International Journal of Applied Sciences, Engineering and Management
ISSN 2320 – 3439, Vol. 04, No. 01, January 2015, pp. 51 – 56

Privacy-Preserving Multi-Keyword Ranked Search over Encrypted Cloud Data

K Srinivasa Rao[1], Dr Y. Vamsidhar[2]
[1]M. Tech Student, Dept of CSE, Amrita Sai Institute of Science and Technology, Paritala, Krishna-521180.
[2]Professor, Dept of CSE, Amrita Sai Institute of Science and Technology, Paritala, Krishna-521180.

Abstract: With the arrival of cloud computing, data owners are stimulated to contract out their composite data executive systems from local sites to the trade public cloud for great give and monetary savings. But for protecting data isolation, responsive data have to be encrypted before outsourcing, which obsoletes conventional data utilization based on plaintext keyword search. Thus, enabling an encrypted cloud data explore service is of dominant importance. Allowing for the large number of statistics users and entry permit in the cloud, it is required to allow multiple keywords in the search demand and return permit in the order of their significance to these keywords. Related works on searchable encryption hub on single keyword search or Boolean keyword search, and hardly ever sort the look for results. In this paper, for the first time, we characterize and solve the demanding problem of privacy-preserving multi-keyword ranked search over encrypted data in cloud computing (MRSE). We found a set of firm privacy requirements for such a secure cloud data consumption system. Among a range of multi-keyword semantics, we decide the competent similarity measure of “organize matching,” i.e., as many matches as possible, to detain the relevance of data travel permit to the search query. We additional use “inner product similarity” to quantitatively estimate such assessment measure. We first recommend a basic idea for the MRSE based on secure inner product computation, and then give two significantly improved MRSE schemes to achieve various inflexible privacy requirements in two different threat models. To improve explore experience of the data search service; we further extend these two schemes to support more search semantics. Thorough analysis investigating privacy and competence guarantees of projected schemes is given. Experiments on the real-world facts set further show planned schemes indeed commence low overhead on addition and communication.

INTRODUCTION

Disseminated computing is a country side of computer science that studies dispersed systems. A distributed system is a software scheme in which mechanism located on networked computers converse and synchronize their actions by passing messages. The mechanisms interrelate with each other in arrange to achieve a common goal. There are a lot of alternatives for the communication passing mechanism, including RPC-like connectors and message queues. Three important characteristics of disseminated systems are: concurrency of apparatus, lack of a global clock, and self-determining malfunction of workings. A significant goal and brave of distributed systems is location simplicity. Examples of distributed systems vary from SOA-based systems to especially multiplayer online playoffs to peer-to-peer applications. A computer program that runs in a disseminated system is called a distributed program, and distributed programming is the development of writing such programs. Distributed computing also refers to the use of distributed systems to solve computational problems. In distributed computing, a problem is divided into many tasks, each of which is solved by one or more computers, which communicate with each other by message passing. The word *distributed* in terms such as "distributed system", "distributed programming", and "distributed algorithm" originally referred to computer networks where individual computers were physically distributed within some geographical area. The terms are nowadays used in a much wider sense, even referring to autonomous processes that run on the same physical computer and interact with each other by message passing. While there is no single definition of a distributed system, the following defining properties are commonly used: There are several autonomous computational entities, each of which has its own local memory. The entities communicate with each other by message passing. In this article, the computational entities are called *computers* or *nodes*.

A distributed system may have a common goal, such as solving a large computational problem.[1] Alternatively, each computer may have its own user with individual needs, and the purpose of the distributed system is to coordinate the use of shared resources or provide communication services to the users.

Other typical properties of distributed systems include the following: The system has to tolerate failures in individual computers. The structure of the system (network topology, network latency, number of computers) is not known in advance, the system may consist of different kinds of computers and network links, and the system may change during the execution of a distributed program. Each computer has only a limited, incomplete view of the system. Each computer may know only one part of the input.

Distributed systems are groups of networked computers, which have the same goal for their work. The terms "concurrent computing", "parallel computing", and "distributed computing" have a lot of overlap, and no clear distinction exists between them. The same system may be characterized both as "parallel" and "distributed"; the processors in a typical distributed system run concurrently in parallel. Parallel computing may be seen as a particular tightly coupled form of distributed computing, and distributed computing may be seen as a loosely coupled form of parallel computing. Nevertheless, it is possible to roughly classify concurrent systems as "parallel" or "distributed" using the following criteria: In parallel computing, all processors may have access to a shared memory to exchange information between processors. In distributed computing, each processor has its own private memory (distributed memory). Information is exchanged by passing messages between the processors.

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.

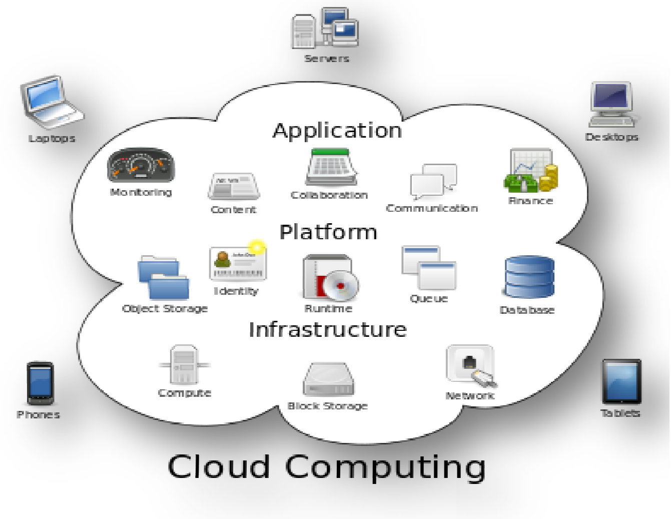


Fig: Structure of cloud computing

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.

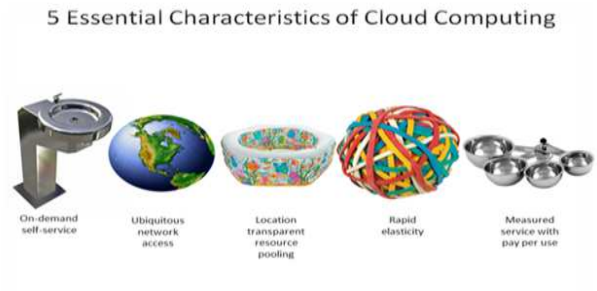

5 Essential Characteristics of Cloud Computing
On-demand self-service
Ubiquitous network access
Location transparent resource pooling
Rapid elasticity
Measured service with pay per use

Fig: Characteristics of cloud computing

Benefits of cloud computing:

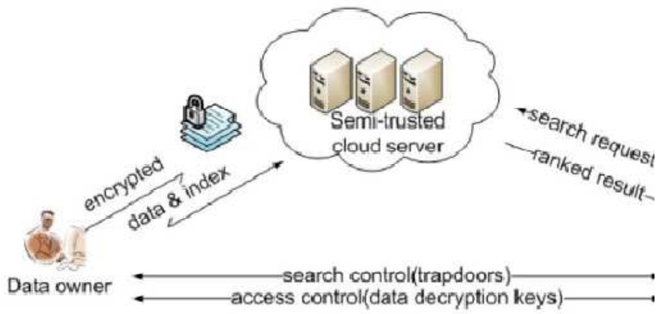
1. **Achieve economies of scale** – increase volume output or productivity with fewer people. Your cost per unit, project or product plummets.
2. **Reduce spending on technology infrastructure.** Maintain easy access to your information with minimal upfront spending. Pay as you go (weekly, quarterly or yearly), based on demand.
3. **Globalize your workforce on the cheap.** People worldwide can access the cloud, provided they have an Internet connection.
4. **Streamline processes.** Get more work done in less time with less people.
5. **Reduce capital costs.** There's no need to spend big money on hardware, software or licensing fees.
6. **Improve accessibility.** You have access anytime, anywhere, making your life so much easier!
7. **Monitor projects more effectively.** Stay within budget and ahead of completion cycle times.

8. **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.
9. **Minimize licensing new software.** Stretch and grow without the need to buy expensive software licenses or programs.
10. **Improve flexibility.** You can change direction without serious "people" or "financial" issues at stake.

Advantages:

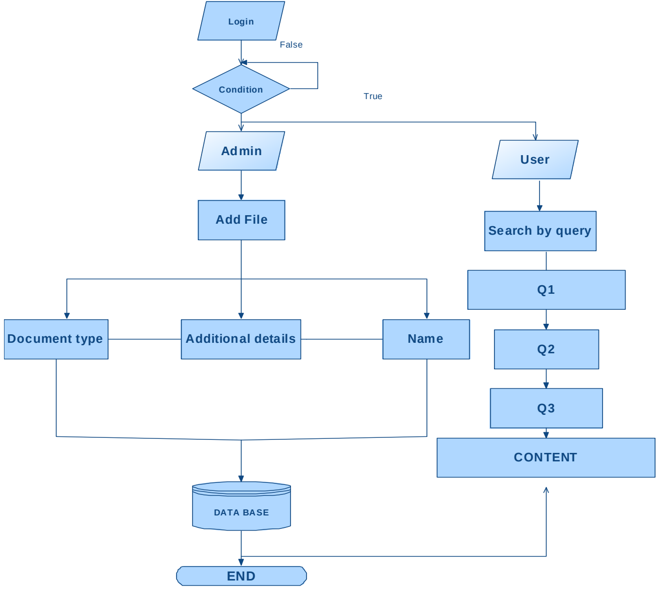
1. **Price:** Pay for only the resources used.
2. **Security**: Cloud instances are isolated in the network from other instances for improved security.
3. **Performance:** Instances can be added instantly for improved performance. Clients have access to the total resources of the Cloud's core hardware.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

SYSTEM ARCHITECTURE:


Semi-trusted cloud server
search request
ranked result
encrypted data & index
Data owner
search control(trapdoors)
access control(data decryption keys)

DATA FLOW DIAGRAM:

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.



RELATED WORK

With the advent of cloud computing, data owners are motivated to outsource their complex data management systems from local sites to the commercial public cloud for great flexibility and economic savings. But for protecting data privacy, sensitive data have to be encrypted before outsourcing, which obsoletes traditional data utilization based on plaintext keyword search. Thus, enabling an encrypted cloud data search service is of paramount importance. Considering the large number of data users and documents in the cloud, it is necessary to allow multiple keywords in the search request and return documents in the order of their relevance to these keywords. Related works on searchable encryption focus on single keyword search or Boolean keyword search, and rarely sort the search results. In this paper, for the first time, we define and solve the challenging problem of privacy-preserving multi-keyword ranked search over encrypted data in cloud computing (MRSE). We establish a set of strict privacy requirements for such a secure cloud data utilization system. Among various multi-keyword semantics, we choose the efficient similarity measure of "coordinate matching," i.e., as many matches as possible, to capture the relevance of data documents to the search query. We further use "inner product similarity" to quantitatively evaluate such similarity measure.

We first propose a basic idea for the MRSE based on secure inner product computation, and then give two significantly improved MRSE schemes to achieve various stringent privacy requirements in two different threat models. To improve search experience of the data search service, we further extend these two schemes to support more search semantics. Thorough analysis investigating privacy and efficiency guarantees of proposed schemes is given. Experiments on the real-world data set further show proposed schemes indeed introduce low overhead on computation and communication. With the increasing adoption of cloud computing for data storage, assuring data service reliability, in terms of data correctness and availability, has been outstanding. While redundancy can be added into the data for reliability, the problem becomes challenging in the “pay-as-you-use” cloud paradigm where we always want to efficiently resolve it for both corruption detection and data repair. Prior distributed storage systems based on erasure codes or network coding techniques have either high decoding computational cost for data users, or too much burden of data repair and being online for data owners. In this paper, we design a secure cloud storage service which addresses the reliability issue with near-optimal overall performance.

By allowing a third party to perform the public integrity verification, data owners are significantly released from the onerous work of periodically checking data integrity. To completely free the data owner from the burden of being online after data outsourcing, this paper proposes an exact repair solution so that no metadata needs to be generated on the fly for repaired data. The performance analysis and experimental results show that our designed service has comparable storage and communication cost, but much less computational cost during data retrieval than erasure codes-based storage solutions. It introduces less storage cost, much faster data retrieval, and comparable communication cost comparing to network coding-based distributed storage systems.

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 architecture would provide to both customers and service providers and give an overview of recent advances in cryptography motivated specifically by cloud storage. We consider the following problem: a user $\mathcal{U}$ wants to store his files in an encrypted form on a remote file server $\mathcal{S}$. Later the user $\mathcal{U}$ wants to efficiently retrieve some of the encrypted files containing (or indexed by) specific keywords, keeping the keywords themselves secret and not jeopardizing the security of the remotely stored files. For example, a user may want to store old e-mail messages encrypted on a server managed by Yahoo or another large vendor, and later retrieve certain messages while travelling with a mobile device. In this paper, we offer solutions for this problem under well-defined security requirements. Our schemes are efficient in the sense that no public-key cryptosystem is involved. Indeed, our approach is independent of the encryption method chosen for the remote files. They are also incremental, in that $\mathcal{U}$ can submit new files which are secure against previous queries but still searchable against future queries.

As Cloud Computing becomes prevalent, more and more sensitive information are being centralized into the cloud. For the protection of data privacy, sensitive data usually have to be encrypted before outsourcing, which makes effective data utilization a very challenging task. Although traditional searchable encryption schemes allow a user to securely search over encrypted data through keywords and selectively retrieve files of interest, these techniques support only exact keyword search. That is, there is no tolerance of minor typos and format inconsistencies which, on the other hand, are typical user searching behavior and happen very frequently. This significant drawback makes existing techniques unsuitable in Cloud Computing as it greatly affects system usability, rendering user searching experiences very frustrating and system efficacy very low. In this paper, for the first time we formalize and solve the problem of effective fuzzy keyword search over encrypted cloud data while maintaining keyword privacy. 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. In our solution, we exploit edit distance to quantify keywords similarity and develop an advanced technique on constructing fuzzy keyword sets, which greatly reduces the storage and representation overheads. Through rigorous security analysis, we show that our proposed solution is secure and privacy-preserving, while correctly realizing the goal of fuzzy keyword search.

SYSTEM ANALYSIS

EXISTING SYSTEM: The effective data retrieval need, the large amount of documents demand the cloud server to perform result relevance ranking, instead of returning undifferentiated results. Such ranked search system enables data users to find the most relevant information quickly, rather than burdensomely sorting through every match in the content collection. Ranked search can also elegantly eliminate unnecessary network traffic by sending back only the most relevant data, which is highly desirable in the “pay-as-you-use” cloud paradigm. For privacy protection, such ranking operation, however, should not leak any keyword related information. On the other hand, to improve the search result accuracy as well as to enhance the user searching experience, it is also necessary for such ranking system to support multiple keywords search, as single keyword search often yields far too coarse results.

DISADVANTAGES OF EXISTING SYSTEM: The encrypted cloud data search system remains a very challenging task because of inherent security and privacy obstacles, including various strict requirements. On enrich the search flexibility; they are still not adequate to provide users with acceptable result ranking functionality

PROPOSED SYSTEM: In this manuscript, for the first time, we describe and solve the difficulty of multi-keyword ranked search over encrypted cloud data (MRSE) while preserving severe system wise isolation in the cloud computing concept. Among different multi-keyword semantics, we decide the capable similarity measure of “coordinate matching,” i.e., as a lot of matches as probable, to confine the significance of data documents to the search question. Exclusively, we use “inner product similarity”, i.e., the number of query keywords appearing in a document, to quantitatively evaluate such similarity measure of that document to the search query. During the index manufacture, each manuscript is associated with a binary vector as a sub-index everywhere each bit represents whether equivalent keyword is restricted in the document. The search uncertainty is also described as a binary vector where each bit means whether corresponding keyword appears in this search request, so the similarity could be accurately measured by the inner item for consumption of the uncertainty vector with the data vector. However, honestly outsourcing the data vector or the query vector will violate the index privacy or the search privacy. To meet the challenge of supporting such multi keyword semantic without privacy breaches, we suggest a basic idea for the MRSE using secure inner product addition, which is modified from a secure k-nearest neighbour (kNN) technique, and then give two extensively improved MRSE schemes in a step-by-step manner to achieve different stringent privacy desires. Search result should be ranked by the cloud server according to some grade criteria. To reduce the communication cost.

IMPLEMENTATION MODULES

- Data Owner Module
- Data User Module
- Encryption Module
- Rank Search Module

MODULES DESCRIPTION

Data Owner Module

This module helps the owner to register those details and also include login details. This module helps the owner to upload his file with encryption using RSA algorithm. This ensures the files to be protected from unauthorized user.

Data User Module

This module includes the user registration login details. This module is used to help the client to search the file using the multiple key words concept and get the accurate result list based on the user query. The user is going to select the required file and register the user details and get activation code in mail email before enter the activation code. After user can download the Zip file and extract that file.

Encryption Module:

This module is used to help the server to encrypt the document using RSA Algorithm and to convert the encrypted document to the Zip file with activation code and then activation code send to the user for download.

Rank Search Module

These modules ensure the user to search the files that are searched frequently using rank search. This module allows the user to download the file using his secret key to decrypt the downloaded data. This module allows the Owner to view the uploaded files and downloaded files

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

1. 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.

2. 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.

3. 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.

1. 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.

2. Select methods for presenting information.

3. 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.

CONCLUSION

In this paper, for the first time we illustrate and solve the difficulty of multi-keyword ranked search over encrypted cloud data, and ascertain a variety of privacy necessities. Among a choice of multi-keyword semantics, we decide the proficient similarity determine of “coordinate matching,” i.e., as a lot of matches as possible, to efficiently capture the significance of outsourced credentials to the query keywords, and use “inner product similarity” to quantitatively estimate such correspondence measure. For convention the dispute of sustaining multi-keyword semantic without privacy breaches, we propose essential idea of MRSE using protected personal product computation. Then, we give two enhanced MRSE schemes to achieve different stringent privacy requirements in two dissimilar threat models. We also examine some further enhancements of our ranked search instrument, as well as supporting more search semantics, i.e., TF_IDF, and dynamic data operations. Thorough analysis investigating privacy and competence guarantees of projected schemes is given, and experiments on the real-world data set show our proposed schemes commence low overhead on both computation and communication. In our future work, we will look at inspection the integrity of the rank order in the search result assume the cloud server is un trusted.

REFERENCES

[1] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, “Privacy-Preserving Multi-Keyword Ranked Search over Encrypted Cloud Data,” Proc. IEEE INFOCOM, pp. 829-837, Apr, 2011.

[2] L.M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, “A Break in the Clouds: Towards a Cloud Definition,” ACM SIGCOMM Comput. Commun. Rev., vol. 39, no. 1, pp. 50-55, 2009.

[3] N. Cao, S. Yu, Z. Yang, W. Lou, and Y. Hou, “LT Codes-Based Secure and Reliable Cloud Storage Service,” Proc. IEEE INFOCOM, pp. 693-701, 2012.

[4] S. Kamara and K. Lauter, “Cryptographic Cloud Storage,” Proc. 14th Int’l Conf. Financial Cryptograpy and Data Security, Jan. 2010.

[5] A. Singhal, “Modern Information Retrieval: A Brief Overview,” IEEE Data Eng. Bull., vol. 24, no. 4, pp. 35-43, Mar. 2001.

[6] I.H. Witten, A. Moffat, and T.C. Bell, Managing Gigabytes: Compressing and Indexing Documents and Images. Morgan Kaufmann Publishing, May 1999.

[7] D. Song, D. Wagner, and A. Perrig, “Practical Techniques for Searches on Encrypted Data,” Proc. IEEE Symp. Security and Privacy, 2000.

[8] E.-J. Goh, “Secure Indexes,” Cryptology ePrint Archive, http:// eprint.iacr.org/2003/216. 2003.

[9] Y. C. Chang and M. Mitzenmacher, “Privacy Preserving Keyword Searches on Remote Encrypted Data,” Proc. Third Int’l Conf. Applied Cryptography and Network Security, 2005.

[10] R. Curtmola, J.A. Garay, S. Kamara, and R. Ostrovsky, “Searchable Symmetric Encryption: Improved Definitions and Efficient Constructions,” Proc. 13th ACM Conf. Computer and Comm. Security (CCS ’06), 2006.

[11] D. Boneh, G.D. Crescenzo, R. Ostrovsky, and G. Persiano, “Public Key Encryption with Keyword Search,” Proc. Int’l Conf. Theory and Applications of Cryptographic Techniques (EUROCRYPT), 2004.

[12] M. Bellare, A. Boldyreva, and A. ONeill, “Deterministic and Efficiently Searchable Encryption,” Proc. 27th Ann. Int’l Cryptology Conf. Advances in Cryptology (CRYPTO ’07), 2007.

[13] 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,” J. Cryptology, vol. 21, no. 3, pp. 350- 391, 2008.

[14] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, “Fuzzy Keyword Search Over Encrypted Data in Cloud Computing,” Proc. IEEE INFOCOM, Mar. 2010.

[15] D. Boneh, E. Kushilevitz, R. Ostrovsky, and W.E.S. III, “Public Key Encryption That Allows PIR Queries,” Proc. 27th Ann. Int’l Cryptology Conf. Advances in Cryptology (CRYPTO ’07), 2007.

[16] P. Golle, J. Staddon, and B. Waters, “Secure Conjunctive Keyword Search over Encrypted Data,” Proc. Applied Cryptography and Network Security, pp. 31-45, 2004.

[17] L. Ballard, S. Kamara, and F. Monrose, “Achieving Efficient Conjunctive Keyword Searches over Encrypted Data,” Proc. Seventh Int’l Conf. Information and Comm. Security (ICICS ’05), 2005.

About the Authors:

Mr. K Srinivas Rao is a student of Amrita Sai Institute of Science And Technology, Paritala, Krishna Dt, Andhra Pradesh. His areas of interest are Cloud Computing and Network Security.

Dr. Y Vamsi Dhar is working as an Professor and Head of Computer Science And Engineering of Mr. K Srinivas Rao is a student of Amrita Sai Institute of Science And Technology, Paritala, Krishna Dt, Andhra Pradesh.