *http://status.aws.amazon.com/s3-20080720.html*, 2008.

[10] H. Perl, Y. Mohammed, M. Brenner, and M. Smith, "Privacy/Performance Trade-off in Private Search on Bio-Medical Data," *Future Generation Computer Systems*, vol. 36, pp. 441–452, 2014.

[11] D. X. Song, D. Wagner, and A. Perrig, "Practical Techniques for Searches on Encrypted Data," *IEEE Symposium on Security and Privacy*, pp. 44–55, 2000.

[12] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public Key Encryption with Keyword Search," *in Proceedings of the Advances in Cryptology-Eurocrypt 2004*, pp. 506–522, 2004.

[13] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable Symmetric Encryption: Improved Definitions and Efficient Constructions," *in Proceedings of the 13th ACM Conference on Computer and Communications Security*, pp. 79–88, 2006.

[14] A. Singhal, "Modern Information Retrieval: A Brief Overview," *IEEE Data Eng. Bull.*, vol. 24, no. 4, pp. 35–43, 2001.

[15] C.-I. Fan and S.-Y. Huang, "Controllable Privacy Preserving Search Based on Symmetric Predicate Encryption in Cloud Storage," *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1716–1724, 2013.

[16] S. Zerr, D. Olmedilla, W. Nejdl, and W. Siberski, "Zerber+r: Top-k Retrieval from a Confidential Index," *in Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology*, pp. 439–449, 2009.

[17] A. Swaminathan, Y. Mao, G.-M. Su, H. Gou, A. L. Varna, S. He, M. Wu, and D. W. Oard, "Confidentiality-Preserving Rank-Ordered Search," *in Proceedings of the 2007 ACM Workshop on Storage Security and Survivability*, pp. 7–12, 2007.

[18] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-Preserving Multi-Keyword Ranked Search over Encrypted Cloud Data," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 1, pp. 222–233, 2014.

[19] H. Hu, J. Xu, C. Ren, and B. Choi, "Processing Private Queries over Untrusted Data Cloud through Privacy Homomorphism," *in Proceedings of the IEEE 27th International Conference on Data Engineering (ICDE)*, pp. 601–612, 2011.

[20] M. Wang, V. Holub, J. Murphy, and P. O'Sullivan, "High Volumes of Event Stream Indexing and Efficient Multi-keyword Searching for Cloud Monitoring," *Future Generation Computer Systems*, vol. 29, no. 8, pp. 1943–1962, 2013.

[21] S. Raghavendra, C. M. Geeta, R. Buyya, K. R. Venugopal, S. S. Iyengar, and L. M. Patnaik, "DRSIG: Domain and Range Specific Index Generation for Encrypted Cloud Data"," in *Proceedings of the International Conference on Computational Techniques in Information and Communication Technologies, ICCTICT 2016*. IEEE, March 17-20 2016.

[22] ——, "DRSMS: Domain and Range Specific Multi-Keyword Search over Encrypted Cloud Data," *International Journal of Computer Science and Information Security (IJCSIS)*, vol. 14, no. 5, pp. 69–78, 2016.

[23] D. Dubin, "The Most Influential Paper Gerard Salton Never Wrote," 2004.

[24] J. Zobel and A. Moffat, "Exploring the Similarity Space," *ACM SIGIR Forum*, vol. 32, no. 1, pp. 18–34, 1998.

[25] J. Kubiatowicz, D. Bindel, Y. Chen, S. Czerwinski, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, W. Weimer *et al.*, "Oceanstore: An Architecture for Global-Scale Persistent Storage," *ACM Sigplan Notices*, vol. 35, no. 11, pp. 190–201, 2000.

[26] A. Adya, W. J. Bolosky, M. Castro, G. Cermak, R. Chaiken, J. R. Douceur, J. Howell, J. R. Lorch, M. Theimer, and R. P. Wattenhofer, "FARSITE: Federated, Available, and Reliable Storage for an Incompletely Trusted Environment," *ACM SIGOPS Operating Systems Review*, vol. 36, no. SI, pp. 1–14, 2002.

[27] A. Muthitacharoen, R. Morris, T. M. Gil, and B. Chen, "IVY: A Read/Write Peer-to-Peer File System," *ACM SIGOPS Operating Systems Review*, vol. 36, no. SI, pp.

https://sites.google.com/site/ijcsis/
ISSN 1947-5500

31–44, 2002.

[28] E.-J. Goh, "Secure Indexes." *in IACR Cryptology ePrint Archive*, p. 216, 2003.

[29] S. M. Bellovin and W. R. Cheswick, "Privacy-Enhanced Searches using Encrypted Bloom Filters," 2007.

[30] Y.-C. Chang and M. Mitzenmacher, "Privacy Preserving Keyword Searches on Remote Encrypted Data," *Applied Cryptography and Network Security*, pp. 442–455, 2005.

[31] J. D. Tygar, "Security with Privacy*," *ISAT 2002 Study*, 2002.

[32] M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, and H. Shi, "Searchable Encryption Revisited: Consistency Properties, Relation to Anonymous IBE, and Extensions," *Advances in Cryptology–CRYPTO 2005*, pp. 205–222, 2005.

[33] D. Boneh, E. Kushilevitz, R. Ostrovsky, and W. E. Skeith III, "Public Key Encryption that Allows PIR Queries," *Advances in Cryptology-CRYPTO 2007*, pp. 50–67, 2007.

[34] R. Ostrovsky, "Efficient Computation on Oblivious RAMs," *Proceedings of the Twenty-Second Annual ACM Symposium on Theory of Computing*, pp. 514–523, 1990.

[35] O. Goldreich and R. Ostrovsky, "Software Protection and Simulation on Oblivious RAMs," *in Journal of the ACM (JACM)*, vol. 43, no. 3, pp. 431–473, 1996.

[36] C. H. Wang and C.-C. Hsu, "Integration of Hierarchical Access Control and Keyword Search Encryption in Cloud Computing Environment," *in International Journal of Computer and Communication Engineering*, vol. 3, no. 2, pp. 333–337, 2013.

[37] Q. Liu, G. Wang, and J. Wu, "An Efficient Privacy Preserving Keyword Search Scheme in Cloud Computing," *in Proceedings of the International Conference on Computational Science and Engineering CSE'09*, vol. 2, pp. 715–720, 2009.

[38] D. Boneh and M. Franklin, "Identity-based Encryption from the Weil Pairing," *in Proceedings of the Advances in Cryptology—CRYPTO 2001*, pp. 213–229, 2001.

[39] Z. Jiang and L. Liu, "Secure Cloud Storage Service with an Efficient DOKS Protocol," *in Proceedings of the IEEE International Conference on Services Computing (SCC)*, pp. 208–215, 2013.

[40] S. KumarVerma, S. Mathew, S. Srivastava, and S. Venkataesan, "An Efficient Dictionary and Lingual Keyword based Secure Search Scheme in Cloud Storage," *in International Journal of Computer Applications*, vol. 68, no. 15, pp. 40–43, 2013.

[41] Y. Lu, "Privacy-Preserving Logarithmic-Time Search on Encrypted Data in Cloud." *in 19th Annual Network and Distributed System Security Symposium, (NDSS)*, 2012.

[42] Z. Xia, Y. Zhu, X. Sun, and J. Wang, "A Similarity Search Scheme over Encrypted Cloud Images based on Secure Transformation," *International Journal of Future Generation Communication and Networking*, vol. 6, no. 6, pp. 71–80, 2013.

[43] X. Pang, B. Yang, M. Zhang, and H. Wang, "Multi-user Noisy Keyword Search over Encrypted Data," *Journal of Computational Information Systems*, vol. 9, no. 5, pp. 1973–1981, 2013.

[44] C. Gu, Y. Guang, Y. Zhu, and Y. Zheng, "Public Key Encryption with Keyword Search from Lattices," *in International Journal of Information Technology*, vol. 19, no. 1, pp. 1–10, 2013.

[45] W. Wang, P. Xu, H. Li, and L. T. Yang, "Secure Hybrid-Indexed Search for High Efficiency over Keyword Searchable Ciphertexts," *Future Generation Computer Systems*, 2014.

[46] S. Raghavendra, C. M. Geeta, K. Shaila, R. Buyya, K. R. Venugopal, S. S. Iyengar, and L. M. Patnaik, "MSSS: Most Significant Single-keyword Search over Encrypted Cloud Data," *in Proceedings of the 6th Annual Intrernational Conference on ICT: BigData, Cloud and Securit*, 2015.

[47] S. Raghavendra, C. M. Geeta, R. Buyya, K. R. Venugopal, S. S. Iyengar, and L. M. Patnaik, "MSIGT: Most Significant Index Generation Technique for Cloud Environment," *Proceedings of the 12th IEEE India International Conference on E^3-C^3(INDICON 2015)*, December 11-13 2015.

[48] C. Wang, N. Cao, K. Ren, and W. Lou, "Enabling Secure and Efficient Ranked Keyword Search over Outsourced Cloud Data," *in IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 8, pp. 1467–1479, 2012.

[49] A. Boldyreva, N. Chenette, Y. Lee, and A. O'neill, "Order-Preserving Symmetric Encryption," *in Proceedings of the International Conference Advances in Cryptology-EUROCRYPT*, pp. 224–241, 2009.

[50] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling Public Verifiability and Data Dynamics for Storage Security in Cloud Computing," *Computer Security–ESORICS 2009*, pp. 355–370, 2009.

[51] C. Wang, Q. Wang, K. Ren, N. Cao, and W. Lou, "Toward Secure and Dependable Storage Services in Cloud Computing," *IEEE Transactions on Services Computing,*, vol. 5, no. 2, pp. 220–232, 2012.

[52] C. Wang, S. S. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Secure Cloud Storage," *IEEE Transactions on Computers*, vol. 62, no. 2, pp. 362–375, 2013.

[53] C. L. Cheng, C. J. Sun, X. L. Xu, and D. Y. Zhang, "A Multi-Dimensional Index Structure based on Improved VA-file and CAN in the Cloud," *in International Journal of Automation and Computing*, vol. 11, no. 1, pp. 109–117, 2014.

[54] S. Buyrukbilen and S. Bakiras, "Privacy-Preserving Ranked Search on Public-Key Encrypted Data," *in Proceedings of the 2013 IEEE International Conference on High Performance Computing and Communications*, pp. 165–174, 2013.

[55] M. Kuzu, M. S. Islam, and M. Kantarcioglu, "Efficient Similarity Search over Encrypted Data," *in IEEE 28th International Conference on Data Engineering (ICDE)*,

pp. 1156–1167, 2012.

[56] Q. Liu, G. Wang, and J. Wu, "Secure and Privacy Preserving Keyword Searching for Cloud Storage Services," *in Journal of Network and Computer Applications*, vol. 35, no. 3, pp. 927–933, 2012.

[57] D. Yuan, Y. Yang, X. Liu, W. Li, L. Cui, M. Xu, and J. Chen, "A Highly Practical Approach Toward Achieving Minimum Data Sets Storage Cost in the Cloud," *IEEE Transactions on Parallel & Distributed Systems*, no. 6, pp. 1234–1244, 2013.

[58] L. M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, "A Break in the Clouds: Towards A Cloud Definition," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 1, pp. 50–55, 2008.

[59] N. Cao, S. Yu, Z. Yang, W. Lou, and Y. T. Hou, "Lt Codes-based Secure and Reliable Cloud Storage Service," *in 2012 Proceedings IEEE INFOCOM*, pp. 693–701, 2012.

[60] M. Bellare, A. Boldyreva, and A. O'Neill, "Deterministic and Efficiently Searchable Encryption," *Advances in Cryptology-CRYPTO 2007*, pp. 535–552, 2007.

[61] M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, and H. Shi, "Searchable Encryption Revisited: Consistency Properties, Relation to Anonymous IBE, and Extensions," *Journal of Cryptology*, vol. 21, no. 3, pp. 350–391, 2008.

[62] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "Fuzzy Keyword Search Over Encrypted Data in Cloud Computing," *in 2010 Proceedings IEEE INFOCOM*, pp. 1–5, 2010.

[63] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving Secure, Scalable, and Fine-Grained Data Access Control in Cloud Computing," *in 2010 Proceedings IEEE INFOCOM*, pp. 1–9, 2010.

[64] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing," *in 2010 Proceedings IEEE INFOCOM*, pp. 1–9, 2010.

[65] J. Yu, S. J. T. P. Lu, Y. Zhu, G. Xue, and M. Li, "Toward Secure Multikeyword Top-k Retrieval over Encrypted Cloud Data," *in IEEE Transactions on Dependable and Secure Computing*, vol. 10, no. 4, pp. 239–250, 2013.

[66] X. Sun, X. Wang, Z. Xia, Z. Fu, and T. Li, "Dynamic Multi-Keyword Top-k Ranked Search over Encrypted Cloud Data." *in International Journal of Security & Its Applications*, vol. 8, no. 1, pp. 319–332, 2014.

[67] C. Orencik, M. Kantarcioglu, and E. Savas, "A Practical and Secure Multi-Keyword Search Method over Encrypted Cloud Data," *in Proceedings of the IEEE Sixth International Conference on Cloud Computing (CLOUD)*, pp. 390–397, 2013.

[68] W. Sun, B. Wang, N. Cao, M. Li, W. Lou, Y. Hou, and H. Li, "Verifiable Privacy-Preserving Multi-keyword Text Search in the Cloud Supporting Similarity-based Ranking," *in IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 11, pp. 3025–3035, 2014.

[69] M. Li, S. Yu, N. Cao, and L. Wenjing, "Authorized Private Keyword Search over Encrypted Data in Cloud Computing," *in Proceedings of the 31st International Conference on Distributed Computing Systems (ICDCS)*, pp. 383–392, 2011.

[70] C. Örencik and E. Savaş, "Efficient and Secure Ranked Multi-keyword Search on Encrypted Cloud Data," *in Proceedings of the 2012 Joint EDBT/ICDT Workshops*, pp. 186–195, 2012.

[71] L. Chen, X. Sun, Z. Xia, and Q. Liu, "An Efficient and Privacy-Preserving Semantic Multi-Keyword Ranked Search over Encrypted Cloud Data," *in International Journal of Security and Its Applications*, vol. 8, no. 2, pp. 323–332, 2014.

[72] K. Li, W. Zhang, K. Tian, R. Liu, and N. Yu, "An Efficient Multi-keyword Ranked Retrieval Scheme with Johnson-Lindenstrauss Transform over Encrypted CloudData," *in Proceedings of the International Conference on Cloud Computing and Big Data (CloudCom-Asia)*, pp. 320–327, 2013.

[73] W. Zhang, S. Xiao, Y. Lin, T. Zhou, and S. Zhou, "Secure Ranked Multi-keyword Search for Multiple Data Owners in Cloud Computing," *in Proceedings of the 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pp. 276–286, 2014.

[74] R. Li, Z. Xu, W. Kang, K. C. Yow, and C.-Z. Xu, "Efficient Multi-Keyword Ranked Query over Encrypted Data in Cloud Computing," *Future Generation Computer Systems*, vol. 30, pp. 179–190, 2014.

[75] W. Sun, B. Wang, N. Cao, M. Li, W. Lou, Y. T. Hou, and H. Li, "Privacy-Preserving Multi-Keyword Text Search in the Cloud Supporting Similarity-Based Ranking," *Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security*, pp. 71–82, 2013.

[76] S. Ji, G. Li, C. Li, and J. Feng, "Efficient Interactive Fuzzy Keyword Search," *in Proceedings of the 18th International Conference on World Wide Web*, pp. 371–380, 2009.

[77] L. Liu, C. Zhang, S. Yao, S. Wang, and W. Zhou, "Fuzzy Keyword Search with Safe Index over Encrypted Cloud Computing," *TELKOMNIKA Indonesian Journal of Electrical Engineering*, vol. 11, no. 10, pp. 5884–5889, 2013.

[78] S. Raghavendra, S. Girish, C. M. Geeta, R. Buyya, K. R. Venugopal, S. S. Iyengar, and L. M. Patnaik, "IGSK: Index Generation on Split Keyword for Search Over Cloud Data." *Proceedings of the 2015 International Conference on Computing and Network Communications (CoCoNet'15)*, pp. 380–386, December 16-19 2015.

[79] X. Sun, Y. Zhu, Z. Xia, J. Wang, and L. Chen, "Secure Keyword-based Ranked Semantic Search over Encrypted Cloud Data," *in Proceedings of the Advanced Science and Technology Letters(MulGraB 2013)*, vol. 31, pp. 271–283, 2013.

[80] P. Xu, H. Jin, Q. Wu, and W. Wang, "Public-Key Encryption with Fuzzy Keyword Search: A Provably

Secure Scheme under Keyword Guessing Attack," *in IEEE Transactions on Computers*, vol. 62, no. 11, pp. 2266–2277, 2013.

[81] W. Wang, P. Xu, H. Li, and L. T. Yang, "Secure Hybrid-Indexed Search for High Efficiency over Keyword Searchable Ciphertexts," *Future Generation Computer Systems*, 2014.

[82] D. Wang, S. Fu, and M. Xu, "A Privacy-Preserving Fuzzy Keyword Search Scheme over Encrypted Cloud Data," *in Proceedings of the 5th International Conference on Cloud Computing Technology and Science (CloudCom)*, vol. 1, pp. 663–670, 2013.

[83] B. Wang, S. Yu, W. Lou, and Y. T. Hou, "Privacy-Preserving Multi-Keyword Fuzzy Search over Encrypted Data in the Cloud," *in Proceedings IEEE INFOCOM*, pp. 2112–2120, 2014.

[84] M. Chuah and W. Hu, "Privacy-Aware Bedtree Based Solution for Fuzzy Multi-Keyword Search over Encrypted Data," *31st International Conference on Distributed Computing Systems Workshops (ICDCSW)*, pp. 273–281, 2011.

[85] C. Wang, Q. Wang, and K. Ren, "Towards Secure and Effective Utilization Over Encrypted Cloud Data," *31st International Conference on Distributed Computing Systems Workshops (ICDCSW)*, pp. 282–286, 2011.

[86] J. Wang, H. Ma, Q. Tang, J. Li, H. Zhu, S. Ma, and X. Chen, "Efficient Verifiable Fuzzy Keyword Search over Encrypted Data in Cloud Computing," *Computer Science and Information Systems/ComSIS*, vol. 10, no. 2, pp. 667–684, 2013.

[87] W. Jie, Y. Xiao, Z. Ming, and W. Yong, "A Novel Dynamic Ranked Fuzzy Keyword Search over Cloud Encrypted Data," *12th International Conference on Dependable, Autonomic and Secure Computing (DASC)*, pp. 91–96, 2014.

[88] D. Wang, S. Fu, and M. Xu, "A Privacy-Preserving Fuzzy Keyword Search Scheme over Encrypted Cloud Data," *5th International Conference on Cloud Computing Technology and Science (CloudCom)*, vol. 1, pp. 663–670, 2013.

[89] S. Bijral and D. Mukhopadhyay, "Efficient Fuzzy Search Engine with B-Tree Search Mechanism," *arXiv preprint arXiv:1411.6773*, 2014.

[90] N. Shekokar, K. Sampat, C. Chandawalla, and J. Shah, "Implementation of Fuzzy Keyword Search over Encrypted Data in Cloud Computing," *Procedia Computer Science*, vol. 45, pp. 499–505, 2015.

[91] T. Balamuralikrishna, C. Anuradha, and N. Raghavendrasai, "Fuzzy Keyword Search over Encrypted Data in Cloud Computing," *Asian Journal of Computer Science & Information Technology*, vol. 1, no. 3, 2013.

[92] W. Zhou, L. Liu, H. Jing, C. Zhang, S. Yao, and S. Wang, "K-Gram Based Fuzzy Keyword Search over Encrypted Cloud Computing," 2013.

[93] L. Ballard, S. Kamara, and F. Monrose, "Achieving Efficient Conjunctive Keyword Searches over Encrypted Data," *Information and Communications Security*, pp. 414–426, 2005.

[94] M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, and H. Shi, "Encryption with Keyword Search, Revisited: Consistency Conditions, Relations to Anonymous IBE, and Extensions," *in Proceedings of Crypto'05*, pp. 205–222, 2005.

[95] B. Chor, E. Kushilevitz, O. Goldreich, and M. Sudan, "Private Information Retrieval," *Journal of the ACM (JACM)*, vol. 45, no. 6, pp. 965–981, 1998.

[96] G. Di-Crescenzo, Y. Ishai, and R. Ostrovsky, "Universal Service-Providers for Database Private Information Retrieval," *in Proceedings of the Seventeenth Annual ACM Symposium on Principles of Distributed Computing*, pp. 91–100, 1998.

[97] P. Golle, J. Staddon, and B. Waters, "Secure Conjunctive Keyword Search over Encrypted Data," *in Proceedings of the Applied Cryptography and Network Security*, pp. 31–45, 2004.

[98] R. Ostrovsky and W. E. Skeith III, "Private Searching on Streaming Data," *Advances in Cryptology–CRYPTO 2005*, pp. 223–240, 2005.

[99] W. Ogata and K. Kurosawa, "Oblivious Keyword Search," *Journal of Complexity*, vol. 20, no. 2, pp. 356–371, 2004.

[100] D. J. Park, K. Kim, and P. J. Lee, "Public Key Encryption with Conjunctive Field Keyword Search," *Information Security Applications*, pp. 73–86, 2005.

[101] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong, "Freenet: A Distributed Anonymous Information Storage and Retrieval System," *Designing Privacy Enhancing Technologies*, pp. 46–66, 2001.

[102] B. R. Waters, D. Balfanz, G. Durfee, and D. K. Smetters, "Building an Encrypted and Searchable Audit Log." *NDSS*, vol. 4, pp. 5–6, 2004.

[103] N. S. Khan, C. R. Krishna, and A. Khurana, "Secure Ranked Fuzzy Multi-Keyword Search over Outsourced Encrypted Cloud Data," *International Conference on Computer and Communication Technology (ICCCT)*, pp. 241–249, 2014.

[104] H. Tuo and M. Wenping, "An Effective Fuzzy Keyword Search Scheme in Cloud Computing," *in 2013 5th International Conference on Intelligent Networking and Collaborative Systems (INCoS)*, pp. 786–789, 2013.

[105] Z. Fu, J. Shu, X. Sun, and N. Linge, "Smart Cloud Search Services: Verifiable Keyword-Based Semantic Search over Encrypted Cloud Data," *IEEE Transactions on Consumer Electronics*, vol. 60, no. 4, pp. 762–770, 2014.

[106] Z. Fu, X. Sun, Z. Xia, L. Zhou, and J. Shu, "Multi-Keyword Ranked Search Supporting Synonym Query over Encrypted Data in Cloud Computing," *2013 IEEE 32nd International Performance Computing and Communications Conference (IPCCC)*, pp. 1–8, 2013.

[107] Z. Fu, J. Shu, X. Sun, and D. Zhang, "Semantic Keyword Search Based on Trie over Encrypted Cloud Data," *Proceedings of the 2nd international Workshop on Security in Cloud Computing*, pp. 59–62, 2014.

[108] J. Ko, S. Shin, S. Eom, M. Song, J. Jung, D. H.

Shin, K. H. Lee, and Y. Jang, "Keyword Based Semantic Search for Mobile Data," *in 2014 IEEE 15th International Conference on Mobile Data Management (MDM)*, vol. 1, pp. 245–248, 2014.

[109] R. ChinnaSamy and S. Sujatha, "An Efficient Semantic Secure Keyword Based Search Scheme in Cloud Storage Services," *in 2012 International Conference on Recent Trends In Information Technology (ICRTIT)*, pp. 488–491, 2012.

[110] T.-S. Moh and K. H. Ho, "Efficient Semantic Search over Encrypted Data in Cloud Computing," *in 2014 International Conference on High Performance Computing & Simulation (HPCS)*, pp. 382–390, 2014.

[111] Q. Xu, H. Shen, Y. Sang, and H. Tian, "Privacy-Preserving Ranked Fuzzy Keyword Search over Encrypted Cloud Data," *in 2013 International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT)*, pp. 239–245, 2013.

[112] M. B. B. Pepsi and K. Mala, "Similarity Search on Metric Data of Outsourced Lung Images," *in 2013 IEEE International Conference on Green High Performance Computing (ICGHPC)*, pp. 1–6, 2013.

[113] S. Raghavendra, K. Nithyashree, C. M. Geeta, R. Buyya, K. R. Venugopal, S. S. Iyengar, and L. M. Patnaik, "RSSMSO Rapid Similarity Search on Metric Space Object Stored in Cloud Environment," *International Journal of Organizational and Collective Intelligence (IJOCI)*, vol. 6, no. 3, pp. 32–47, 2016.

[114] ——, "FRORSS: Fast Result Object Retrieval using Similarity Search on Cloud," *Proceedings of the International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER 2016)*, 2016.

[115] S. Kozak, D. Novak, and P. Zezula, "Secure Metric-based Index for Similarity Cloud," *Secure Data Management*, pp. 130–147, 2012.

[116] G. Aggarwal, T. Feder, K. Kenthapadi, S. Khuller, R. Panigrahy, D. Thomas, and A. Zhu, "Achieving Anonymity via Clustering," *in Proceedings of the Twenty-Fifth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pp. 153–162, 2006.

[117] M. L. Yiu, I. Assent, C. S. Jensen, and P. Kalnis, "Outsourced Similarity Search on Metric Data Assets," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 2, pp. 338–352, 2012.

[118] P. Ciaccia, M. Patella, and P. Zezula, "DEIS-CSITE-CNR," *in Proceedings of the International Conference on Very Large Data Bases*, vol. 23, pp. 426–435, 1997.

[119] A. Khoshgozaran and C. Shahabi, "Blind Evaluation of Nearest Neighbor Queries using Space Transformation to Preserve Location Privacy," *Advances in Spatial and Temporal Databases*, pp. 239–257, 2007.

[120] M. Connor and P. Kumar, "Fast Construction of k-Nearest Neighbor Graphs for Point Clouds," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 4, pp. 599–608, 2010.

[121] W. K. Wong, D. W.-l. Cheung, B. Kao, and N. Mamoulis, "Secure kNN Computation on Encrypted Databases," *in Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, pp. 139–152, 2009.

[122] T. Bozkaya and M. Ozsoyoglu, "Indexing Large Metric Spaces for Similarity Search Queries," *ACM Transactions on Database Systems (TODS)*, vol. 24, no. 3, pp. 361–404, 1999.

[123] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu, "Order Preserving Encryption for Numeric Data," *in Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pp. 563–574, 2004.

[124] S. Kozak and P. Zezula, "Efficiency and Security in Similarity Cloud Services," *in Proceedings of the Very Large Data Base Endowment*, vol. 6, no. 12, pp. 1450–1455, 2013.

[125] Y. Zhu, X. Sun, Z. Xia, L. Chen, T. Li, and D. Zhang, "Enabling Similarity Search over Encrypted Images in Cloud," *Information Technology Journal*, vol. 13, no. 5, pp. 824–831, 2014.

[126] D. Chen and Y. He, "A Study on Secure Data Storage Strategy in Cloud Computing," *Journal of Convergence Information Technology*, vol. 5, no. 7, pp. 175–179, 2010.

[127] H. Lu, B. C. Ooi, H. T. Shen, and X. Xue, "Hierarchical Indexing Structure for Efficient Similarity Search in Video Retrieval," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 11, pp. 1544–1559, 2006.

[128] B. Cui, B. C. Coi, J. Su, and K.-L. Tan, "Indexing High-Dimensional Data for Efficient In-Memory Similarity Search," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 3, pp. 339–353, 2005.

[129] Z. Xia, Y. Zhu, X. Sun, and L. Chen, "Secure Semantic Expansion Based Search over Encrypted Cloud Data Supporting Similarity Ranking," *Journal of Cloud Computing*, vol. 3, no. 1, pp. 1–11, 2014.

[130] M. Jang, M. Yoon, and J.-W. Chang, "A k-Nearest Neighbor Search Algorithm for Privacy Preservation in Outsourced Spatial Databases," *International Journal of Smart Home*, vol. 7, no. 3, pp. 239–247, 2013.

[131] G. R. Hjaltason and H. Samet, "Index-Driven Similarity Search in Metric Spaces (Survey Article)," *ACM Transactions on Database Systems (TODS)*, vol. 28, no. 4, pp. 517–580, 2003.

[132] A. Tsymbal, E. Meissner, M. Kelm, and M. Kramer, "Towards Cloud-Based Image-Integrated Similarity Search in Big Data," *in IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*, pp. 593–596, 2014.

[133] V. Athitsos, M. Potamias, P. Papapetrou, and G. Kollios, "Nearest Neighbor Retrieval using Distance-Based Hashing," *in IEEE 24th International Conference on Data Engineering (ICDE 2008)*, pp. 327–336, 2008.

[134] K. Hajebi, Y. Abbasi-Yadkori, H. Shahbazi, and H. Zhang, "Fast Approximate Nearest-Neighbor Search with k-Nearest Neighbor Graph," *in Proceedings of the International Joint Conference on Artificial Intelligence*

(IJCAI), vol. 22, no. 1, pp. 1312–1317, 2011.

[135] I. Popivanov and R. J. Miller, "Similarity Search over Time-Series Data using Wavelets," *in Proceedings of the 18th International Conference on Data Engineering*, pp. 212–221, 2002.

[136] G. Amato and P. Savino, "Approximate Similarity Search in Metric Spaces using Inverted Files," *in Proceedings of the 3rd International Conference on Scalable Information Systems*, no. 28, pp. 1–10, 2008.

[137] V. Gil-Costa and M. Marin, "Load Balancing Query Processing in Metric-Space Similarity Search," *in 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid)*, pp. 368–375, 2012.

[138] M. Yoon, H.-I. Kim, M. Jang, and J.-W. Chang, "Linear Function Based Transformation Scheme for Preserving Database Privacy in Cloud Computing," *in 2013 International Conference on Parallel and Distributed Systems (ICPADS)*, pp. 498–503, 2013.

[139] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica *et al.*, "A View of Cloud Computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[140] M. Kallahalla, E. Riedel, R. Swaminathan, Q. Wang, and K. Fu, "Plutus: Scalable Secure File Sharing on Untrusted Storage." *Fast*, vol. 3, pp. 29–42, 2003.

[141] E.-J. Goh, H. Shacham, N. Modadugu, and D. Boneh, "SiRiUS: Securing Remote Untrusted Storage." *NDSS*, vol. 3, pp. 131–145, 2003.

[142] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved Proxy Re-encryption Schemes with Applications to Secure Distributed Storage," *ACM Transactions on Information and System Security (TISSEC)*, vol. 9, no. 1, pp. 1–30, 2006.

[143] S. Raghavendra, P. A. Doddabasappa, C. M. Geeta, R. Buyya, K. R. Venugopal, S. S. Iyengar, and L. M. Patnaik, "Secure Multi-Keyword Search and Multi-User Access Control over an Encrypted Cloud Data," *International Journal of Information Processing(IJIP)*, vol. 10, no. 2, pp. 51–61, 2016.

[144] S. Raghavendra, K. Meghana, P. Doddabasappa, C. Geeta, R. Buyya, K. Venugopal, S. Iyengar, and L. Patnaik, "Index Generation and Secure Multi-user Access Control over an Encrypted Cloud Data," *Procedia Computer Science*, vol. 89, pp. 293–300, 2016.

[145] D. Boneh, X. Boyen, and H. Shacham, "Short Group Signatures," *Advances in Cryptology–CRYPTO 2004*, pp. 41–55, 2004.

[146] D. Chaum and E. Van Heyst, "Group Signatures," *Advances in Cryptology—EUROCRYPT'91*, pp. 257–265, 1991.

[147] A. Fiat and M. Naor, "Broadcast Encryption," *Advances in Cryptology—CRYPTO'93*, pp. 480–491, 1994.

[148] C. Delerablée, P. Paillier, and D. Pointcheval, "Fully Collusion Secure Dynamic Broadcast Encryption with Constant-Size Ciphertexts or Decryption Keys," *Pairing-Based Cryptography–Pairing 2007*, pp. 39–59, 2007.

[149] R. Lu, X. Lin, X. Liang, and X. S. Shen, "Secure Provenance: the Essential of Bread and Butter of Data Forensics in Cloud Computing," *in Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, pp. 282–292, 2010.

[150] B. Wang, B. Li, and H. Li, "Knox: Privacy-Preserving Auditing for Shared Data with Large Groups in the Cloud," *Applied Cryptography and Network Security*, pp. 507–525, 2012.

[151] D. Naor, M. Naor, and J. Lotspiech, "Revocation and Tracing Schemes for Stateless Receivers," *Advances in Cryptology—CRYPTO 2001*, pp. 41–62, 2001.

[152] S. Kamara and K. Lauter, "Cryptographic Cloud Storage," *Financial Cryptography and Data Security*, pp. 136–149, 2010.

[153] F. Bao, R. H. Deng, X. Ding, and Y. Yang, "Private Query on Encrypted Data in Multi-user Settings," *in Proceedings of the International Conference on Information Security Practice and Experience*, pp. 71–85, 2008.

[154] E. Shi, J. Bethencourt, T.-H. Chan, D. Song, and A. Perrig, "Multi-dimensional Range Query over Encrypted Data," *in IEEE Symposium on Security and Privacy*, pp. 350–364, 2007.

[155] Q. Xia, J. Ni, A. J. B. A. Kanpogninge, and J. C. Gee, "Searchable Public-Key Encryption with Data Sharing in Dynamic Groups for Mobile Cloud Storage," *Journal of Universal Computer Science*, vol. 21, no. 3, pp. 440–453, 2015.

[156] X. Liu, Y. Zhang, B. Wang, and J. Yan, "Mona: Secure Multi-Owner Data Sharing for Dynamic Groups in the Cloud," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 6, pp. 1182–1191, 2013.

[157] K. Fu, M. Kallahalla, S. Rajagopalan, and R. Swaminathan, "Secure Rotation on Key Sequences," *Submitted for Publication*, 2002.

[158] M. Castro and B. Liskov, "Proactive Recovery in a Byzantine-Fault-Tolerant System," *in Proceedings of the 4th Conference on Symposium on Operating System Design & Implementation-Volume 4*, pp. 19–19, 2000.

[159] G. R. Ganger, P. K. Khosla, M. Bakkaloglu, M. W. Bigrigg, G. R. Goodson, S. Oguz, V. Pandurangan, C. A. Soules, J. D. Strunk, and J. J. Wylie, "Survivable Storage Systems," *in Proceedings of the DARPA Information Survivability Conference & Exposition II, 2001. DISCEX'01*, vol. 2, pp. 184–195, 2001.

[160] A. Rowstron and P. Druschel, "Storage Management and Caching in PAST, A Large-Scale, Persistent Peer-to-Peer Storage Utility," *ACM SIGOPS Operating Systems Review*, vol. 35, no. 5, pp. 188–201, 2001.

[161] K. E. Fu, "Group Sharing and Random Access in Cryptographic Storage File Systems," Ph.D. dissertation, Citeseer, 1999.

[162] Z. Xiao and Y. Xiao, "Security and Privacy in Cloud Computing," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 2, pp. 843–859, 2013.

[163] D. Chen and H. Zhao, "Data Security and Privacy Protection Issues in Cloud Computing," *in 2012 International Conference on Computer Science and Electronics*

Engineering (ICCSEE), vol. 1, pp. 647–651, 2012.

[164] M. Zhou, R. Zhang, W. Xie, W. Qian, and A. Zhou, "Security and Privacy in Cloud Computing: A Survey," *in 2010 Sixth International Conference on Semantics Knowledge and Grid (SKG)*, pp. 105–112, 2010.

[165] J.-S. Wang, C.-H. Liu, and G. T. Lin, "How to Manage Information Security in Cloud Computing," *in 2011 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 1405–1410, 2011.

[166] Y.-H. Wang, "The Role of SaaS Privacy and Security Compliance for Continued SaaS Use," *in 2011 7th International Conference on Networked Computing and Advanced Information Management (NCM)*, pp. 303–306, 2011.

[167] N. Oza, K. Karppinen, and R. Savola, "User Experience and Security in the Cloud–An Empirical Study in the Finnish Cloud Consortium," *in 2010 IEEE Second International Conference on Cloud Computing Technology and Science (CloudCom)*, pp. 621–628, 2010.

[168] R. Sarathy and K. Muralidhar, "Secure and Useful Data Sharing," *Decision Support Systems*, vol. 42, no. 1, pp. 204–220, 2006.

[169] D. Butler, "Data Sharing Threatens Privacy," *Nature News*, vol. 449, no. 7163, pp. 644–645, 2007.

[170] L. Feldman, D. Patel, L. Ortmann, K. Robinson, and T. Popovic, "Educating for the Future: Another Important Benefit of Data Sharing," *The Lancet*, vol. 379, no. 9829, pp. 1877–1878, 2012.

[171] E. Sahafizadeh and S. Parsa, "Survey on Access Control Models," *in 2010 2nd International Conference on Future Computer and Communication*, 2010.