Data Deduplication and Enhanced Cloud Security Using Searchable Encryption

D. Jackson Prabu[1], C.Subash[2], R.Kingsy Grace[3]
Department of Computer Science and Engineering
[1, 2, 3] Sri Ramakrishna Engineering College, Coimbatore - 641 022, INDIA

Abstract-*In this paper, we describe our cryptographic method for the problem of searching on encrypted data and provide maximu security for the resulting crypto systems. Our technique have a number of crucial advantages. They are provably secure: they provide provable security for encryption, in the sense that the untrusted server cannot know anything about the text when only given the cipher-text; they provide query isolation for search, meaning that the untrusted servers 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 the user may ask the untrusted servers to search for a secret word without knowing to the server . The algorithm we present are simple, fast and introduce almost no space and hence are practical to use today.*

Keywords- Dup-Adj, Data Deduplication, Searchable Encryption

I. INTRODUCTION

Today' s mail servers such as IMAP servers, file servers and other data storage server typically must be trusted. they have access to the datas, and hence must be trusted not to reveal it which introduces undesirable security and privacy risk in application. Cloud computing is similar to the computer networks which includes more than computing resource commonly referred as a server and the computing resource is connected through a communication network such as an internet, an intranet, local area network (LAN)or wide area network(WAN).

Instead of using personal computer for every time to applications, we can use the cloud to run the applications of the users from anywhere at any time and the processing power for the application is provided by the cloud servers. European countries mainly get serves from cloud computer facility in European business hours with a specific application (e.g., email)and also it provides same service to the North American countries in their business hours with a different application(e.g., a web server).

Cloud computing increases the computer usages and reduce the power and also reduces the space needed to maintain the personal computers. Multiple users can save, retrieve and update their own information with single server is only because of cloud computing concept. It also allows multiple users to access different applications.

The main aim of the cloud computer is too maximize the uses of shared resources. In olden days, much organization moving from capex model to opex model to reduce the use of more dedicated PC for every organizations and it makes the organizations to use the shared resource and we can pay only for our use. It reduces the maintaining and managing cost for PC‟s in the organization.

The algorithms we presents are simple and fast. More specifically, for a document of length n, the encryptions and search algorithm only needs O(n) number of streams cipher and block cipher operations. Our method introduce essentially no spaces and communications over-head. They are also flexible and can be easily extended to supports more advanced searches. Our methods all take the form of probabilistic searching: a search for the word W returns all the position where W occurs in the plaintext, as well as possibly some other erroneous position.

we increase the security and privacy of the cloud datas and we effectively retrieves large amount of datas on demand from cloud and provide the relevance results based on search query instead of getting unwanted results. These search system enables data user to finds the most relevant information quickly rather than perform some sorting through every match in the content collections. The ranked search also reduce the network traffic by back only the most relevant datas. For privacy ranking operation does not leak any informations.

II.RELATED WORKS

Yuan et al. [8] proposed a deduplication system in the cloud storage to reduce the storage size of the tags for integrity check. Kaaniche, N [10] proposed client-side deduplication and implements Symmetric encryption for enciphering the data files and, Asymmetric encryption for

metadata filesTo enhance the security of deduplication and protect the data confidentiality, Li et al. [4] addressed the key management issue in block-level deduplication by distributing these keys across multiple servers after encrypting the files. Bugiel et al. [2] provided an architecture consisting of twin clouds for secure outsourcing of data and arbitrary computations to an untrusted commodity cloud. Zhang et al. [9] also presented the hybrid cloud techniques to support privacy-aware dataintensive computing resulting in protection of sensitive data from public cloud. Bellare et al. [1] showed Data confidentiality by transforming the predictable message into unpredictable message. Introduced key server as third party to generate the file tag for duplicate check. Puzio et al. [11] implements an additional encryption operation and an access control mechanism as metadata manager to handle key management. Stanek et al. [6] presented a novel encryption scheme that provides differential security for popular data and unpopular data. For popular data: the traditional conventional encryption is performed. For unpopular data: Another two-layered encryption scheme with stronger security while supporting deduplication is proposed. Xu et al. [7] also addressed the problem and showed a secure convergent encryption for efficient encryption, without considering issues of the keymanagement and block-level deduplication. Halevi et al. [3] proposed the notion of "proofs of ownership" (PoW) for deduplication systems, such that a client can efficiently prove to the cloud storage server that he/she owns a file without uploading the file itself. Ng et al. [5] extended PoW for encrypted files, but they do not address how to minimize the key management overhead.

III.EXISTING SYSTEM

In existing method, we presented DARE, a low-overhead Deduplication-Aware Resemblance detection and Elimination scheme that effectively exploit existing duplicate-adjacency information for highly efficient resemblance detections in data deduplication based backup storage system. The main idea behind DARE is to employ schemes, call Dup-Adjacency based Resemblance Detection (DupAdj), by considering any two data chunk to be similar. If their respective adjacent chunks are duplicate in a deduplication system, and then further enhance the resemblance detection efficiency by an improved super-feature approach.

3.1 Disadvantage

- High over head in computing the feature
- Low security

3.2 Proposed method

In this paper, we describe our cryptographic schemes for the problem of searching on encrypted data and provide maximum security for the resulting crypto systems. Our techniques have lot of crucial advantages. They are secure: they provide provable secrecy for encryptions, in the sense that the untrusted servers cannot learn about the text when only given the cipher-text; they provide query isolation for search, meaning that the untrusted servers cannot learn more about the plaintext than the search results; they provide controlled searching, so that the untrusted server cannot search for an arbitrary words without the user's authorization; they also support hidden query, so that the user may ask the un-trusted servers to search for a secret word without revealing the word to the servers .

3.3 Advantage

- High security
- Simple and fast

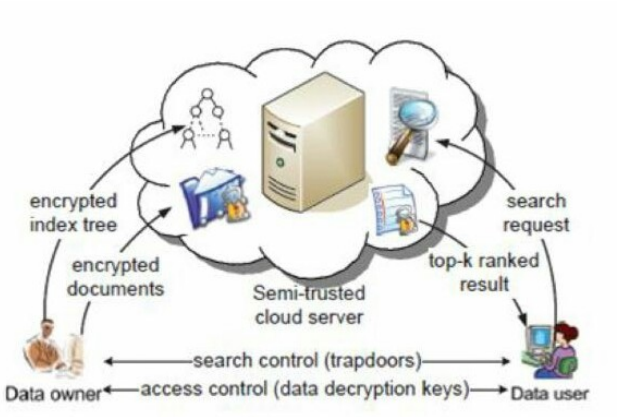

Fig.1 Architecture And Key Datastructure

IV.SUPER-FEATURE DETECTION

For similar chunks that differ only in a tiny fraction of bytes, most of their features will be identical due to the random distribution of the chunk's maximal-feature positions. Thus two data chunk can be considered very similar if any one of their super-features matches. Two chunks will have the same content of hashing region with a very high probability if they have the same Rabin fingerprint. Next, the probability of two similar chunks having the same feature is highly dependent upon their similarity degree the less similar two data chunks are to each other, the smaller the probability there will be of them having the same feature. Thus, the probability of two data chunks S1 and S2 being detected as resembling.

4.1 DupAdj Detection

This approach is to consider chunk pairs closely adjacent to any confirmed duplicate-chunk pair between two data streams as resembling pairs and thus candidates for delta compression. Which allows an efficient search of the duplicate-adjacent chunks for resemblance detection by traversing to prior or next chunks DupAdj Detection module of DARE processes an input segment, it will traverse all the chunks by the aforementioned doubly-linked list to find the already duplicate-detected chunks. Similar chunk pairs between segments A and B, until a dissimilar chunk or an already detected duplicate or similar chunk is found. Note that the detected chunks here are considered dissimilar (i.e., NOT similar) to others if their similarity degree (i.e., dela compressed size chunk size) is smaller than a predefined threshold, for resemblance detection.

4.2 Deduplication

The input data stream, first chunked, duplicate-detected, and then grouped into segments of sequential chunks. DARE will first detect duplicate chunks by the Deduplication module. Deduplication approaches be implemented here and the preservation of the backup-stream logical locality in the segments is required for further resemblance detection. A segment consists of the metadata of a number of sequential chunks, such as the chunk fingerprints, size, etc., which serves as the atomic unit in preserving the backup-stream logical locality for data reduction.

4.3 Searchable Encryption Algorithm

Searchable encryption is to provide privacy-preserving keyword searches of encrypted data. The first searchable encryption scheme was the Public-key Encryption with Keyword Search (PEKS) scheme based on Identity-Based Encryption (IBE) Searchable encryption scheme can be built using either a non-keyword based approach or an index/keyword based approach. In the non-keyword based approachs, the scheme scans the entire document words to find out the word *W* of interest. This provide the functionality to search any words in the documents. However, it takes a long search time for a large number of documents. On the other hand, index/keyword based solution builds up indexes, for each word *W* of interest and list out the corresponding documents that contain *W*. This provides a faster search operations when the document set is large.However, storing and updating an index can be an overhead.

4.4 Algorithm Details

Searchable encryption scheme is a cryptographic technique that allows search of specific informations in an encrypted content. The search operations are initiated at the user device and performed in the cloud servers. The operation consists of two methods: search Token () and search (). Search Token (): This takes the keywords and document ID as input and computes the search token at the client devices. The keyword encryption key is used to compute the encrypted value of the user given keyword and returns the generated search token. The search token is then sent to the servers. Search (): This takes place at the server when it receives a search request from the users. This method takes the search token and the user database as input to out the document IDs containing the search results. Finally, server retrieves the corresponding document and sends back to the client for decryption. A faster encryption process can be obtained if the keyword extraction module can work without the document conversion processes.

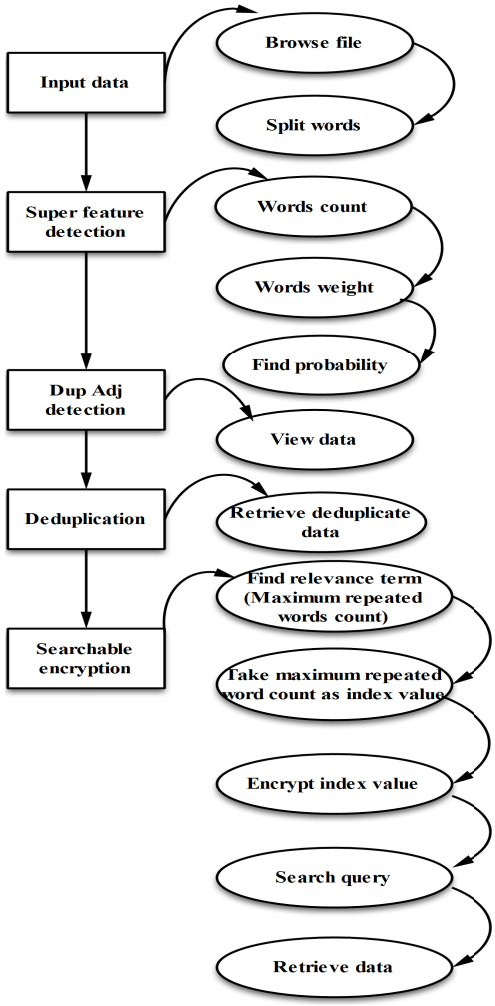

Fig2.Data Flow Diagram

4.5 Searching on Encrypted Data

We first define the problem of searching on encrypted data. Assume Alice has a set of documents and

stores them on an untrusted server Bob. For example, Alice could be a mobile user who stores her email messages on an untrusted mail server .Because Bob is untrusted, Alice wishes to en-crypt her documents and only store the ciphertext on Bob. Each document can be divided up into 'words'. Each 'word' may be any token; it may be a 64-bit block, an English word, a sentence, or some other atomic quantity, according to the application domain of interest. For simplicity, we typically assume these 'words' have the same length. Because Alice may have only a low-bandwidth network connection to the server Bob, she wishes to only retrieve the documents which contain the word W. In or-der to achieve this goal, we need to design a scheme so that after performing certain computations over the ciphertext, Bob can determine with some probability whether each document contains the word W without learning anything else.

There seem to be two types of approaches. One possibility is to build up an index that, for each word W of interest, lists the documents that contain W. An alternative is to per-form a sequential scan without the indexe. The advantage of using an index is that it may be faster than the sequential scan when the document is large. The disadvantage of using the indexes is that storing and updating the index can be of substantial overhead. So the approach of using an index is more suitable for mostly-read-only data. We first describe our schemes for searching on encrypted data without an index.

V.CONCLUSION

We have described new techniques for remote searching on encrypted data using an untrusted server and provided proofs of security for the resulting crypto systems. Our techniques have a number of crucial advantages: they are provably secure; they support controlled and hidden search and query isolation; they are simple and fast (More specifically , for a document of length n, the encryption and search algorithms only need O(n) stream cipher and block cipher operations); and they introduce almost no space and communication overhead. Our scheme is also very flexible, and it can easily be extended to support more advanced search queries. We conclude that this provides a powerful new building block for the construction of secure services in the untrusted infrastructure.

REFERENCES

[1] M. Bellare, S. Keelveedhi, and T. Ristenpart. "Dupless: Serveraided encryption for deduplicated storage". In USENIX Security Symposium, 2013.

[2] S. Bugiel, S. Nurnberger, A. Sadeghi, and T. Schneider. „Twin clouds: An architecture for secure cloud computing". In Workshop on Cryptography and Security in Clouds (WCSC 2011), 2011.

[3] S. Halevi, D. Harnik, B. Pinkas, and A. Shulman-Peleg. "Proofs of ownership in remote storage systems". In Y. Chen, G. Danezis, and V. Shmatikov, editors, ACM Conference on Computer and Communications Security, pages 491–500. ACM, 2011

[4] J. Li, X. Chen, M. Li, J. Li, P. Lee, andW. Lou. "Secure deduplication with efficient and reliable convergent key management". In IEEE Transactions on Parallel and Distributed Systems, 2013.

[5] W. K. Ng, Y. Wen, and H. Zhu. "Private data deduplication protocols in cloud storage". In S. Ossowski and P. Lecca, editors, Proceedings of the 27th Annual ACM Symposium on Applied Computing, pages 441–446. ACM, 2012.

[6] J. Stanek, A. Sorniotti, E. Androulaki, and L. Kencl. "A secure data deduplication scheme for cloud storage". In Technical Report, 2013.

[7] J. Xu, E.-C. Chang, and J. Zhou. "Weak leakage-resilient clientside deduplication of encrypted data in cloud storage". In ASIACCS, pages 195–206, 2013.

[8] J. Yuan and S. Yu. "Secure and constant cost public cloud storage auditing with deduplication". IACR Cryptology ePrint Archive, 2013:149, 2013.

[9] K Zhang, X Zhou, Y Chen and X Wang, "Sedic Privacy-Aware Data Intensive Computing".

[10] Kaaniche, N. ; Inst. Mines-Telecom, Telecom Sud Paris, Evry, France; Laurent,M.A "Secure Client Side Deduplication Scheme in Cloud Storage Environments".

[11] Puzio, P. ; SecludIT, Sophia-Antipolis, France ; Molva, R.; Onen,M.; Loureiro,S. "ClouDedup Secure Deduplication with Encrypted Data for Cloud Storage".