International Journal of Innovative Research and Advanced Studies (IJIRAS)
Volume 4 Issue 12, December 2017

ISSN: 2394-4404

Searching, Ranking & Downloading The Dynamic Multi-Keyword Ranked Based Encrypted Cloud Data

M. Santhosh

PG Scholar, Dept of CSE, Kuppam Engineering College, Kuppam, AP, India

K. N. Dharanidhar

Assistant Professor, Dept of CSE, Kuppam Engineering College, Kuppam, AP, India

Abstract: ***Due to the increasing popularity of cloud computing, more and more data owners are motivated to outsource their data to cloud servers for great convenience and reduced cost in data management. However, sensitive data should be encrypted before outsourcing for privacy requirements, which obsoletes data utilization like keyword-based document retrieval. In this paper, we present a secure multi-keyword ranked search scheme over encrypted cloud data, which simultaneously supports dynamic update operations like deletion and insertion of documents. Specifically, the vector space model and the widely-used TF X IDF model are combined in the index construction and query generation. We construct a special tree-based index structure and propose a "Greedy Depth-first Search" algorithm to provide efficient multi-keyword ranked search. The secure kNN algorithm is utilized to encrypt the index and query vectors, and meanwhile ensure accurate relevance score calculation between encrypted index and query vectors. In order to resist statistical attacks, phantom terms are added to the index vector for blinding search results. Due to the use of our special tree-based index structure, the proposed scheme can achieve sub-linear search time and deal with the deletion and insertion of documents flexibly. Extensive experiments are conducted to demonstrate the efficiency of the proposed scheme.***

I. INTRODUCTION

Cloud computing has been considered as a new model of enterprise IT infrastructure, which can organize huge resource of computing, storage and applications, and enable users to enjoy ubiquitous, convenient and on-demand network access to a shared pool of configurable computing resources with great efficiency and minimal economic overhead. Attracted by these appealing features, both individuals and enterprises are motivated to outsource their data to the cloud, instead of purchasing software and hardware to manage the data themselves. Despite of the various advantages of cloud services, outsourcing sensitive information (such as e-mails, personal health records, company finance data, government documents, etc.) to remote servers brings privacy concerns. The cloud service providers (CSPs) that keep the data for users may access users' sensitive information without authorization. A general approach to protect the data confidentiality is to encrypt the data before outsourcing [2]. However, this will cause a huge cost in terms of data usability. For example, the existing techniques on keyword-based information retrieval, which

- ✓ Zhihua Xia, Xinhui Wang, and Xingming Sun are with the Jiangsu Engineering Center of Network Monitoring, Jiangsu Collaborative Innovation Center on Atmospheric Environment and Equipment Technology, and School of Computer and Software, Nanjing University of Information Science & Technology, Nanjing, China. E-mail: xia zhihua, wxh nuist, sunnudt@163.com.
- ✓ Qian Wang is with the the School of Computer, Wuhan University, Wuhan, China. E-mail: qianwang@whu.edu.cn. are widely used on the plaintext data, cannot be directly applied on the encrypted data. Downloading all the data from the cloud and decrypt locally is obviously impractical.

WHAT IS CLOUD COMPUTING?

Cloud computing is the use of computing resources (hardware and software) that are delivered as a service over a network (typically the Internet). The name comes from the common use of a cloud-shaped symbol as an abstraction for the complex infrastructure it contains in system diagrams. Cloud computing entrusts remote services with a user's data, software and computation. Cloud computing consists of hardware and software resources made available on the Internet as managed third-party services. These services typically provide access to advanced software applications and high-end networks of server computers.

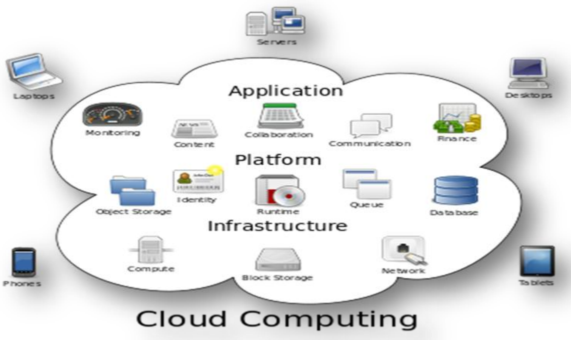

Figure 1: Structure of cloud computing

HOW CLOUD COMPUTING WORKS?

The goal of cloud computing is to apply traditional supercomputing, or high-performance computing power, normally used by military and research facilities, to perform tens of trillions of computations per second, in consumer-oriented applications such as financial portfolios, to deliver personalized information, to provide data storage or to power large, immersive computer games.

The cloud computing uses networks of large groups of servers typically running low-cost consumer PC technology with specialized connections to spread data-processing chores across them. This shared IT infrastructure contains large pools of systems that are linked together. Often, virtualization techniques are used to maximize the power of cloud computing.

CHARACTERISTICS AND SERVICES MODELS

The salient characteristics of cloud computing based on the definitions provided by the National Institute of Standards and Terminology (NIST) are outlined below:

ON-DEMAND SELF-SERVICE: A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service's provider.

- ✓ *Broad network access:* Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, laptops, and PDAs).
- ✓ *Resource pooling:* The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location-independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or data center). Examples of resources include storage, processing, memory, network bandwidth, and virtual machines.
- ✓ *Rapid elasticity*: Capabilities can be rapidly and elastically provisioned, in some cases automatically, to quickly scale out and rapidly released to quickly scale in. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be purchased in any quantity at any time.

MEASURED SERVICE: Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be managed, controlled, and reported providing transparency for both the provider and consumer of the utilized service.

Figure 2: Characteristics of cloud computing

SERVICES MODELS

Cloud Computing comprises three different service models, namely Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS). The three service models or layer are completed by an end user layer that encapsulates the end user perspective on cloud services. The model is shown in figure below. If a cloud user accesses services on the infrastructure layer, for instance, she can run her own applications on the resources of a cloud infrastructure and remain responsible for the support, maintenance, and security of these applications herself. If she accesses a service on the application layer, these tasks are normally taken care of by the cloud service provider.

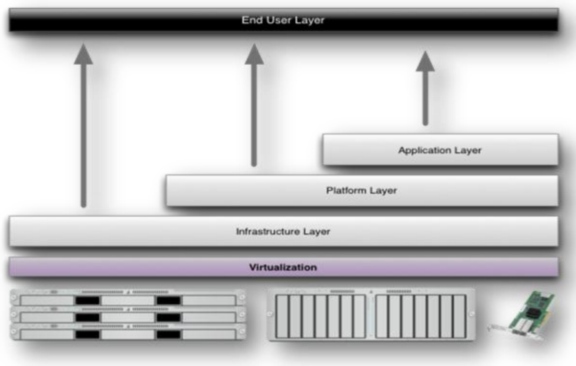

Figure 3: Structure of service models

BENEFITS OF CLOUD COMPUTING:

- ✓ Achieve economies of scale – increase volume output or productivity with fewer people. Your cost per unit, project or product plummets.

International Journal of Innovative Research and Advanced Studies (IJIRAS)
Volume 4 Issue 12, December 2017

ISSN: 2394-4404

- ✓ Reduce spending on technology infrastructure. Maintain easy access to your information with minimal upfront spending. Pay as you go (weekly, quarterly or yearly), based on demand.
- ✓ Globalize your workforce on the cheap. People worldwide can access the cloud, provided they have an Internet connection.
- ✓ Streamline processes. Get more work done in less time with less people.
- ✓ Reduce capital costs. There's no need to spend big money on hardware, software or licensing fees.
- ✓ Improve accessibility. You have access anytime, anywhere, making your life so much easier!
- ✓ Monitor projects more effectively. Stay within budget and ahead of completion cycle times.
- ✓ Less personnel training is needed. It takes fewer people to do more work on a cloud, with a minimal learning curve on hardware and software issues.
- ✓ Minimize licensing new software. Stretch and grow without the need to buy expensive software licenses or programs.
- ✓ Improve flexibility. You can change direction without serious "people" or "financial" issues at stake.

ADVANTAGES:

- ✓ Price: Pay for only the resources used.
- ✓ Security: Cloud instances are isolated in the network from other instances for improved security.
- ✓ Performance: Instances can be added instantly for improved performance. Clients have access to the total resources of the Cloud's core hardware.
- ✓ Scalability: Auto-deploy cloud instances when needed.
- ✓ Uptime: Uses multiple servers for maximum redundancies. In case of server failure, instances can be automatically created on another server.
- ✓ Control: Able to login from any location. Server snapshot and a software library lets you deploy custom instances.
- ✓ Traffic: Deals with spike in traffic with quick deployment of additional instances to handle the load.

II. SYSTEM ANALYSIS

EXISTING SYSTEM:

- ✓ A general approach to protect the data confidentiality is to encrypt the data before outsourcing.
- ✓ Searchable encryption schemes enable the client to store the encrypted data to the cloud and execute keyword search over ciphertext domain. So far, abundant works have been proposed under different threat models to achieve various search functionality, such as single keyword search, similarity search, multi-keyword boolean search, ranked search, multi-keyword ranked search, etc. Among them, multi-keyword ranked search achieves more and more attention for its practical applicability. Recently, some dynamic schemes have been proposed to support inserting and deleting operations on document collection. These are significant works as it is highly possible that the data owners need to update their data on the cloud server.

DIS-ADVANTAGES OF EXISTING SYSTEM:

- ✓ Huge cost in terms of data usability. For example, the existing techniques on keyword-based information retrieval, which are widely used on the plaintext data, cannot be directly applied on the encrypted data. Downloading all the data from the cloud and decrypt locally is obviously impractical.
- ✓ Existing System methods not practical due to their high computational overhead for both the cloud sever and user.

PROPOSED SYSTEM:

- ✓ This paper proposes a secure tree-based search scheme over the encrypted cloud data, which supports multi-keyword ranked search and dynamic operation on the document collection. Specifically, the vector space model and the widely-used "term frequency (TF) × inverse document frequency (IDF)" model are combined in the index construction and query generation to provide multi-keyword ranked search. In order to obtain high search efficiency, we construct a tree-based index structure and propose a "Greedy Depth-first Search" algorithm based on this index tree.
- ✓ The secure kNN algorithm is utilized to encrypt the index and query vectors, and meanwhile ensure accurate relevance score calculation between encrypted index and query vectors.
- ✓ To resist different attacks in different threat models, we construct two secure search schemes: the basic dynamic multi-keyword ranked search (BDMRS) scheme in the known ciphertext model, and the enhanced dynamic multi-keyword ranked search (EDMRS) scheme in the known background model.

Advantages of Proposed system:

- ✓ Due to the special structure of our tree-based index, the proposed search scheme can flexibly achieve sub-linear search time and deal with the deletion and insertion of documents.
- ✓ We design a searchable encryption scheme that supports both the accurate multi-keyword ranked search and flexible dynamic operation on document collection.
- ✓ Due to the special structure of our tree-based index, the search complexity of the proposed scheme is fundamentally kept to logarithmic. And in practice, the proposed scheme can achieve higher search efficiency by executing our "Greedy Depth-first Search" algorithm. Moreover, parallel search can be flexibly performed to further reduce the time cost of search process.

III. SYSTEM ARCHITECTURE

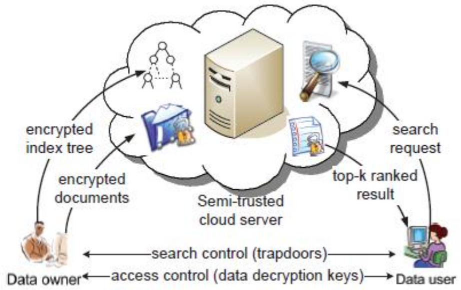

Figure 4

INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- ✓ What data should be given as input?
- ✓ How the data should be arranged or coded?
- ✓ The dialog to guide the operating personnel in providing input.
- ✓ Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES

- ✓ Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.
- ✓ It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.
- ✓ When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow.

OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

- ✓ Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.
- ✓ Select methods for presenting information.
- ✓ Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- ✓ Convey information about past activities, current status or projections of the
- ✓ Future.
- ✓ Signal important events, opportunities, problems, or warnings.
- ✓ Trigger an action.
- ✓ Confirm an action.

IV. IMPLEMENTATION

MODULES

- ✓ Users
- ✓ Known Cipher text Model
- ✓ Known Background Model
- ✓ Index Construction of UDMRS Scheme
- ✓ Search Process of UDMRS Scheme
- ✓ EDMRS Scheme

MODULE DESCRIPTION

USERS

Data owner has a collection of documents F = {f1,f2,...,fn} that he wants to outsource to the cloud server in encrypted form while still keeping the capability to search on them for effective utilization. In our scheme, the data owner firstly builds a secure searchable tree index I from document collection F, and then generates an encrypted document collection C for F. Afterwards, the data owner outsources the encrypted collection C and the secure index I to the cloud server, and securely distributes the key information of trapdoor generation (including keyword IDF values) and document decryption to the authorized data users.

Data users are authorized ones to access the documents of data owner. With t query keywords, the authorized user can generate a trapdoor TD according to search control mechanisms to fetch k encrypted documents from cloud server. Then, the data user can decrypt the documents with the shared secret key.

Cloud server stores the encrypted document collection C and the encrypted searchable tree index I for data owner. Upon receiving the trapdoor TD from the data user, the cloud server executes search over the index tree I, and finally returns the corresponding collection of top-k ranked encrypted documents.

KNOWN CIPHER TEXT MODEL

In this model, the cloud server only knows the encrypted document collection C, the searchable index tree I, and the search trapdoor TD submitted by the authorized user. That is to say, the cloud server can conduct cipher text-only attack (COA) in this model.

KNOWN BACKGROUND MODEL

Compared with known cipher text model, the cloud server in this stronger model is equipped with more knowledge, such as the term frequency (TF) statistics of the document collection. This statistical information records how many documents are there for each term frequency of a specific keyword in the whole document collection.

INDEX CONSTRUCTION OF UDMRS SCHEME

We have briefly introduced the KBB index tree structure, which assists us in introducing the index construction. In the process of index construction, we first generate a tree node for each document in the collection. These nodes are the leaf nodes of the index tree. Then, the internal tree nodes are generated based on these leaf nodes. The formal construction process of the index is presented in Algorithm 1. An example of our index tree. Note that the index tree T built here is a plaintext.

SEARCH PROCESS OF UDMRS SCHEME

The search process of the UDMRS scheme is a recursive procedure upon the tree, named as "Greedy Depth first Search (GDFS)" algorithm. We construct a result list denoted as RList, whose element is defined as ⟨RScore,FID⟩. Here, the RScore is the relevance score of the document fFID to the query, which is calculated according to Formula. The RList stores the k accessed documents with the largest relevance scores to the query. The elements of the list are ranked in descending order according to the RScore, and will be updated timely during the search process.

EDMRS SCHEME

The security analysis above shows that the BDMRS scheme can protect the Index Confidentiality and Query Confidentiality in the known cipher text model. However, the cloud server is able to link the same search requests by tracking path of visited nodes. In addition, in the known background model, it is possible for the cloud server to identify a keyword as the normalized TF distribution of the keyword can be exactly obtained from the final calculated relevance scores. The primary cause is that the relevance score calculated from Iu and TD is exactly equal to that from Du and Q. A heuristic method to further improve the security is to break such exact equality. Thus, we can introduce some tunable randomness to disturb the relevance score calculation. In addition, to suit different users' preferences for higher accurate ranked results or better protected keyword privacy, the randomness are set adjustable.

V. CONCLUSION

In this paper, a secure, efficient and dynamic search scheme is proposed, which supports not only the accurate multi-keyword ranked search but also the dynamic deletion and insertion of documents. We construct a special keyword balanced binary tree as the index, and propose a "Greedy Depth-first Search" algorithm to obtain better efficiency than linear search. In addition, the parallel search process can be carried out to further reduce the time cost. The security of the scheme is protected against two threat models by using the secure kNN algorithm. Experimental results demonstrate the efficiency of our proposed scheme. There are still many challenge problems in symmetric SE schemes. In the proposed scheme, the data owner is responsible for generating updating information and sending them to the cloud server. Thus, the data owner needs to store the unencrypted index tree and the information that are necessary to recalculate the IDF values. Such an active data owner may not be very suitable for the cloud computing model. It could be a meaningful but difficult future work to design a dynamic searchable encryption scheme whose updating operation can be completed by cloud server only, meanwhile reserving the ability to support multi-keyword ranked search. In addition, as the most of works about searchable encryption, our scheme mainly considers the challenge from the cloud server. Actually, there are many secure challenges in a multi-user scheme. Firstly, all the users usually keep the same secure key for trapdoor generation in a symmetric SE scheme. In this case, the revocation of the user is big challenge. If it is needed to revoke a user in this scheme, we need to rebuild the index and distribute the new secure keys to all the authorized users. Secondly, symmetric SE schemes usually assume that all the data users are trustworthy. It is not practical and a dishonest data user will lead to many secure problems. For example, a dishonest data user may search the documents and distribute the decrypted documents to the unauthorized ones. Even more, a dishonest data user may distribute his/her secure keys to the unauthorized ones. In the future works, we will try to improve the SE scheme to handle these challenge problems.

VI. LITERATURE SURVEY

A. SECURITY CHALLENGES FOR THE PUBLIC CLOUD

AUTHOR: K. Ren, C.Wang, Q.Wang et al.,

ABSTRACT: Cloud computing represents today's most exciting computing paradigm shift in information technology. However, security and privacy are perceived as primary obstacles to its wide adoption. Here, the authors outline several critical security challenges and motivate further investigation of security solutions for a trustworthy public cloud environment.

B. CRYPTOGRAPHIC CLOUD STORAGE

AUTHOR: Seny Kamara, Kristin Lauter

ABSTRACT: We consider the problem of building a secure cloud storage service on top of a public cloud infrastructure where the service provider is not completely trusted by the customer. We describe, at a high level, several architectures that combine recent and non-standard cryptographic primitives in order to achieve our goal. We survey the benefits such an architecture would provide to both customers and service providers and give an overview of recent advances in cryptography motivated specifically by cloud storage.

C. A FULLY HOMOMORPHIC ENCRYPTION SCHEME

AUTHOR: C. Gentry

ABSTRACT: We propose the first fully homomorphic encryption scheme, solving an old open problem. Such a scheme allows one to compute arbitrary functions over encrypted data without the decryption key—i.e., given encryptions E(m1), ..., E(mt) of m1, ..., m t, one can efficiently compute a compact ciphertext that encrypts f(m1, ..., m t) for any efficiently computable function f.

Fully homomorphic encryption has numerous applications. For example, it enables encrypted search engine queries—i.e., a search engine can give you a succinct encrypted answer to your (boolean) query without even knowing what your query was. It also enables searching on encrypted data; you can store your encrypted data on a remote server, and later have the server retrieve only files that (when decrypted) satisfy some boolean constraint, even though the server cannot decrypt the files on its own. More broadly, it improves the efficiency of secure multiparty computation.In our solution, we begin by designing a somewhat homomorphic "boostrappable" encryption scheme that works when the function f is the scheme's own decryption function. We then show how, through recursive self-embedding, bootstrappable encryption gives fully homomorphic encryption.

D. SOFTWARE PROTECTION AND SIMULATION ON OBLIVIOUS RAMS

AUTHOR: O. Goldreich and R. Ostrovsky

ABSTRACT: Software protection is one of the most important issues concerning computer practice. There exist many heuristics and ad-hoc methods for protection, but the problem as a whole has not received the theoretical treatment it deserves. In this paper, we provide theoretical treatment of software protection. We reduce the problem of software protection to the problem of efficient simulation on oblivious RAM.A machine is oblivious if the sequence in which it accesses memory locations is equivalent for any two inputs with the same running time. For example, an oblivious Turing Machine is one for which the movement of the heads on the tapes is identical for each computation. (Thus, the movement is independent of the actual input.) What is the slowdown in the running time of a machine, if it is required to be oblivious? In 1979, Pippenger and Fischer showed how a two-tape oblivious Turing Machine can simulate, on-line, a one-tape Turing Machine, with a logarithmic slowdown in the running time. We show an analogous result for the random-access machine (RAM) model of computation. In particular, we show how to do an on-line simulation of an arbitrary RAM by a probabilistic oblivious RAM with a poly logaithmic slowdown in the running time. On the other hand, we show that a logarithmic slowdown is a lower bound.

E. PUBLIC KEY ENCRYPTION WITH KEYWORD SEARCH

AUTHOR: D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano

ABSTRACT: We study the problem of searching on data that is encrypted using a public key system. Consider user Bob who sends email to user Alice encrypted under Alice's public key. An email gateway wants to test whether the email contains the keyword "urgent" so that it could route the email accordingly. Alice, on the other hand does not wish to give the gateway the ability to decrypt all her messages. We define and construct a mechanism that enables Alice to provide a key to the gateway that enables the gateway to test whether the word "urgent" is a keyword in the email without learning anything else about the email. We refer to this mechanism as Public Key Encryption with keyword Search. As another example, consider a mail server that stores various messages publicly encrypted for Alice by others. Using our mechanism Alice can send the mail server a key that will enable the server to identify all messages containing some specific keyword, but learn nothing else. We define the concept of public key encryption with keyword search and give several constructions.

REFERENCES

[1] K. Ren, C. Wang, and Q. Wang, "Security challenges for the public cloud," IEEE Internet Comput., vol. 16, no. 1, pp. 69–73, Jan-Feb. 2012.
[2] S. Kamara and K. Lauter, "Cryptographic cloud storage," in Proc. Financ. Cryptography Data Secur., 2010, pp. 136–149.
[3] C. Gentry, "A fully homomorphic encryption scheme," Ph.D. dissertation, Stanford Univ., Stanford, CA, USA, 2009.
[4] O. Goldreich and R. Ostrovsky, "Software protection and simulation on oblivious rams," J. ACM, vol. 43, no. 3, pp. 431–473, 1996.
[5] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in Proc. Adv. Cryptol.-Eurocrypt, 2004, pp. 506–522.
[6] D. Boneh, E. Kushilevitz, R. Ostrovsky, and W. E. Skeith III, "Public key encryption that allows pir queries," in Proc. Adv. Cryptol., 2007, pp. 50–67.
[7] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in Proc. IEEE Symp. Secur. Privacy, 2000, pp. 44–55.
[8] E.-J. Goh, "Secure indexes," IACR Cryptol. ePrint Archive, vol. 2003, p. 216, 2003.
[9] Y.-C. Chang and M. Mitzenmacher, "Privacy preserving keyword searches on remote encrypted data," in Proc. 3rd Int. Conf. Appl. Cryptography Netw. Secur., 2005, pp. 442–455.
[10] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: Improved definitions and efficient constructions," in Proc. 13th ACM Conf. Comput. Commun. Secur., 2006, pp. 79–88.

[11] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, “Fuzzy keyword search over encrypted data in cloud computing,” in IEEE Proc. INFOCOM, 2010, pp. 1–5.
[12] M. Kuzu, M. S. Islam, and M. Kantarcioglu, “Efficient similarity search over encrypted data,” in Proc. IEEE 28th Int. Conf. Data Eng., 2012, pp. 1156–1167.
[13] C. Wang, K. Ren, S. Yu, and K. M. R. Urs, “Achieving usable and privacy-assured similarity search over outsourced cloud data,” in Proc. IEEE INFOCOM, 2012, pp. 451–459.
[14] B. Wang, S. Yu, W. Lou, and Y. T. Hou, “Privacy-preserving multikeyword fuzzy search over encrypted data in the cloud,” in Proc. IEEE INFOCOM, 2014, pp. 2112–2120.
[15] P. Golle, J. Staddon, and B. Waters, “Secure conjunctive keyword search over encrypted data,” in Proc. Appl. Cryptography Netw. Secur., 2004, pp. 31–45.
[16] Y. H. Hwang and P. J. Lee, “Public key encryption with conjunctive keyword search and its extension to a multi-user system,” in Proc. 1st Int. Conf. Pairing-Based Cryptography, 2007, pp. 2–22.
[17] L. Ballard, S. Kamara, and F. Monrose, “Achieving efficient conjunctive keyword searches over encrypted data,” in Proc. 7th Int. Conf. Inf. Commun. Secur., 2005, pp. 414–426.
[18] D. Boneh and B. Waters, “Conjunctive, subset, and range queries on encrypted data,” in Proc. 4th Conf. Theory Cryptography, 2007, pp. 535–554.
[19] B. Zhang and F. Zhang, “An efficient public key encryption with conjunctive-subset keywords search,” J. Netw. Comput. Appl., vol. 34, no. 1, pp. 262–267, 2011.
[20] J. Katz, A. Sahai, and B. Waters, “Predicate encryption supporting disjunctions, polynomial equations, and inner products,” in Proc. Adv. Cryptol., 2008, pp. 146–162.
[21] E. Shen, E. Shi, and B. Waters, “Predicate privacy in encryption systems,” in Proc. 6th Theory oCryptography Conf. Theory Cryptography., 2009, pp. 457–473.