© 2016 IJSRST | Volume 2 | Issue 2 | Print ISSN: 2395-6011 | Online ISSN: 2395-602X
Themed Section: Science and Technology

Detecting File Repeatation Using Convergent Encryption Scheme on Cloud

G Jeyasankar, E Deepanraj, R Ramakrishnan
Department of Information Technology, Dhanalakshmi College of Engineering, Chennai, Tamilnadu, India

ABSTRACT

Detecting repeatation of data in a cloud is technique of removing duplicate copies of data. It is widely used in cloud to save space and bandwidth. However, single copy of each file stored in cloud server even if many number of users owns that single copy. As a result, the de-duplication technique increases the storage usability while decreasing the reliability. By which the every user encrypt the file before uploading. The confidentiality on data is made by using convergent encryption algorithm, which also detects data repeatation. In convergent encryption scheme a user or data owner derives the key from the file content and encrypts the file with that key. In addition the user generate the tag for that file copy and put it into the data center, such that the tag is used for detecting repeatation of data on cloud.

Keywords: De-Duplication, Proof of Ownership, Convergent Encryption, Confidentiality.

I. INTRODUCTION

Cloud storage is a model of networked enterprise where data or files are stored in virtualized pools .It motivates enterprises and organizations to storage of the outsource data to cloud service provider. Detecting data repeatation is a technique where a single copy of data or file is stored on cloud and it provides data link to the user. The user needs to check whether the file is existing or not on cloud when they are uploading the file into the cloud. Here the concept of detecting repeatation of data is used to whether the data or file is already exists or not. Now day's continuously increasing number of user and the size of their data, de-duplication technique to greater extent for cloud storage.

There are two levels of de-duplications are take place. One is file level de-duplication and another one is block level de-duplication. The file level de-duplication refers to the entire data or file content and the block level de-duplication method refers to the fixed or variable size of data.

To make secure de-duplication we need to use a few security mechanisms. In normal encryption, different files and different keys are producing different cipher text and same file with different keys producing different Cipher text.

This will create problems for storage server, they will have to save these different cipher texts for the same file it will create a memory problem. The convergent encryption scheme algorithm [1] [5] providing better solution for that problem. Using any hash function such as SHA, the hash value for the files is generated. Since, both files are same they will produce same cipher text. Convergent key encryption uses this hash value as key to encrypt the files. The simple idea behind convergent encryption is same file producing same cipher text since, keys and files are same.

II. METHODS AND MATERIAL

Related Work

The client side de-duplication enables to send data to directly to the cloud storage. In client side de-duplication the user can check whether the data is duplicate or not. If the data is not duplicate then the user can be directly send the data to the cloud storage. If the

data or file is already present on the cloud storage the proof of ownership protocol is performed. The proof of ownership protocol [2] [6] [8] provide the solution to keep the security on client side de-duplication check. Here the client and server can act as prover (i.e., user) and verifier (i.e., storage server). In the proof of ownership protocol a client can prove to the server that the client has exactly target files.

By using proof of ownership protocol [2] [6] to keep the security on client side de-duplication [3]. The verifier (i.e.,storage server) derives the short value from a file and generates set of challenges and sends them to the client. The prover (i.e., client) responds with the proof of file ownership. It is passing to the server if and only that are same and the proof is correct.

In our de-duplication mechanism, we deploy both file and block level de-duplication. Uploading a file to the storage server the user initially, examine the file level de-duplication. If the file is already, present then the all its blocks must be duplicated. Otherwise the user further checks the block level de-duplication before uploading to the cloud storage. The tag generation algorithm which helps to the client to generate set of tags and put it into the file content before uploading a file into the storage server.

III. RESULT AND DISCUSSION

1. Implementation Details

A. Existing System

In the previous work the computational load at cloud server and cloud user is too huge for tag generation. The security considered in the previous work prevention of leakage of side channel information.in order to prevent the leakage of side channel information we use the tradition of proof of ownership protocol between client and cloud server.

The using of traditional algorithms will create the problem for storage server, because of in traditional algorithm the same file producing the different cipher text.

Disadvantages

- Data loss and lots of duplicate files.
- Huge computational load at client side.
- Data confidentiality is not achieved.

B. Proposed System

In our proposed system the data confidentiality is achieved in client side data or file de-duplication by performing the proof ownership protocol .which allows the user to directly examine the data de-duplication. By using convergent encryption algorithm the user can encrypt the file content before uploading. The challenges of de-duplication on encrypted data or file is prevention of dictionary attack.

Advantages

- Data confidentiality can be achieved.
- Duplicate files are mapped with single copy of existing file in the cloud.
- Data integrity including tag consistency, can be achieved

C. Convergent Encryption Algorithm

Convergent encryption also known as content hash keying is a cryptosystem that produces identical cipher text from identical plain texts. Which used for removing duplicate files from storage server without the provide having access to the encryption keys

Convergent encryption [1] [5] provides data or file confidentiality in de-duplication mechanism. In convergent encryption scheme, if both files are same they will produce the same cipher text. The convergent encryption scheme algorithm uses the hash values as a key to encrypt the files. The user or data owner derives the convergent key [1] [5] [7] from the file content and encrypts the file content with the convergent key. In addition, the user extract the tag (i.e., hash key) for the file content, such that the tag will be used for the detect the repeatation (i.e., duplicate) of data file.

- KeyGen(M) : The key generation algorithm takes file content M as input and outputs the convergent key ckM of file M.
- Encrypt(ckM,M) : The encryption algorithm takes the convergent key ckM and file content M as input and outputs the cipher text ctM.

- Decrypt(ckM,ctM) : The decryption algorithm takes the convergent key ckM and cipher text ctM as input and outputs the plain text M.
- TagGen(M) : The tag generation algorithm takes file
  Content M as input and produce the output as tagM of M. We allow TagGen to generate tag from the corresponding cipher text by using tagM = TagGen(ct), Where c = Encrypt(ckM,M).

Before uploading a file, the client can duplicate check with cloud server to verify if such a file content is already stored on cloud storage or not. If there is duplicate proof of ownership (POW) protocol [8] [9] [10] execute between cloud client and cloud server. Before uploading the file content to the cloud server, tag generation algorithm, which helps to the user to generate tags (i.e., TagGen(M) and put it into the file content and send them to the cloud server. Tag generation is used to detect the duplicate files. If no duplicate is found the user will processed further steps.

If user need to download a file M from cloud storage. The user downloads the encrypted file (i.e., ckM, M) from cloud storage. It needs to decrypt the file by recovering the convergent key (i.e, ckM) and using these key the user can decrypt file to obtain the original plain file (i.e., M).

2. Security Analysis

The file has been encoded (i.e., cipher text) by using encryption key before stored on the cloud storage. Thus, make the data confidentiality in cloud. In our construction the key server generate the convergent key and hash value for the duplicate check. If any adversary cannot generate valid convergent key without the private key.

Even if the file is predictable by the adversary cannot guess the file with brute- force attack. If the adversary cannot allowed to collude with the key server.

By using the security mechanism of proof of ownership protocol [2][8][9] scheme, any adversary without the file cannot convince to the cloud storage.

D. System Model

Cloud Clients

The cloud clients or user outsource the data to the cloud server and can access the data from cloud storage server. The user or client can be upload a unique file or data but does not upload the duplicate file.

Cloud Storage Server

The cloud storage provide the facility of outsource the data service to the user. Where, data's are stored in the virtualize resource of storage pools. Only the single copy of these files is stored on cloud storage. All data copies and tags of these files are stored in the cloud storage. These tags used for duplicate check.

3. Architecture

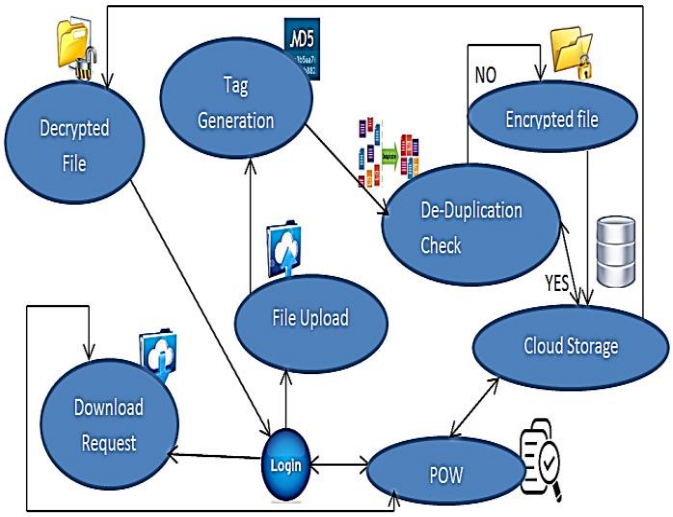

Figure 1: System Architecture

IV. CONCLUSION

We proposed the de-duplication system to increase the Data confidentiality and improving the data reliability while outsource the data to cloud storage. In addition it enables secure de-duplication through the proof of ownership protocol to prevent the leakage of side channel information. In our advanced construction motivated by the fact that the user always need to encrypt their data before uploading on the cloud.

V. REFERENCES

[1] J. Li, X. Chen, M. Li, J. Li, P. Lee, andW. Lou. Secure De-duplication with efficient and reliable convergent key management. In IEEE Transactions on Parallel and Distributed Systems, 2014.

[2] S. Halevi, D. Harnik, B. Pinkas, and A. Shulman-Peleg,‘‘Proofs of Ownership in Remote Storage Systems,’’ in Proc. ACM Conf. Comput. Commun. Security, Y. Chen, G. Danezis, and V. Shmatikov, Eds., 2011, pp. 491-500.

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

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

[5] J. Li, X. Chen, M. Li, J. Li, P. Lee, and W. Lou, "Secure deduplication with efficient and reliable convergent key management," in IEEE Transactions on Parallel and Distributed Systems, 2014, pp. vol. 25(6), pp. 1615–1625.

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

[7] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in Advances in Cryptology — CRYPTO 2001, ser. Lecture Notes in Computer Science, J. Kilian, Ed. Springer Berlin Heidelberg, 2001, vol. 2139, pp. 213–229.

[8] R. D. Pietro and A. Sorniotti, "Boosting efficiency and security in proof of ownership for deduplication." in ACM Symposium on Information, Computer and Communications Security, H. Y. Youm and Y. Won, Eds. ACM, 2012, pp. 81–82.

[9] G. Ateniese, R. Burns, R. Curtmola, J. Herring, O. Khan, L. Kissner, Z. Peterson, and D. Song, "Remote data checking using provable data possession," ACM Trans. Inf. Syst. Secur., vol. 14, no. 1, pp. 12:1–12:34, 2011.

[10] G. Ateniese, R. Di Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in Proceedings of the 4th International Conference on Security and Privacy in Communication Networks, ser. SecureComm ’08. New York, NY, USA: ACM, 2008, pp. 9:1–9:10.

[11] C. Erway, A. K¨upc¸ ¨u, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," in Proceedings of the 16th ACM Conference on Computer and Communications Security, ser. CCS ’09. New York, NY, USA: ACM, 2009, pp. 213–222.