International Journal of Advances in Engineering and Management (IJAEM)
Volume 4, Issue 6 June 2022, pp: 679-684 www.ijaem.net ISSN: 2395-5252

Dynamic Mechanism to Preserve Cloud Storage Data from Unauthorized Access using Time Stamp

Ms. Alisha Sayyed[*1], Prof.Vinayak V. Palmur[*2]

[*1]*Master of Engineering student, Department of Computer Science and Technology, V.V.P.I.E.T. Solapur, Maharashtra, India.*

[*2]*HOD, Department of Computer Science and Engineering, V.V.P.I.E.T. Solapur, Maharashtra, India.*

Submitted: 01-06-2022 Revised: 10-06-2022 Accepted: 15-06-2022

ABSTRACT: Dynamic Searchable symmetric encryption (DSSE) allows a party to outsource the storage of his data to another party in a private manner while maintaining the ability to selectively search over it. This problem has been a focus of active research and several security definitions and constructions have been proposed. Dynamic searchable symmetric encryption (DSSE) is a useful cryptographic tool in encrypted cloud storage, But it usually suffers from the search, access, and size patterns, it also leaks (during searches) the document identifiers that were deleted in the past and match the keyword content leak of deleted documents and file-injection attacks. In addition to satisfying stronger security guarantees, our constructions are more efficient than all previous constructions. we enhance the existing IM-DSSE protocol with a more secured time stamp protocol to have a secure data sharing protocol in Cloud and reduce the privacy risks involved in the system. This system proposes a dynamic key that an adversary/attacker cannot able to predict/crack. The dynamic Secret key is generated using the Time Stamp protocol. Only the approved party has the authority on having an encrypted secret key which is generated using our proposed time stamp protocol of the client for cloud storage and updates under the encrypted state in each time duration. An adversary cannot form any substantiation in any period even if it gets the decryption secret key by attacking the client.

Keyword : Privacy-enhancing technologies, private cloud services; dynamic searchable symmetric encryption.

I. INTRODUCTION

What is cloudcomputing?

Cloud computing is the use of computing resources (hardware and software) that are delivered as a service over a network (typically the Internet). Instead of storing information on your computer's hard drive or another local storage device, you save it to a remote database. The Internet provides the connection between your computer and the database. 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.

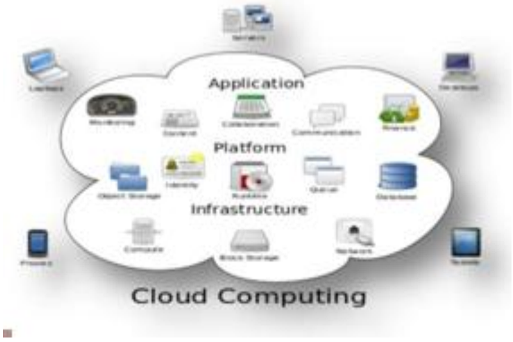


Fig1:Structure of cloud computing

How does Cloud Computing work?

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 computergames.

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 cloudcomputing.

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'sprovider.
- **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, andPDAs).
- **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 utilizedservice.

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 layers are completed by an end-user layer that encapsulates the end-user perspective on cloud services. The model is shown in the figure below. If a cloud user accesses services on the infrastructure layer, for instance, she can run her applications on the resources of 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 serviceprovider.

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 ondemand.
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 fewer people.
5. **Reduce capital costs.** There's no need to spend big money on hardware, software, or licensingfees.
6. **Improve accessibility.** You have access anytime, anywhere, making your life so mucheasier!
7. **Monitor projects more effectively.** Stay within budget and ahead of completion cycletimes.
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 softwareissues.
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.

II. SECURE SEARCH IN THE CLOUD:

Searchable Symmetric Encryption (SSE) enables a client to encrypt data in such a way that they can later perform keyword searches on it. These encrypted queries are performed via “Search Tokens" over an encrypted index which represents the relationship between search tokens (keywords) and encrypted files. A prominent application of SSE is to enable privacy- preserving keyword search on the cloud (e.g., Amazon S3), where a data owner can outsource a collection of encrypted files and perform keyword searches on it without revealing the file and query contents. Preliminary SSE schemes only provide search- only functionality on static data (i.e., no dynamism), which strictly limits their applicability due to the lack of update capacity. Later, several Dynamic Searchable Symmetric Encryption (DSSE) schemes were proposed that permit the user to add and delete files after the system is set up. we provide the full-fledged implementation of our preliminary DSSE scheme proposed, as well as extended schemes, which are specially designed to meet various application requirements and cloud data storage-as-a-service infrastructures inpractice.

III. LITERATURE SURVEY

D. X. Song, D. Wagner, A. Perrig, and others are desirable to store data on data storage servers such as mail servers and file servers in encrypted form to reduce security and privacy risks. But this usually implies that one has to sacrifice functionality for security. For example, if a client wishes to retrieve only documents containing certain words, it was not previously known how to let the data storage server perform the search and answer the query, without loss of data confidentiality. We describe our cryptographic schemes for the problem of searching for encrypted data and provide proofs of security for the resulting cryptosystems. Our techniques have several crucial advantages. They are provably secure: they provide provable secrecy for encryption, in the sense that the untrusted server cannot learn anything about the plaintext when only given the ciphertext; they provide query isolation for searches, meaning that the untrusted server cannot learn anything more about the plaintext than the search result; they provide controlled searching so that the untrusted server cannot search for an arbitrary word without the user's authorization; they also support hidden queries, so that the user may ask the untrusted server to search for a secret word without revealing the word to the server. The algorithms presented are simple, fast (for a document of length n, the encryption and search algorithms only need O(n) stream cipher and block cipher operations), and introduce almost no space and communication overhead, and hence are practical to usetoday[1].

N. Cao, C. Wang, M. Li, K. Ren, and W. Lou 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 a 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 measures. 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 the search experience of the data search service, we further extend these two schemes to support more search semantics. A thorough analysis investigating the privacy and efficiency guarantees of proposed schemes is given. Experiments on the real- world data set further show that proposed schemes indeed introduce low overhead on computation and communication[2].

W. Sun, B. Wang, N. Cao, M. Li, W. Lou, Y. T. Hou, and H. Li With the growing popularity of cloud computing, a huge amount of documents are outsourced to the cloud for reduced management cost and ease of access. Although encryption helps protect user data confidentiality, it leaves the well-functioning yet practically-efficient secure search functions over encrypted data a challenging problem. In this paper, we present a verifiable privacy-preserving multi-keyword text search (MTS) scheme with similarity-based ranking to address this problem. To support multi-keyword search and search result ranking, we propose to build the search index based on term frequency- and the vector space model with cosine similarity measure to achieve higher search result accuracy. To improve the search efficiency, we propose a tree-based index structure and various adaptive methods for a multi-dimensional (MD) algorithm so that the practical search efficiency is much better than that of linear search. To further enhance search privacy, we propose two secure index schemes to meet the stringent privacy requirements under strong threat models, i.e., the known ciphertext modeland the known background model. In addition, we devise a scheme upon the proposed index tree structure to enable authenticity check over the returned search results. Finally, we demonstrate the effectiveness and efficiency of the proposed schemes through extensive experimentalevaluation[3].

M. Naveed, M. Prabhakaran, and C. A. Gunter's Dynamic Searchable Symmetric Encryption allow a client to store a dynamic collection of encrypted documents with a server, and later quickly carry out keyword searches on these encrypted documents, while revealing minimal information to the server. In this paper we present a new dynamic SSE scheme that is simpler and more efficient than existing schemes while revealing less information to the server than prior schemes, achieving fully adaptive security against honest-but-curious servers. We implemented a prototype of our scheme and demonstrated its efficiency on datasets from prior work. Apart from its concrete efficiency, our scheme is also simpler: in particular, it does not require the server to support any operation other than upload and download of data. Thus the server in our scheme can be based solely on a cloud storage service, rather than a cloud computation service as well, as in prior work. In building our dynamic SSE scheme, we introduce a new primitive called Blind Storage, which allows a client to store a set of files on a remote server in such a way that the server does not learn how many files are stored, or the lengths of the individual files, as each file is retrieved, the server learns about its existence (and can notice the same file being downloaded subsequently), but the file's name and contents are not revealed. This is a primitive with several applications other than SSE and is of independent interest[4].

Z. Fu, X. Wu, C. Guan, X. Sun, K. Ren, and others are used Keyword-based search over encrypted outsourced data has become an important tool in the current cloud computing scenario. The majority of the existing techniques are focusing on a multi-keyword exact match or single keyword fuzzy search. However, those existing techniques find less practical significance in real-world applications compared with the multi- keyword fuzzy search technique over encrypted data. The first attempt to construct such a multi-keyword fuzzy search scheme was reported by Wang et al., who used locality-sensitive hashing functions and Bloom filtering to meet the goal of the multi-keyword fuzzy search. Nevertheless, Wang's scheme was only effective for a one-letter mistake in keyword but was not effective for other common spelling mistakes. Moreover, Wang's scheme was vulnerable to server out- of-order problems during the ranking process and did not consider the keyword weight. In this paper, based on Wang et al.'s scheme, we propose an efficient multi-keyword fuzzy ranked search scheme based on Wang et al.'s scheme that can address the aforementioned problems. First, we develop a new method of keyword transformation based on the uni- gram, which will simultaneously improve the accuracy and creates the ability to handle other spelling mistakes. In addition, keywords with the same root can be queried using the stemming algorithm. Furthermore, we consider the keyword weight when selecting an adequate matching file set. Experiments using real-world data show that our scheme is practically efficient and achieves high accuracy[5].

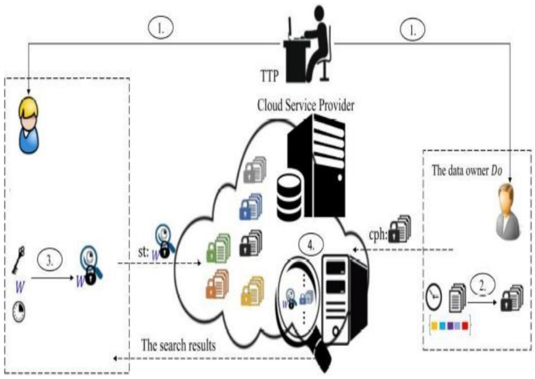
IV. OBJECTIVE:

The project is to provide a dynamic key that an adversary/attacker cannot able to predict/crack. The existing IM-DSSE protocol with a more secured time stamp protocol to have a secure data sharing protocol in Cloud and reduce the privacy risks involved in thesystem.

We have the following objectives:

I. To provide highly secure against File-InjectionAttacks
II. To provide less processingtime
III. To provide Updates with ImprovedFeature
IV. Fully Parallelized
V. Detailed experimental evaluation and open-sourceframework

V. SYSTEM ARCHITECTURE:



INPUT DESIGN AND OUTPUT DESIGN

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 the correct information from the computerized system. It is achieved by creating user-friendly screens for the data entry to handle the 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 manipulation 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 to 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

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 into 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 that it provides security and ease of use with retaining privacy. Input Design considered the followingthings:

- ➢ What data should be given asinput?
- ➢ How the data should be arranged orcoded?
- ➢ The dialog guides the operating personnel in providinginput.
- ➢ Methods for preparing input validations and steps to follow when errorsoccur.

OUTPUT DESIGN

Quality output is one, which meets the requirements of the end-user and presents the information. In any system results of processing are communicated to the users and another system through outputs. In output design, it is determined how the informationistobedisplacedforimmediate need and also the hard copy output. It is the most important and direct source of information for 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 analyzing design computer output, they should Identify the specific output that is needed to meet the requirements.
2. Select methods for presenting the information.
3. Create documents, reports, or other formats that contain information produced by thesystem.

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 ofthe
- ❖ Future.
- ❖ Signal important events, opportunities, problems, orwarnings.
- ❖ Trigger anaction.
- ❖ Confirm anaction.

VI. CONCLUSION

In this article, we presented IM-DSSE, a new DSSE framework that offers very high privacy, efficient updates, and low search latency simultaneously. Our constructions rely on a simple yet efficient incidence matrix data structure in combination with two hash tables that allow efficient and secure search and update operations. Our framework offers various DSSE constructions, which are specifically designed to meet the needs of cloud infrastructure and personal usage in different applications and environments. All of our schemes in the IM-DSSE framework are proven to be secure and achieve the highest privacy among their counterparts. We conducted a detailed experimental analysis to evaluate the performance of our schemes on real Amazon EC2 cloud systems. Our results showed the high practicality of our framework, even when deployed on mobile devices with large datasets. We have released the full-fledged implementation of our framework for public use andanalysis.

REFERENCES

[1] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, “Searchable symmetric encryption: improved definitions and efficient constructions,” in Proc. 13th ACM Conf. Comput. Commun. security, ser. CCS ’06. ACM, 2006, pp.79–88.

[2] E. Stefanov, C. Papamanthou, and E. Shi, "Practical dynamic searchable encryption with a small leakage," in 21st Annu. Network and Distributed System Security Symp. — NDSS 2014. The Internet Soc., February 23-26, 2014.

[3] S. Kamara, C. Papamanthou, and T. Roeder, “Dynamic searchable symmetric encryption,” in Proc. 2012 ACM Conf. Comput. Commun. security. New York, NY, USA: ACM, 2012, pp. 965–976.

[4] D. X. Song, D. Wagner, and A. Perrig, “Practical techniques for searches on encrypted data,” in Proc. 2000 IEEE Symp. Security and Privacy, 2000, pp.44–55.

[5] D. Cash, J. Jaeger, S. Jarecki, C. Jutla, H. Krawcyk, M.-C. Rosu, and M. Steiner, “Dynamic searchable encryption in very-large databases: Data structures and implementation,” in 21th Annu. Network Distributed System Security Symp. — NDSS 2014. The Internet Soc., February 23-26,2014.

[6] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, “Privacy-preserving multi-keyword ranked search over encrypted cloud data,” IEEE Trans. Parallel Distributed Syst., vol. 25, no. 1, pp. 222–233,2014.

[7] W. Sun, B. Wang, N. Cao, M. Li, W. Lou, Y. T. Hou, and H. Li, “Verifiable privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking,” IEEE Trans. Parallel Distributed Syst., vol. 25, no. 11, pp. 3025–3035,2014.

[8] S. Kamara and C. Papamanthou, “Parallel and dynamic searchable symmetric encryption,” in Financial Cryptography and Data Security (FC), ser. Lecture Notes in Comput. Sci. Springer Berlin Heidelberg, 2013, vol. 7859, pp. 258–274.