An enhanced SECURE EVALUATION AND PREVENTION OF DUPLICATE DATA IN CLOUD for different applications

Kamal narren , v vinay krishna

[1]Research scholar, [2,3]Professor

[1,2,3] CSE Department

[1,3]JJT University, Jhunjhunu, Rajasthan, [2]Institute of Aeronautical Engineering, Hyderabad.

ABSTRACT

Cloud computing provides a way of storing a voluminous data and can be easily accessed anywhere. This work deals about the prevention of a duplicate file storage in cloud. Here there are three important components in our system namely the owner of the data who generated and will store it in the cloud, the user can be a valid person who will download the file after providing the suitable credentials namely the user will provide the encrypted key he obtained through his mail to download a file from the cloud, the third important component is the cloud. Here the file is provided a unique value which can be used to identify it SHA is used for this purpose. Then for the secure encryption and decryption RSA algorithm is used. The user needs to provide this key to decrypt and obtain file ensure confidentiality of data. A owner can upload a file and when he uploads a duplicate copy of the file the cloud server will notify a error. This duplicate prevention in a cloud ensures that the memory space is effectively utilized thereby reducing the processing overhead.

Key words- cloud computing, Secure Hash Algorithm (SHA), encryption, decryption, RSA, confidentiality.

1.0 INTRODUCTION

For data management scalable in cloud computing, deduplication is a well-known technique. Data deduplication is a specialized data compression technique for removing duplicate copies of data in storage. The technique is used to improve storage utilization and we can also apply to network data transfers to reduce the number of bytes that must be sent. Instead of keeping multiple copies of data with the same content, deduplication eliminates redundant data by keeping only one physical copy and referring other redundant data to that copy. Deduplication can take either the file level or the block level. Although data deduplication brings a lot of

benefits, security and privacy concerns arise as users' sensitive data are susceptible to both insider and outsider attacks. Traditional encryption, while providing data confidentiality, is incompatible with data deduplication. Specifically, traditional encryption requires different users to encrypt their data with their own keys.

Thus, identical data copies of different users will lead to different cipher texts, making deduplication impossible. Convergent encryption has been proposed to enforce data con-fidentiality while making deduplication feasible. It en-crypts/decrypts a data copy with a convergent key, which is obtained by computing the cryptographic hash value of the content of the data copy. After key generation and data encryption, users retain the keys and send the ciphertext to the cloud. Since the encryption operation is deterministic and is derived from the data content, identical data copies will generate the same convergent key and hence the same ciphertext. To prevent unauthorized access, a secure proof of ownership protocol is also needed to provide the proof that the user indeed owns the same file when a duplicate is found. After the proof, subsequent users with the same file will be provided a pointer from the server without needing to upload the same file.

A user can download the encrypted file with the pointer from the server, which can only be decrypted by the corresponding data owners with their convergent keys. Thus, convergent encryption allows the cloud to perform deduplication on the cipher texts and the proof of ownership prevents the unauthorized user to access the file. However, previous deduplication systems cannot sup-port differential authorization duplicate check, which is important in many applications.

2.0 LITERATURE REVIEW

M. Bellare, S. Keelveedhi, and T. Ristenpart (2013) Securing information and Data deduplication is one of important techniques for eliminating duplicate copies of existing secured information, and is widely used in cloud storage to minimize the amount of storage space and save bandwidth. To protect the confidentiality of available data while supporting deduplication, the convergent encryption technique is proposed to encrypt the data before outsourcing. To implement information security, this paper does make the first attempt to formally address the problem of authorized data deduplication. Different from traditional deduplication systems, the differential privileges of users are further considered in duplicate check besides the data itself. Here, in this paper, We also present different new deduplication constructions supporting

authorized duplicate check in a Secured De-duplication architecture. Security analysis demonstrates that our system is secure in terms of the definitions specified in the proposed security model. We implement a prototype of our proposed authorized duplicate check scheme and conduct testbed experiments using our prototype, As a proof of concept. We show that our proposed authorized duplicate check scheme incurs minimal overhead compared to normal operations.

S. Bugiel, S. Nurnberger, A. Sadeghi, and T. Schneider (2011) Rendering efficient storage and security for all data is very important for cloud computing. Securing and privacy preserving of data is of high priority when it comes to cloud storage. Therefore to provide efficient storage for cloud data owners and render high security for data this paper proposes Cloud Computing Secure Framework (CCSF). Intrusion detection and prevention are performed manually by network operators in the existing system. In our proposed architecture the intrusion detection and prevention is performed automatically by defining rules for the major attacks and alert the system automatically. The major attacks/events includes vulnerabilities, cross site scripting (XSS), SQL injection, cookie poisoning, wrapping. Data deduplication technique allows the cloud users to manage their cloud storage space effectively by avoiding storage of repeated data's and save bandwidth. The data are finally stored in cloud server namely Cloud Me. To ensure data confidentiality the data are stored in an encrypted type using Advanced Encryption Standard (AES) algorithm.

S. Halevi, D. Harnik, B. Pinkas, and A. Shulman-Peleg (2011) During the last decade, cloud computing technology becomes an attractive trend of leveraging cloud based services for large scale content storage, processing and distribution. Thus, data deduplication becomes more and more a necessity for cloud service provider. Also security and privacy are top concern for the public cloud. Aiming to address the above storage and security challenges, this paper makes the attempt to formalize the notion of secured and efficient cloud storage system. We developed a prototype consist of client side deduplication for the security, storing and sharing outsourced data using the hybrid cloud. As a proof of our framework, we implement prototype and demonstrate that the incurred overhead is very limited in realistic environment.

J. Stanek, A. Sorniotti, E. Androulaki, and L. Kencl (2013) In the recent trend, every data and contents are stored in the cloud using cloud storage services. With the huge amount of data from every client may affect the cloud storage. In specific, the redundant content may perform

more worst in the storage part. The de-duplication method is generally used to reduce the storage cost and resource requirements of data services in the cloud by eliminating redundant data and storing only a single copy of them. Deduplication is most effective when multiple users outsource the same data to the cloud storage services, but it creates several issues relating to search and security. Data mining is an effective way to solve such problems in the cloud service. This paper surveys various techniques and methods used to detect duplicate records in the cloud storage service.

3.0 METHODOLOGY

Functions of data deduplication

It compares objects (usually files or blocks) and removes objects (copies) that already exist in the data set. The deduplication process removes blocks that are not unique.

1. Divide the input data into blocks or "chunks."
2. Calculate a hash value for each block of data.
3. Use these values to determine if another block of the same data has already been stored.
4. Replace the duplicate data with a reference to the object already in the database.

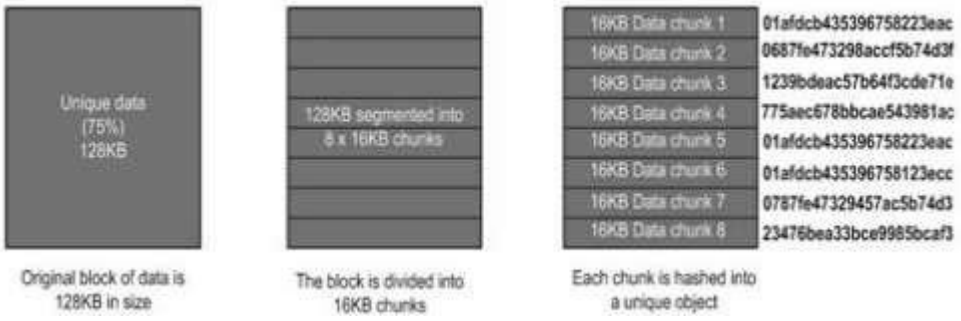

Figure working mode of data deduplication

Once the data is chunked, an index can be created from the results, and the duplicates can be found and eliminated. Only single instance of data is stored The actual process of data deduplication can be implemented in a number of different ways. We can eliminate duplicate data by simply comparing two files and making the decision to delete one that is older or no longer needed. The most common methods of implementing deduplication are:

- File-based compare
- File-based versioning
- File-based hashing
- Block or sub-block versioning

• Block or sub-block hashing

File-based compare

File system-based deduplication is a simple method to reduce duplicate data at the file level, and usually is just a compare operation within the file system or a file system based algorithm that eliminates duplicates. An example of this method is comparing the name, size, type and date-modified information of two files with the same name being stored in a system. If these parameters match, you can be pretty sure that the files are copies of each other and that you can delete one of them with no problems. Although this example isn't a foolproof method of proper data deduplication, it can be done with any operating system and can be scripted to automate the process, and best of all, it's free. Based on a typical enterprise environment running the usual applications, you could probably squeeze out between 10 percent to 20 percent better storage utilization by just getting rid of duplicate files.

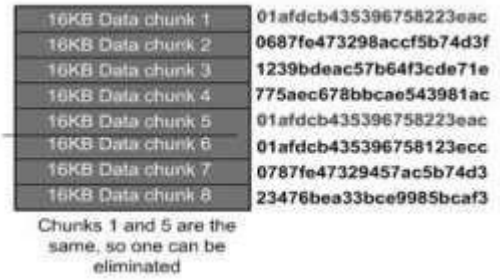

Figure Data elimination process

File-based delta versioning and hashing

More intelligent file-level deduplication methods actually look inside individual files and compare differences within the files themselves, or compare updates to a file and then just store the differences as a "delta" to the original file. File versioning associates updates to a file and just stores the deltas as other versions. File-based hashing actually creates a unique mathematical "hash" representation of files, and then compares hashes for new files to the original. If there is a hash match, you can guarantee the files are the same, and one can be removed.

4.0 RESULTS

We evaluated our prototype by conducting experiments in a LAN, where each machine equipped with Intel Core-2 Duo 2.93 GHz CPU, 4GB RAM, Windows 7 Professional 32-bit Operating System. The LAN machines are connected with 100Mbps Ethernet network.

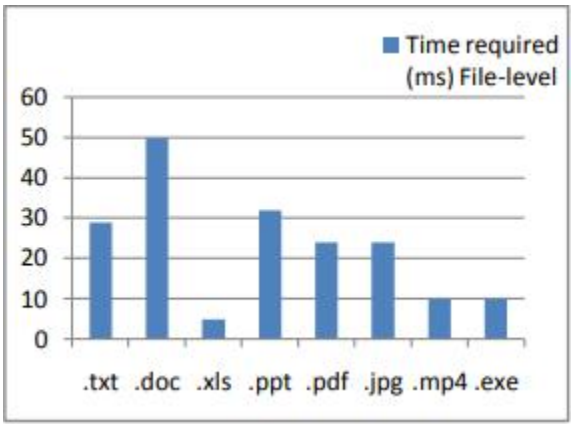

Figure Deduplication time factor at File levels

As our proposed system involves byte-level duplication check along with file and variable-size block-level too, we evaluated the deduplication system on basis of different factors:

1) File-type

2) Unique-file uploads time

3) Deduplication ratio, by breaking down our process into minimal steps as:

a)Duplicate check at 3 levels(file/block/byte)

b)Creating file pointer if duplicate found

c) Encryption of file, if not duplicate and

d)Finally transfer or upload to CSP.

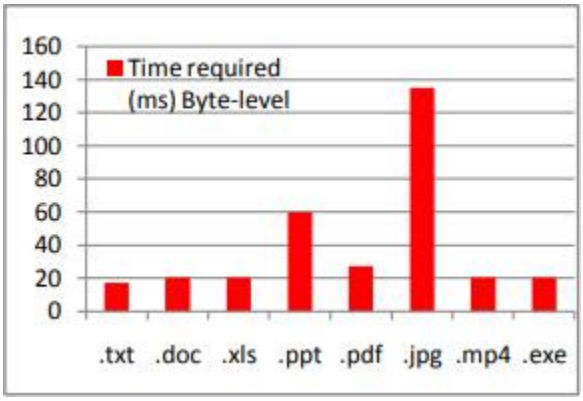

Figure Deduplication time factor at Byte levels

With increase in file size, the time spent on duplicate check, encryption and transfer increases. We evaluated the effect of time to upload unique files by uploading 150 1MB unique files of different types. For every type, time remains constant for file encryption and upload.

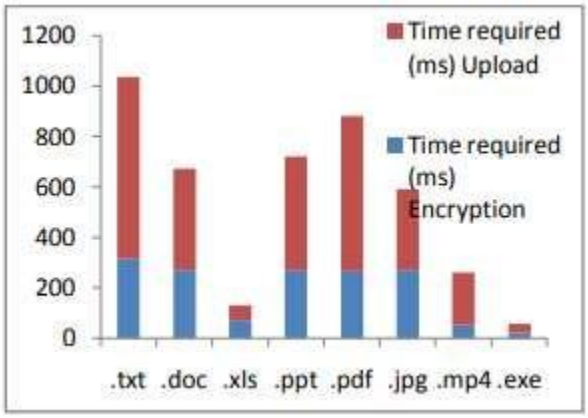

Figure Time required uploading unique file of different type

To evaluate the deduplication ratio, we uploaded same set of 150 1MB files again. Here, in case of duplicate files, encryption and uploading time would be skipped, thus achieving deduplication ratio 100%. The time required to assured duplication in either of the level is mentioned.

5.0 CONCLUSION

The notion of authorized data de-duplication technique is specialized data compression technique which eliminates redundant data as well as improves storage and bandwidth utilization. Convergent encryption technique is proposed to enforce confidentiality during de-duplication, which encrypt data before outsourcing. Security analysis demonstrates that the schemes are secure in terms of insider and outsider attacks. To better protect data security, we present Two Factor Authentication scheme (2FA) of user along with PoW of files, to address problem of authorized data de-duplication, in which the duplicate-check tokens of files are generated by the private cloud server with private keys.

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] Iuon-Chang and Po-Ching Chien, “Data Dedupliction Scheme for Cloud storage”. IJ3C, Vol. 1, No. 2 (2012)