International Journal of Advanced Research in

Computer Science Engineering and Information Technology

Volume: 2 Issue: 1 08-Feb-2014,ISSN_NO: 2321-3337

Keyword Ranked MetaData Indexed Object Clawler in Cloud Servers

R.Bakyalakshmi[1] , Dr.R.Rameh[2]

PG Student of computer science and engineering.
Chennai institute of technology College, India [1].
Associate.Professor, Department.of Computer Science,
Chennai institute of technology College, India[2]
bakyalakshmiraja@gmail.com , ramesh@citchennai.net.

.

ABSTRACT— *Ranked search greatly enhances system usability by enabling search result relevance ranking instead of sending undifferentiated results, and further ensures the file retrieval accuracy. The statistical measure approach, i.e., relevance score, from information retrieval to build a secure searchable index, and develop a one-to-many order-preserving mapping technique to properly protect those sensitive score information. The resulting design is able to facilitate efficient server-side ranking without losing keyword privacy. Thorough analysis shows that our proposed solution enjoys "as strong-as-possible" security guarantee compared to previous searchable encryption schemes, while correctly realizing the goal of ranked keyword search. In this system provides B-Tree conceptual data storage associated with easy indexing and fastest searching option on the dynamic ranked data. Extensive experimental results demonstrate the efficiency of the proposed solution.*

Keywords— Cloud, data privacy, ranking, similarity relevance, homomorphic encryption, vectorspace model.

1. INTRODUCTION

1.1SYNOPSIS

Cloud Computing is the long dreamed vision of computing as a utility, where cloud customers can remotely store their data into the cloud so as to enjoy the on-demand high-quality applications and services from a shared pool of configurable computing resources. The benefits brought by this new computing model include but are not limited to: relief of the burden for storage management, universal data access with independent geographical locations, and avoidance of capital expenditure on hardware, software, and personnel maintenances, etc.,

As Cloud Computing becomes prevalent, more and more sensitive information are being centralized into the cloud, such as e-mails, personal health records, company finance data, and government documents, etc. The fact that data owners and cloud server are no longer in the same trusted domain may put the outsourced unencrypted data at risk : the cloud server may leak data information to unauthorized entities or even be hacked

Data encryption makes effective data utilization a very challenging task given that there could be a large amount of outsourced data files. Besides, in Cloud Computing, data owners may share their outsourced data with a large number of users, who might want to only retrieve certain specific data files they are interested in during a given session. One of the most popular ways to do so is through keyword-based search. Such keyword search technique allows users to selectively retrieve files of interest and has been widely applied in plaintext search scenarios.

Traditional searchable encryption schemes allow a user to securely search over encrypted data through keywords without first decrypting it, these techniques support only conventional Boolean keyword search, without capturing any relevance of the files in the search result. When directly applied in large collaborative data outsourcing cloud environment, they may suffer from the following two main drawbacks.

For the first time, Here define the problem of secure ranked keyword search over encrypted cloud data, and provide such an effective protocol, which fulfills the secure ranked search functionality with little relevance score information leakage against keyword privacy. Thorough security analysis shows that our ranked searchable symmetric encryption scheme indeed enjoys “as-strong-as-possible” security guarantee compared to previous searchable symmetric encryption (SSE) schemes.

Investigate the practical considerations and enhancements of ranked search mechanism, including the efficient support of relevance score dynamics, the authentication of ranked search results, and the reversibility of proposed one to- many order-preserving mapping technique. Extensive experimental results demonstrate the effectiveness and efficiency of the proposed solution.

2. RANKED SEARCH SYMMETRIC ENCRYPTION

To enable ranked searchable symmetric encryption for effective utilization of outsourced and encrypted cloud data under the aforementioned model, This system design should achieve the following security and performance guarantee.

2.1 Ranked keyword search:

To explore different mechanisms for designing effective ranked search schemes based on the existing searchable encryption framework;

2.2 Security guarantee:

To prevent cloud server from learning the plaintext of either the data files or the searched keywords, and achieve the “as-strong-as-possible” security strength compared to existing searchable encryption schemes;

2.3 Efficiency:

Above goals should be achieved with minimum communication and computation overhead.

To authenticate a ranked search result ,one need to ensure:

1) the retrieved results are the most relevant ones;

2) the relevance sequence among the results are not disrupted.

To achieve this two authentication requirements, we propose to utilize the one way hash chain technique, which can be added directly on top of the previous RSSE design some cryptographic one-way hash function, such as SHA-1. Our mechanism requires one more secret value u in the Setup phase to be generated and shared between data owner and users.

3. SYSTEM ANALYSIS

System Analysis is a combined process dissecting the system responsibilities that are based on the problem domain characteristics and user requirements.

3.1 EXISTING SYSTEM:

Ranked search greatly enhances system usability by enabling search result relevance ranking instead of sending undifferentiated results, and further ensures the file retrieval accuracy. Ranked Searchable encryption allows data owner to outsource his data in an

encrypted manner while maintaining the selectively search capability over the encrypted data. In the existing system, the documents and data is stored into a secure cloud storage whereas the documents were scanned with the keywords and an index of information were stored for future searching options. The existing system have used symmetric encryption algorithm to store the data securely.

Disadvantages:

- Secure searchable encryption scheme does not perform any functions when there occurs new updates in files or when any modifications are performed.
- The relevance score algorithm is not updated frequently when there are some modifications in the owner files.

3.2 PROPOSED SYSTEM:

In Cloud Computing, outsourced file collection might not only be accessed but also updated frequently for various application purposes. Hence, supporting the score dynamics in the searchable index for a secure storage engine which is reflected from the corresponding file collection updates, is thus of practical importance. In our system, we consider score dynamics as adding newly encrypted scores for newly created files, or modifying old encrypted scores for modification of existing files in the file collection. Symmetric key encryption doesn’t have major scope in security perspective that’s why we are opting MD5 encryption algorithm which is bit more complex when compared to the traditional algorithms in storing the data. B Tree indexing and storing of data provides a peak level performance in searching times.

Advantages:

To solve the problem of supporting efficient ranked keyword search for achieving effective utilization of remotely stored encrypted data in Cloud Computing.

4. IMPLEMENTATION

Implementation is the stage of the project when the theoretical design is turned out into a working system.

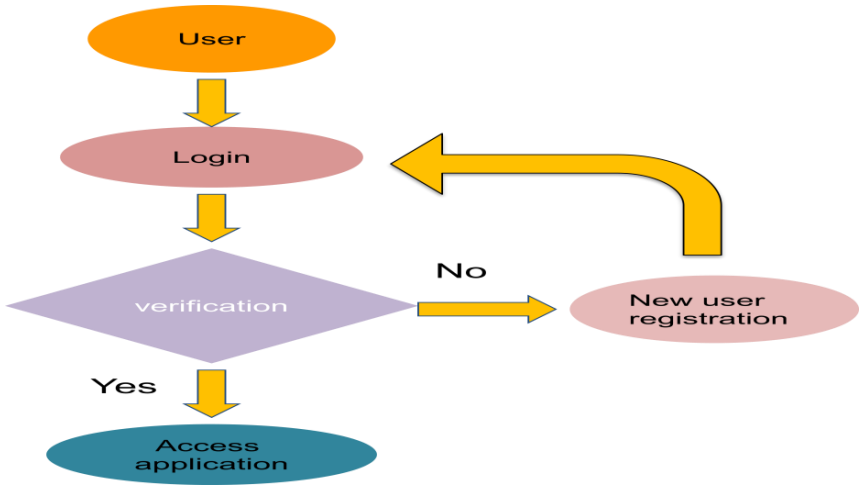
4.1 MODULE DESCRIPTION:

1. AUTHENTICATION MODULE
2. DOCUMENT UPLOAD MODULE
3. DOCUMENT PARSE MODULE
4. LIBRARY VIEW MODULE
5 DOCUMENT SEARCH MODULE

4.1.1. AUTHENTICATION MODULE:

Authentication Module describes the interface between the user and system and the admin provided the type of authentication. The user is allowed to create his testimonial to login into the system.

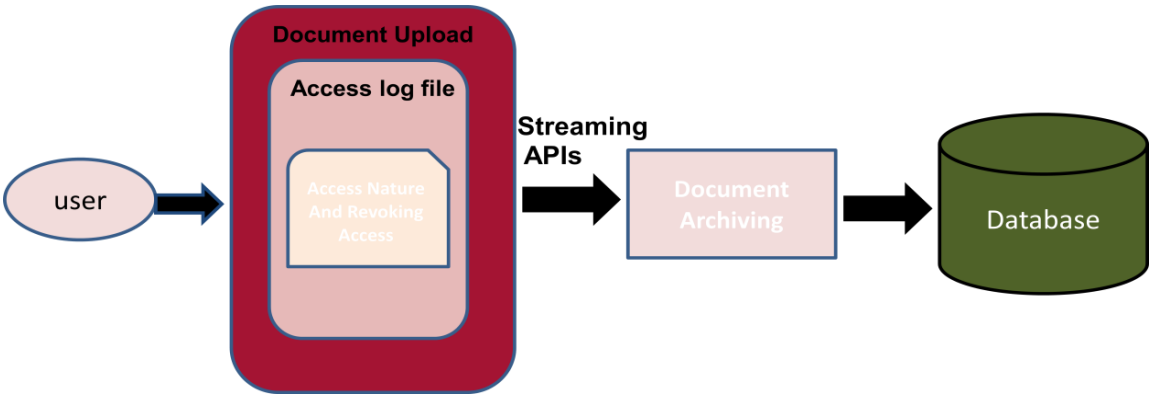
An admin needs to approve the users created and login approval the user will be allowed to access the application. Authentication is provided by encrypting the user name and password. Protecting sensitive information from users.



4.1.2 DOCUMENT UPLOAD MODULE:

In this module, the users are allowed to upload their documents. While uploading, the user is allowed to provide the access nature on the document. The documents will be archived with the help of rich streaming APIs which integrates the SQL Server Database Engine with an NTFS file system by storing varbinary (max) binary large object (BLOB) data as files on the file system. In addition, the user will be given the

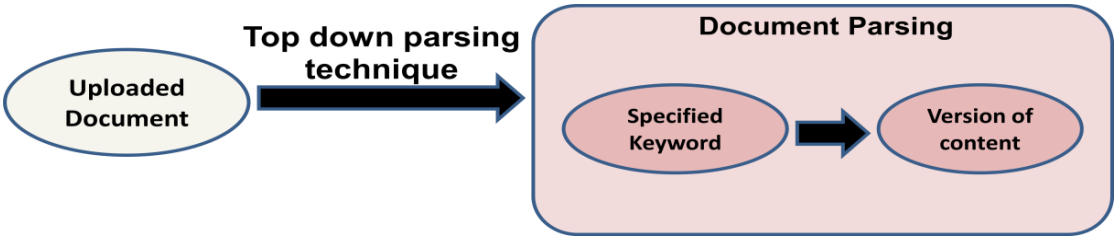
option of revoking the access at any time. Based on the provided accessibility, the documents will be accessed by the other users. Internally, the access details will be logged in the Access log file.



4.1.3 DOCUMENT PARSE MODULE:

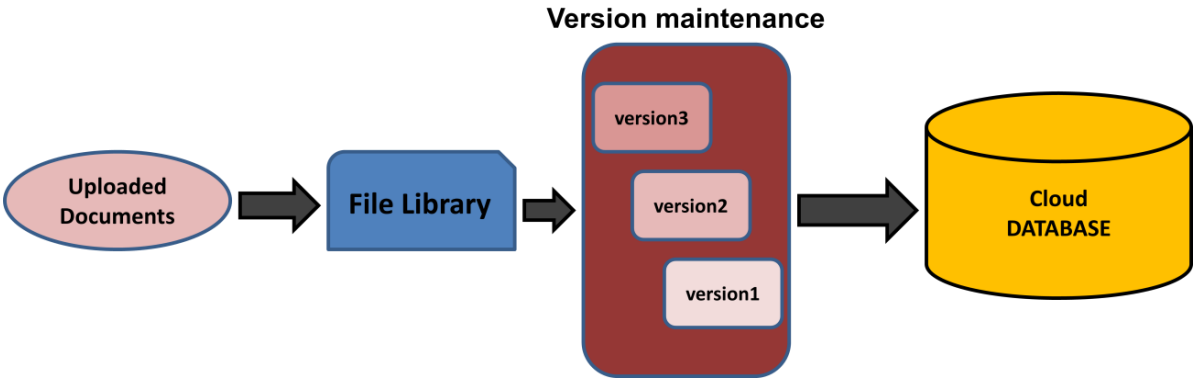
In this module the uploaded document is parsed by using the document parser interface.

A mechanism with the top-down keyword parsing technique reads the complete document with specified keywords. In addition it matches the version of the content type definition that is used by a list or document library.



4.1.4 LIBRARY VIEW MODULE:

The major advantage of this is related to maintaining the history of the documents. All the files that are uploaded are stored in the library. Various versions of the documents were maintained by the cloud database. The data owner will be provided an option of taking back the previous version documents. This is one of the major jargons specified in our proposed system**.**



4.1.5 DOCUMENT SEARCH MODULE:

In this module, the secondary users can search for the documents and access the documents based on the user provided access. The search is made very accurate using a keyword .An automatic pure logging of access details which includes time, access personnel details, document details were loaded.

5. SOFTWARE DESCRIPTION

5.1. Features Of .Net

Microsoft .NET is a set of Microsoft software technologies for rapidly building and integrating XML Web services, Microsoft Windows-based applications, and Web solutions. The .NET Framework is a language-neutral platform for writing programs that can easily and securely interoperate. There's no language barrier with .NET: there are numerous languages available to the developer including Managed C++, C#, Visual Basic and Java Script. The .NET framework provides the foundation for components to interact seamlessly, whether locally or remotely on different platforms. It standardizes common data types and communications protocols so that components created in different languages can easily interoperate.

".NET" is also the collective name given to various software components built upon the .NET platform. These will be both products (Visual Studio.NET and Windows.NET Server, for instance) and services (like Passport, .NET My Services, and so on).

5.2. THE .NET FRAMEWORK

The .NET Framework has two main parts:

1. The Common Language Runtime (CLR).

2. A hierarchical set of class libraries.

The CLR is described as the "execution engine" of .NET. It provides the environment within which programs run. The most important features are

♦ Conversion from a low-level assembler-style language, called Intermediate Language (IL), into code native to the platform being executed on.

♦ Memory management, notably including garbage collection.

- Checking and enforcing security restrictions on the running code.
- Loading and executing programs, with version control and other such features.
- The following features of the .NET framework are also worth description:

Managed Code

The code that targets .NET, and which contains certain extra Information - "metadata" - to describe itself. Whilst both managed and unmanaged code can run in the runtime, only managed code contains the information that allows the CLR to guarantee, for instance, safe execution and interoperability.

Managed Data

With Managed Code comes Managed Data. CLR provides memory allocation and Deal location facilities, and garbage collection. Some .NET languages use Managed Data by default, such as C#, Visual Basic.NET and JScript.NET, whereas others, namely C++, do not. Targeting CLR can, depending on the language you're using, impose certain constraints on the features available. As with managed and unmanaged code, one can have both managed and unmanaged data in .NET applications - data that doesn't get garbage collected but instead is looked after by unmanaged code.

Common Type System

The CLR uses something called the Common Type System (CTS) to strictly enforce type-safety. This ensures that all classes are compatible with each other, by describing types in a common way. CTS define how types work within the runtime, which enables types in one language to interoperate with types in another language, including cross-language exception handling. As well as ensuring that types are only used in appropriate ways, the runtime also ensures that code doesn't attempt to access memory that hasn't been allocated to it.

Common Language Specification

The CLR provides built-in support for language interoperability. To ensure that you can develop managed code that can be fully used by developers using any programming language, a set of language features and rules for using them called the Common Language Specification (CLS) has been defined. Components that follow these rules and expose only CLS features are considered CLS-compliant

6. CONCLUSION

6.1 SUMMARY:

In this paper, as an initial attempt, To motivate and solve the problem of supporting efficient ranked keyword search for achieving effective utilization of remotely stored encrypted data in Cloud Computing. The first give a basic scheme and show that by following the same existing searchable encryption framework, it is very inefficient to achieve ranked search and then appropriately weaken the security guarantee, resort to the newly developed crypto primitive OPSE, and derive an efficient one-to-many order preserving mapping function, which allows the effective RSSE to be designed.

6.2 FUTURE ENHANCEMENT:

In this paper investigate some further enhancements of ranked search mechanism, including the efficient support of relevance score dynamics, the authentication of ranked search results, and the reversibility of proposed one-to-many order-preserving mapping technique

.

REFERENCES

[1] C. Wang, N. Cao, J. Li, K. Ren, and W. Lou, "Secure Ranked Keyword Search over Encrypted Cloud Data," Proc. IEEE 30th Int'l Conf. Distributed Computing Systems (ICDCS '10), 2010.

[2] P. Mell and T. Grance, "Draft Nist Working Definition of Cloud Computing," http://csrc.nist.gov/groups/SNS/cloudcomputing/ index.html, Jan. 2010. 12 IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 23, NO. X, XXX 2012

[3] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R.H. Katz, A. Konwinski, G. Lee, D.A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the Clouds: A Berkeley View of Cloud Computing," Technical Report UCB-EECS-2009-28, Univ. of California, Berkeley, Feb. 2009.

[4] Cloud Security Alliance "Security Guidance for Critical Areas of Focus in Cloud Computing," http://www.cloudsecurityalliance. org, 2009.

[5] Z. Slocum, "Your Google Docs: Soon in Search Results?" http:// news.cnet.com/8301-17939_109-10357137-2.html, 2009.

[6] B. Krebs, " Payment Processor Breach May Be Largest Ever in cloud computing ," http://voices.washingtonpost.com/securityfix/2009/01/payment_processor_breach_may_b.html, Jan. 2009.

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

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