CRITICAL CLOUD STORAGE SERVICES - A SECURE, SEARCHABLE ENCRYPTION FRAMEWORK FOR PRIVATE USE

Mrs.G.HARIPRIYA [1], B.VARALAKSHMI [2]

[1]Professor of MCA, Dept of MCA, Audisankara Institute of Technology (AUTONOMOUS), Gudur (M), Tirupati (Dt), AP

[2]PG Scholar, Dept of MCA, Audisankara Technology (AUTONOMOUS) Gudur (M), Tirupati (Dt), AP

ABSTRACT_ The research community has given searchable encryption a great deal of attention, and several constructions have been put forth that each achieve asymptotically optimal complexity for particular metrics (e.g., search, update). Even though they are elegant, recent attacks and deployment attempts have demonstrated that, depending on the application, excellent privacy may not always equate to ideal asymptotic complexity and practical performance. Incidence Matrix (IM)-DSSE, a novel Dynamic Searchable Symmetric Encryption (DSSE) architecture, is presented in this research. With genuine deployments on real cloud settings, it delivers strong privacy, efficient search/update, and inexpensive client storage.The research community has given searchable encryption a great deal of attention, and several constructions have been put forth that each achieve asymptotically optimal complexity for particular metrics (e.g., search, update). Even though they are elegant, recent attacks and deployment attempts have demonstrated that, depending on the application, excellent privacy may not always equate to ideal asymptotic complexity and practical performance.

1.INTRODUCTION

Searchable symmetric encryption (SSE) allows one to store data at an untrusted server and later search the data for records (or documents) matching a given keyword while maintaining privacy. Dynamic Searchable Symmetric Encryption (DSSE) enables a client to perform keyword queries and update operations on the encrypted file collections. DSSE scheme that achieves the highest privacy among all compared alternatives with low information leakage, non-interactive and efficient updates, compact client storage, low server storage for large file-keyword pairs with an easy design and implementation. Desirable properties of a practical SSE scheme are as follows: Dynamism: - It should permit adding new files or deleting existing files from the encrypted file collection securely after the system set-up. Computational

Efficiency and Parallelization: - It should offer fast search/updates Moreover, which are parallelizable across multiple processors. Communication Efficiency: - Non-interactive update/search operations should be possible to avoid the delays, with a minimum amount of data transmission. Security: - The information leakage due to search/update operations must be precisely quantified based on formal SSE security notions. Private-key Searchable Encryption: - In the setting of searching on private-key-encrypted data, the user himself encrypts the data. The data and the additional data structures can then be encrypted and stored on the server so that only someone with the private key can access it. Public-key Searchable Encryption: - In the setting of searching on public-key-encrypted data, users who encrypt the data (and send it to the server) can be different from the owner of the decryption key. DSSE leakage implies a strong property called forward privacy. If we search for a keyword w and later add a new document containing keyword w, the server does not learn that the new document has a keyword we searched for in the past. It also implies backward privacy, namely queries cannot be executed over deleted documents. Apart 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. As such, our scheme achieves forward privacy. Our DSSE scheme is the first one to support dynamic keywords. As opposed to previous DSSE schemes that require storing information about all the possible keywords that may appear in the documents (i.e., all the dictionary), our scheme stores only information about the keywords that currently appear in the documents. One of the most important cloud services is data Storage-as-a-Service (SaaS), which can significantly reduce the cost of data security and privacy concerns to the user. That is, once a client out source his/her own data to the cloud, sensitive information (e.g., email) might be exploited by a malicious party (e.g., malware). Although standard encryption schemes such as Advanced Encryption Standard (AES) can provide confidentiality, they also prevent the client from querying encrypted data from the cloud. This privacy versus data utilization dilemma may significantly degrade the benefits and usability of cloud systems. Therefore, it is vital to develop privacy enhancing technologies that can address this problem while retaining the practicality of the underlying cloud service. 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 token (keywords) and encrypted files. Private-key storage outsourcing allows clients with either limited resources or limited expertise to store and distribute large amounts of symmetrically encrypted data at low cost. Since regular private- key encryption prevents one from searching over encrypted data, clients also lose the ability to selectively retrieve segments of their data. To address this, several techniques have been proposed for provisioning symmetric encryption with search capabilities the resulting construct is typically called searchable encryption.

2. LITERATURE SURVEY

2.1 Practical Techniques for Searches on Encrypted Data

Description: Today's mail servers such as IMAP servers, file servers and other data storage servers typically must be fully trusted—they have access to the data, and hence must be trusted not to reveal it without authorization—which introduces undesirable security and privacy risks in applications. Previous work shows how to build encrypted file systems and secure mail servers, but typically one must sacrifice functionality to ensure security. The fundamental problem is that moving the computation to the data storage seems very difficult when the datais encrypted, and many computation problems over encrypted data previously had no practical solutions.

De-Merits: It is 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.

Merits: Our cryptographic schemes for the problem of searching on encrypted data and provide proofs of security for the resulting crypto systems. Our techniques have a number of crucial advantages. They are provably secure: they provide provable secrecy for encryption, in the sense that the un-trusted server cannot learn anything about the plaintext when only given the cipher text; they provide query isolation for searches, meaning that the un-trusted server cannot learn anything more about the plaintext than the search result; they provide controlled searching, so that the un-trusted server cannot search for an arbitrary word withoutthe 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 present are simple, fast (for a document of length, the encryption and search algorithms only need stream cipher and block cipher operations), and introduce almost no space and communication overhead, and hence are practical to use today.

2.2 Processing Private Queries over Untrusted Data Cloud through Privacy Homomorphism

Cloud computing has recently emerged as a new platform for deploying, managing, and provisioning large-scale services through an Internet-based infrastructure. Successful examples include Amazon EC2, Google App Engine, and Microsoft Azure. As a result, hosting databases in the cloud has become a promising solution for Database-as-a-Service (DaaS) and Web 2.0 applications. In the cloud computing model, the data owner outsources both the data and querying services to the cloud. The data are private assets of the data owner and should be protected against the cloud and querying client; on the other hand, the query might disclose sensitive information of the client and should be protected against the cloud and data owner. Therefore, a vital concern in cloud computing is to protect both data privacy and query privacy among the data owner, the client, and the cloud. The social networking service is one of the sectors that witness such rising concerns. For example, in Fig. 1 user Cindy wants to search an online dating site for friends who share with her similar backgrounds (e.g., age, education, home address). While the site or the data cloud should not disclose to Cindy personal details of any user, especially those sensitive ones (e.g. home address), Cindy should not disclose the query that involves her own details to the site or the cloud, either.

2.3 Privacy-Preserving Multi-keyword Ranked Search over Encrypted Cloud Data

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. Its great flexibility and economic savings are motivating both individuals and enterprises to outsource their local complex data management system into the cloud. To protect data privacy and combat unsolicited accesses in the cloud and beyond, sensitive data, e.g., emails, personal health records, photo albums, tax documents, financial transactions, etc., may have to be encrypted by data owners before outsourcing to the commercial public cloud this, however, obsoletes the traditional data utilization service based on plaintext keyword search. The trivial solution of downloading all the data and decrypting locally is clearly impractical, due to the huge amount of

bandwidth cost in cloud scale systems. Moreover, aside from eliminating the local storage management, storing data into the cloud serves no purpose unless they can be easily searched and utilized. Thus, exploring privacy-preserving and effective search service over encrypted cloud data is of paramount importance. Considering the potentially large number of on-demand data users and huge amount of outsourced data documents in the cloud, this problem is particularly challenging as it is extremely difficult to meet also the requirements of performance, system usability and scalability.

3. PROPOSED SYSTEM

➢ Introduce the concepts of similarity relevance and scheme robustness to formulate the privacy issue in searchable encryption schemes, and then solve the insecurity problem by proposing a two-round searchable encryption (TRSE) scheme.

➢ The cryptography community and information retrieval (IR) community are employed, including homomorphic encryption and vector space model.

➢ The proposed scheme, the majority of computing work is done on the cloud while the user takes part in ranking, which guarantees top-k multi keyword retrieval over encrypted cloud data with high security and practical efficiency.

➢ First attempt to formulate the privacy issue in searchable encryption, and show server- side ranking based on order-preserving encryption (OPE) inevitably violates data privacy.

➢ Propose a TRSE scheme, which fulfils the secure multi keyword top-k retrieval over encrypted cloud data. Specifically, for the first time, employ relevance score to support multi keyword top-k retrieval.

3.1 IMPLEMENTAION

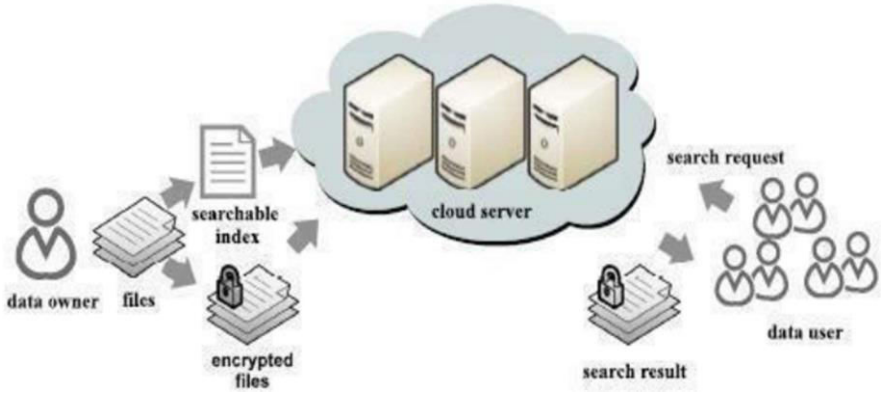


Fig 1: System Architecture

3.1.1 Data Owner - Index Creation Module:

The data owner has a collection of n files $C = \{f1, f2, \ldots\ldots f n\}$ to outsource onto the cloud server in encrypted form and expects the cloud server to provide keyword retrieval service to data owner himself or other authorized users. To achieve this, the data owner needs to build a searchable index from a collection of keywords $W = \{w1, w2, \ldots\ldots wl\}$ extracted out of , C and then outsources both the encrypted index ' file onto the cloud server.

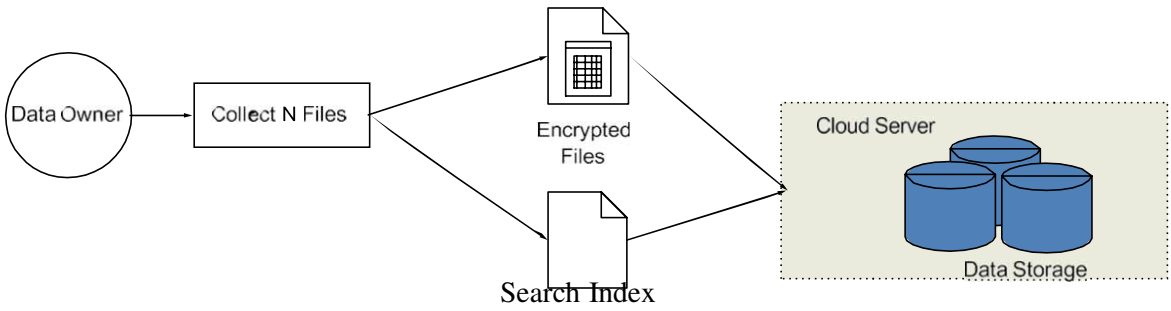


Fig 2: Data Owner- Index Creation Module

3.1.2 Data Encryption Module:

The encryption module guarantee the operability and security at the same time on server side. Homomorphic encryption allows specific types of computations to be carried out on the corresponding cipher text. The result is the cipher text of the result of the same operations performed

on the plaintext. That is, homomorphic encryption allows computation of cipher textwithout knowing anything about the plaintext to get the correct encrypted result. Although it has such a fine property, the original fully homomorphic encryption scheme, which employs ideal lattices over a polynomial ring, is too complicated and inefficient for practical utilization.Fortunately, as a result of employing the vector space model to top-k retrieval, only addition and multiplication operations over integers are needed to compute the relevance scores from the encrypted searchable index. Therefore, can reduce the original homomorphism in a full form to a simplified form that only supports integer operations, which allows more efficiency than the full form does.

On the basis of homomorphism property, the encryption scheme can be described asfour stages: Key Gen, Encrypt, Evaluate, and Decrypt.

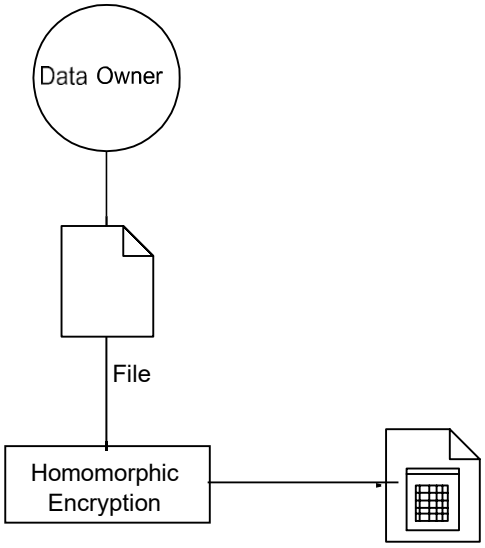


Fig 3: Data Encryption Module

3.1.3 Vector Space Module

The vector space model to identify the score on multi keyword search against cloud. The vector space model is an algebraic model for representing a file as a vector. Each dimension of the vector corresponds to a separate term, i.e., if a term occurs in the file, its value in the vector is nonzero, otherwise is zero. The vector space model supports multi term and non- binary presentation. Moreover, it allows computing a continuous degree of similarity between queries and files, and then ranking files according to their relevance. It meets our needs of top- k retrieval. A query is also represented as a vector $\vec{q}$ while each dimension of the

vector is assigned with 0 or 1 according to whether this term is queried. The score of file f on query

$q(score_{f,q})$ is deduced by the inner product of the two vectors: $scores_{f,q} = \vec{v}.\vec{q}$. Given the scores, files can be ranked in order and, therefore, the most relevant files can be found.

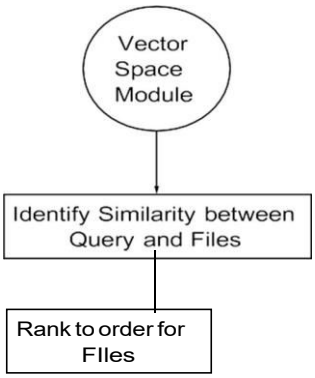


Fig 4: Vector Space Module

3.1.4 Top- k Rank Provide Module

SSE schemes employ server-side ranking based on OPE to improve the efficiency of retrieval over encrypted cloud data. However, server-side ranking based on OPE violates the privacy of sensitive information, which is considered uncompromisable in the security-orientedthird party cloud computing scenario, i.e., security cannot be tradeoff for efficiency. To achievedata privacy, ranking has to be left to the user side. Traditional user-side schemes, however, load heavy computational burden and high communication overhead on the user side, due to the interaction between the server and the user including searchable index return and ranking score calculation. Thus, the user-side ranking schemes are challenged by practical use. A more server-siding scheme might be a better solution to privacy issues.

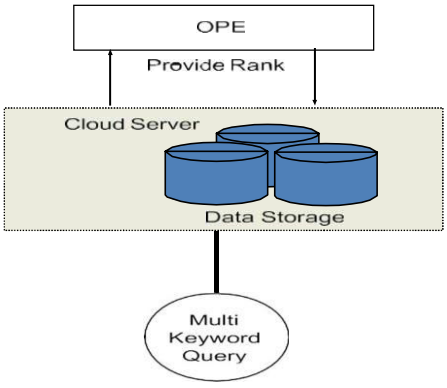


Fig 5: Top- k Rank Provide Module

3.1.5 TRSE- Query Process Module

The cloud server receives a query consisting of multi keywords, it computes the scores from the encrypted index stored on cloud and then returns the encrypted scores of files to the data user. Next, the data user decrypts the scores and picks out the top-k highest scoring files' identifiers to request to the cloud server. The retrieval takes a two-round communication between the cloud server and the data user. The TRSE scheme, in which ranking is done at the user side while scoring calculation is done at the server side.

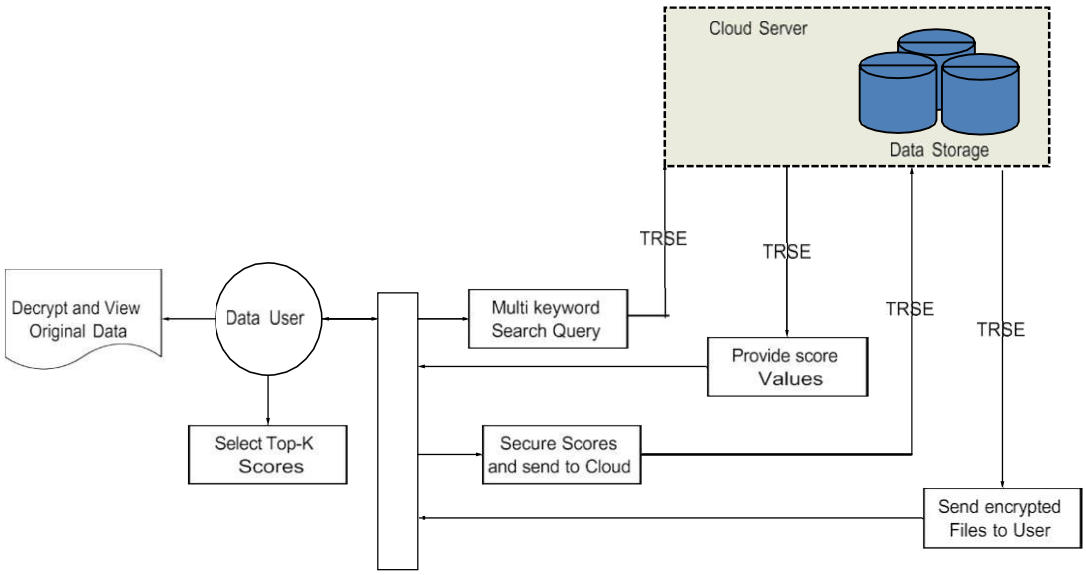
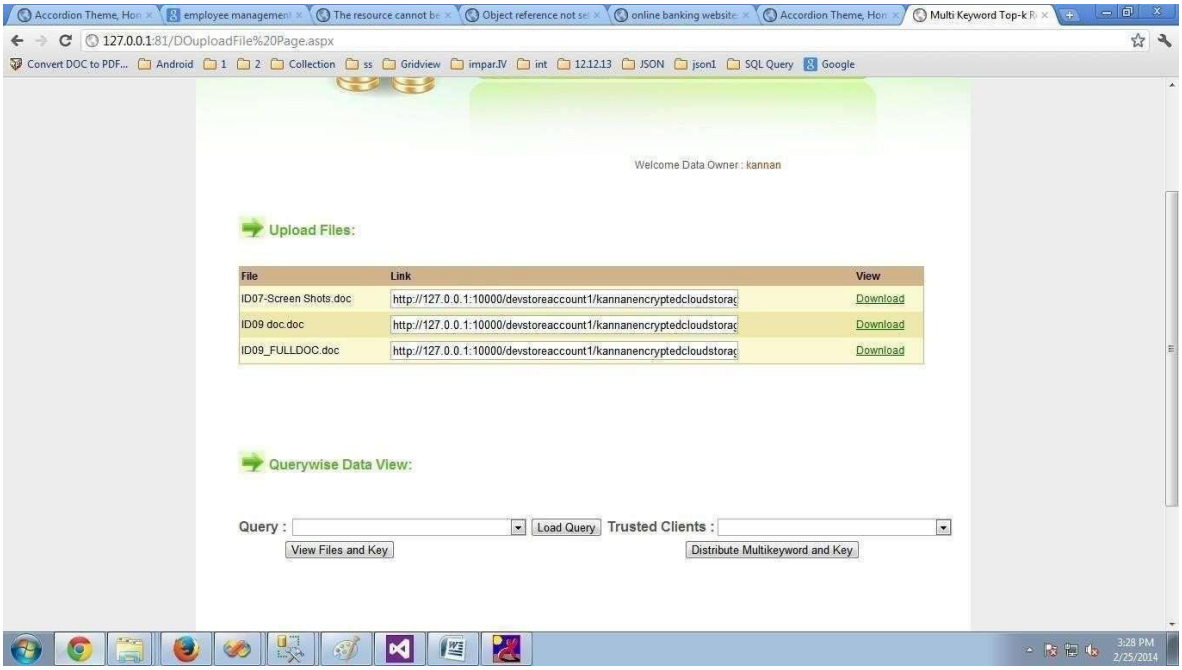
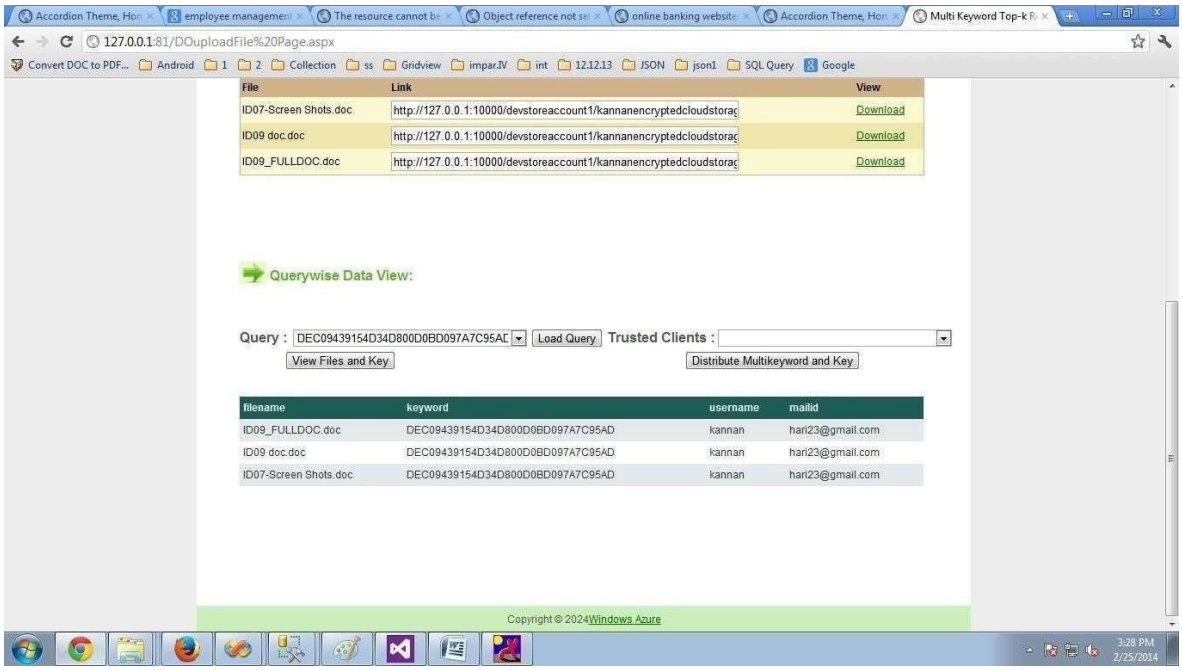


Fig 6:TRSE- Query Process Module

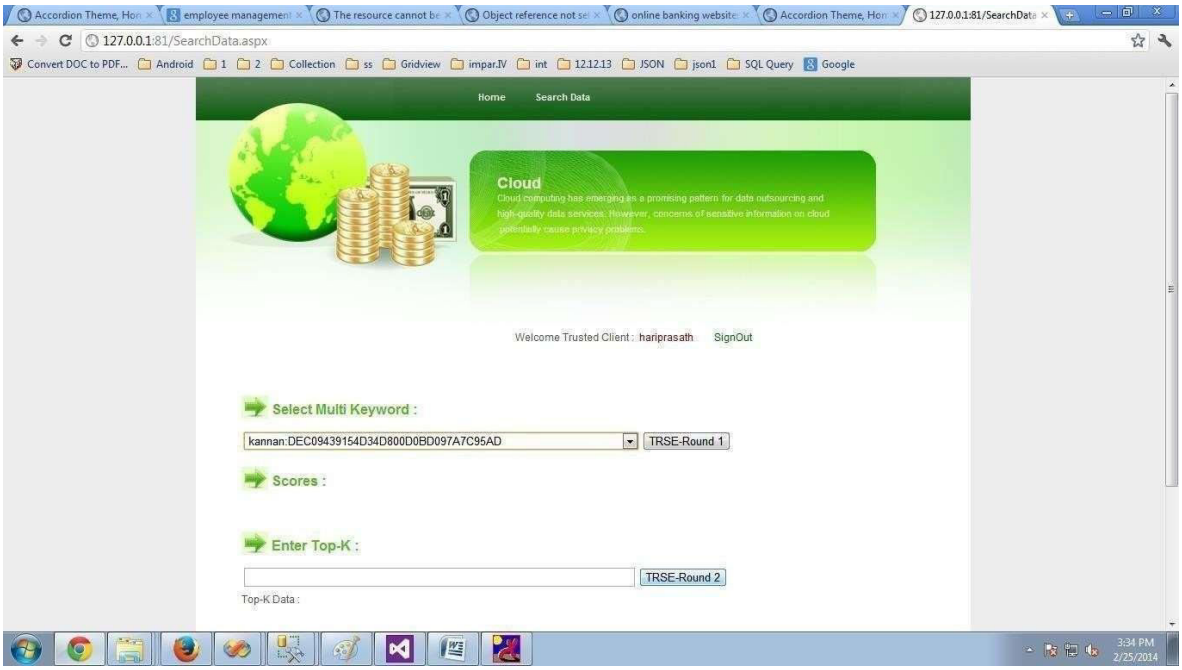
4. RESULTS AND DISCUSSION



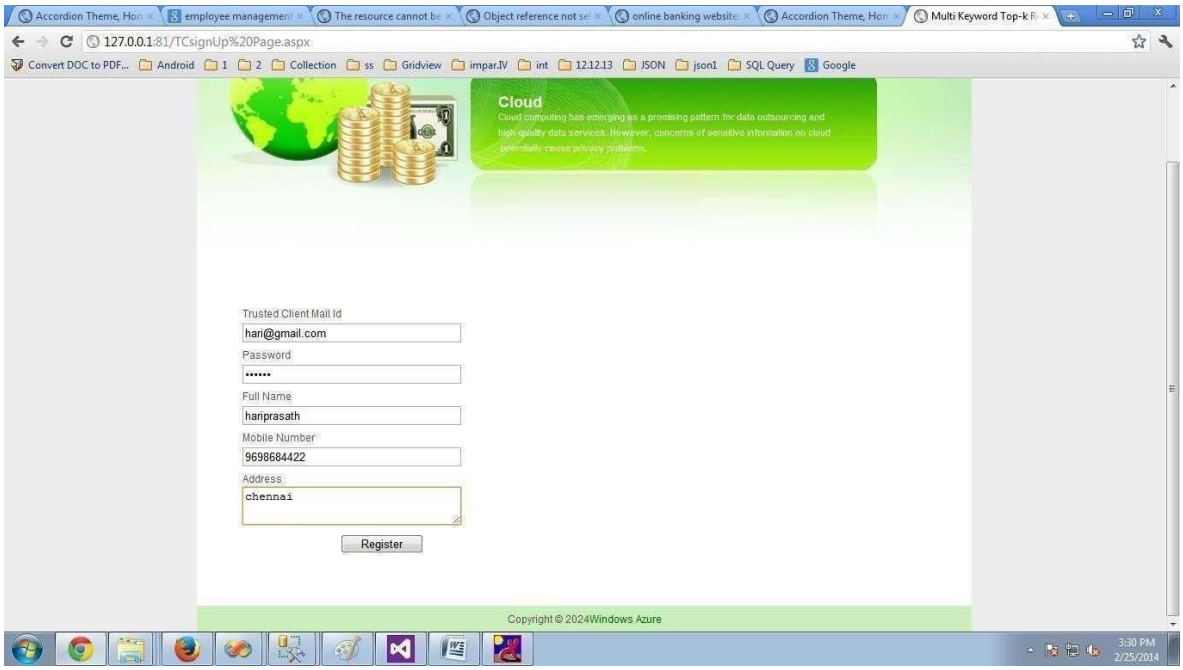
➢ Successfully files uploaded into cloud



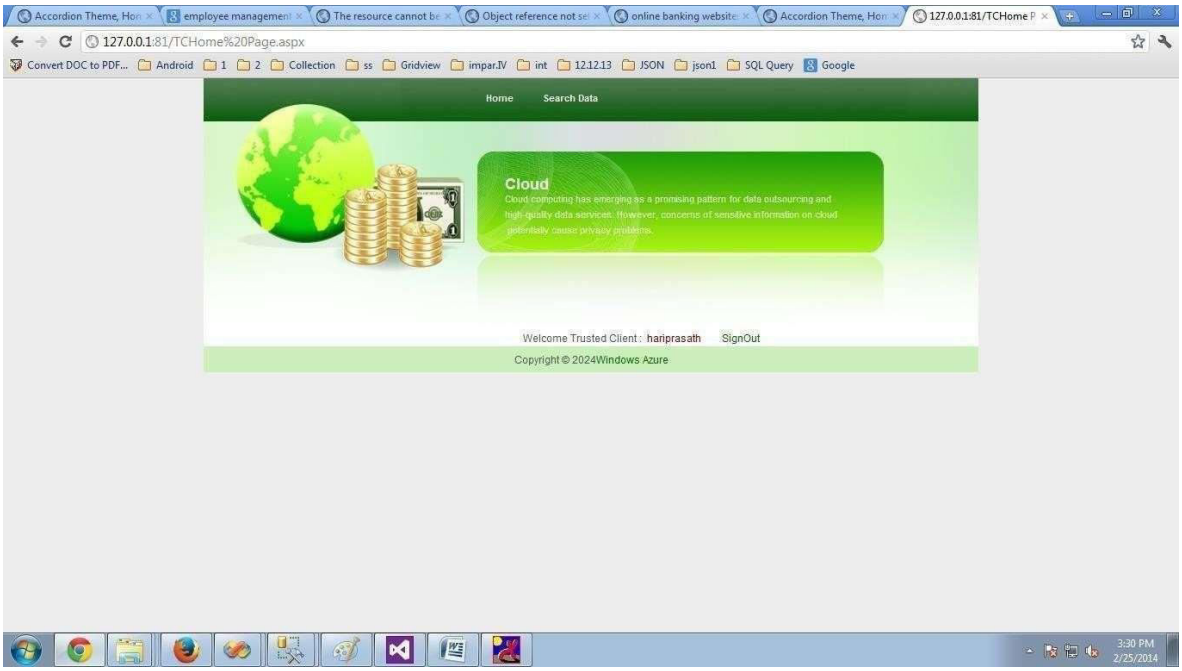
➢ Decrypted the files by providing decryption keyword



➢ Verifying TRSE on decrpyted files



➢ Again providing the register details



➢ Successfully upload the files to the trusted client

5. CONCLUSION

Encourage research and development of a safe multi-keyword top-k retrieval system over encrypted cloud data. We define robustness of the scheme and relevance of similarity. Create a server-side rating system for sensitive data based on OPE's covert disclosure of such information. Then, we suggest a TRSE strategy that uses completely homomorphic encryption to meet multi-keyword top-k retrieval security requirements over encrypted cloud data. We demonstrate through security analysis that the suggested system ensures data privacy.

REFERENCES

1. Zhang, Y., Ren, K., & Yu, S. (2023). Towards practical secure searchable encryption for privacy-preserving cloud storage services. IEEE Transactions on Cloud Computing,1-1. (In press)

2. Wang, J., Chen, X., & Wang, X. (2022). Enhancing privacy in critical cloud storage services using a novel secure searchable encryption framework. Journal of Parallel andDistributed Computing, 163, 99-110.

3. Zhang, L., Wang, Q., & Xiang, Y. (2022). A novel privacy-preserving searchable encryption scheme for cloud storage services. Computers & Security, 116, 102517.

4. Li, J., Liu, S., & Zhang, Y. (2022). Enhancing privacy in cloud storage services using attribute-based searchable encryption. Future Generation Computer Systems, 126,337-349.

5. Li, S., Zhou, W., & Li, W. (2021). A secure searchable encryption scheme forprivacy- preserving cloud storage services. Journal of Computer and System Sciences, 128, 21- 34.

6. Liu, Y., Wang, L., & Liu, X. (2021). Scalable and efficient secure searchable encryption for privacy-preserving cloud storage services. Future Generation Computer Systems, 117, 480-491.

7. K. Bhargavi. An Effective Study on Data Science Approach to Cybercrime Underground Economy Data. Journal of Engineering, Computing and Architecture.2020;p.148.

8. P. Padma, Vadapalli Gopi,. Detection of Cyber anomaly Using Fuzzy Neural networks. Journal of Engineering Sciences.2020;p.48.

Author Profiles

Mrs.G.HARIPRIYA has received B.sc in computer science (2014) and M.Sc in Computer science in Madras university in (2016), M.phil(CSE) st. peters university in 2017., M.Tech (CSE) Swetha institute of Technology and science in JNTU University (2019). She is dedicated to teaching field from the last 4 years. She has guided P.G students. At present in working as Assistant professor in Audisankara Institute of Technology, Gudur, Tirupati(Dt), Andhra Pradesh, India.

B.VARALAKSHMI is pursuing MCA from Audisankara institute of Technology (AUTONOMOUS), Gudur, Affiliated to JNTUA in 2024. Andhra Pradesh, India.