Irish Interdisciplinary Journal of Science & Research (IIJSR)
Vol.4, Iss.3, Pages 91-95, July-September 2020

Condense Data Replication on Encrypted File Storing for Improving Cloud Server Efficiency

D.Poorani[1] & K.Ilango[2]

[1]PG Scholar, Department of Computer Science and Engineering, M.A.M. College of Engineering, Tamilnadu, India.
[2]Head of the Department, Computer Science and Engineering, M.A.M. College of Engineering, Trichy, Tamilnadu, India.

Article Received: 19 June 2020 Article Accepted: 27 August 2020 Article Published: 11 September 2020

ABSTRACT

Cloud storage such as Dropbox and Bitcasa is one of the most popular cloud services. Currently, with the prevalence of cloud computing, users can even collaboratively edit the newest version of documents and synchronize the newest files. A remarkable feature of current cloud storage is its virtually infinite storage. To support unlimited storage, the cloud storage provider uses data deduplication techniques to reduce the data to be stored and therefore reduce the storage expense. Moreover, the use of data deduplication also helps significantly reduce the need for bandwidth and therefore improve the user experience. Nevertheless, in spite of the above benefits, data deduplication has its inherent security weaknesses. Among them, the most severe is that the adversary may have an unauthorized file downloading via the file hash only. Convergent encryption has been proposed to safeguard the information secrecy while making de-duplication done. It encrypts/decrypts an information duplicate with a convergent key, which is acquired by computing the cryptographic hash estimation of the substance of the information duplicate. When key generation and encryption of data is executed, client retains the key and sends the encoded data to the cloud. As the encryption process is defined and cannot be differentiated from the actual information content, indistinguishable information duplicates. And also to avoid unauthorized access, a protected proof of ownership is expected to provide the assurance that the client to be clear in requesting a same record if the copy is found. In this article we first review the previous solutions and identify their performance weaknesses. Then we propose an alternative design that achieves cloud server efficiency.

Keywords: *Cloud computing, Cloud server, Data encryption, Efficiency.*

1. Introduction

Security of outsourced data is the important concern for Cloud Client. For that, they encrypt their file before outsourcing. Traditional encryption mechanisms conflicts with Deduplication mechanism. Using these encryption mechanisms, Cloud Clients encrypts the file with a randomly chosen encryption key [1-2]. It results indistinguishable ciphertext files for identical plaintext files. In this way, Deduplication cannot verify existence of an identical file. Convergent

Encryption is a mechanism which allows Deduplication. We consider convergent encryption for our approach [3]. Deduplication process uses file hash value as Deduplication identity to file existence verification. However, it raises significant security issues like. (i) A malicious Cloud Client learns hash value by listening the communication between file owner and Cloud Server. He shows this hash value and gets access to the outsourced file of other Cloud Client. (ii) An adversary is able to use Cloud Storage as Content Distribution network by publishing hash value (e.g. on his web page) as Deduplication identity. Any Cloud Client can get access to file by presenting hash value. This behavior conflicts business model of Cloud Storage. It supports piracy and malicious content distribution. Cloud storage services are actually designed for multiple upload operations and few share operations [4-5].

We propose secure and efficient Proof of Ownership (PoW) to mitigate discussed threats. PoW is a mechanism to verify file ownership of requester. In PoW, Cloud Server is a challenger and Cloud Client is its responder. We deploy an idea of computing matrix based challenges. Cloud Server considers the file as a matrix of bits and selects random square sub matrix. It challenges Cloud Client to send inverse of chosen sub-matrix. If Cloud Client owns the file, he is able to compute sub-matrix inverse. Randomness in PoW Challenges secures the Deduplication system from discussed security threats. Moreover, PoW

communication between Cloud Client and Cloud Server does not reveal any information about plain text or ciphertext of the file. To the best of our knowledge, our approach requires less computation to generate PoW Response at client side and verify PoW Response at server side.

2. Problem statement

We consider following three entities in our system model as shown in Figure 1.

(1) Cloud Clients (CCs)

A large number of Cloud Clients are connected with Cloud Server. Locally, CCs have limited computation and storage resources. They outsource their data on cloud server. They expect security and low charges on outsourced data. They allow Deduplication to minimize communication cost and storage cost. CCs participate in proof of ownership process to protect the data from malicious access.

(2) Cloud Server (CS)

Cloud Server performs upload, download, delete and update operations which are requested by CC. CS performs Deduplication verification before uploading data. CS assists PoW process to protect cloud storage from Content Distribution Network attack.

(3) Cloud Storage

Cloud Storage is a physical storage device which can perform write, read, erase and re-write operations on stored data.

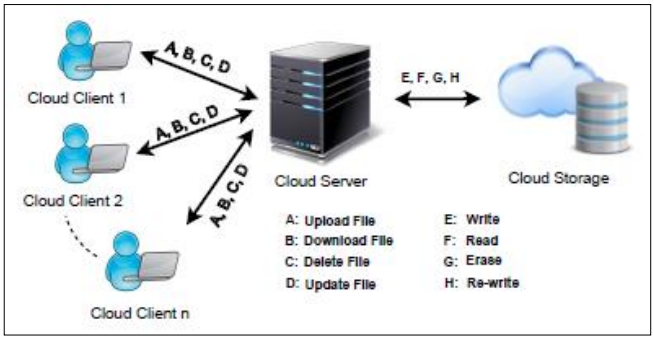

Fig.1 System Model

3. Algorithm Methodology

3.1 Convergent Encryption

Convergent encryption is a mechanism to provide data security while realizing Deduplication. Convergent encryption consists following elements:

Key←Hash (F): *Hash() is a* ***cryptographic hash*** *function. It takes file F as input and outputs convergent key.*

CF ← Encrypt(Key, F): *Encrypt() is a standard encryption function. It takes file F and convergent key as input and outputs ciphertext file.*

F←Decrypt(Key, 𝐶): *Decrypt() is the decryption function. It takes ciphertext file CF and convergent key as input and outputs file plaintext F.*

Proof of Ownership (PoW): Proof of Ownership is the mechanism to verify file ownership of requester. Two entities participate in PoW mechanism namely, Challenger and Responder.

Cloud Server is a challenger. Cloud Server sends challenges to the responder Cloud Client who requests to upload data. PoW can be executed via (i) Hash of data as proof.

3.2 Merkle Hash Tree as proof

In (i), responder shows hash of data as proof to get access to deduplicated data. However, this method is weak PoW. Adversary can gain knowledge about hash of data by listening communication between Cloud Client and Cloud Server and uses it as proof to get access to outsourced data by other Cloud Clients. In (ii), Cloud Client and Cloud Server computes Merkle Hash Tree. Merkle Hash Tree is computed using fixed sized chunks of file.

In this paper, system framework improved with security. Various benefits keys impelled efficient encryption for the document. Repetition of data cannot be identified without comparison of client data. Unauthorized client access does not allow message duplicate checking and cannot decode the message even conspire the CSP. Security examination proves that the proposed framework is safe as it is indicated in the proposed system. Convergent encryption has been proposed to safeguard the information secrecy while making de-duplication done. It encrypts/decrypts an information duplicate with a convergent key, which is acquired by computing the cryptographic hash estimation of the substance of the information duplicate. When key generation and encryption of data is executed, client retains the key and sends the encoded data to the cloud. As the encryption process is defined and cannot be differentiated from the actual information content, indistinguishable information duplicates will produce the same focalized key and thus similar figure content. To avoid unauthorized access, a protected evidence of proprietorship convention is expected to provide the assurance that the client to be clear in requesting a same record if the copy is found.

3.3 Algorithm Upload File

procedure Cloud Client: Upload File(File)

Key←Hash(File)

CF ←Encrypt(Key, File)

Compute Hash(CF)

Presence←VerifyPresence(Hash(CF))

if Presence = FALSE then

Upload(Hash(CF), CF)

else

Response(Challenge)

Procedure Cloud Server: Verify presence(Hash(CF))

if Hash(CF) ∈ FilesList then

Return "TRUE"

else

Return "FALSE"

3.4 Algorithm Download File

procedure CLOUD CLIENT: DOWNLOAD REQUEST(

Hash(C))

Sends Hash(C) to SSP

procedure CLOUD SERVER: DOWNLOAD RESPONSE(

Hash(C))

if Cloud Client /∈ {FileOwners} then

Return "NULL"

else

(Hash(CF))

procedure CLOUD CLIENT: DECRYPT FILE((CF , Hash(CF)))

if Hash(CF)≠ Received Hash(CF) then

Return

else

File = Decrypt(Key, CF)

4. Conclusion

In this paper, Convergent encryption has been proposed to safeguard the information secrecy while making de-duplication done. It encrypts/decrypts an information duplicate with a convergent key, which is acquired by computing the cryptographic hash estimation of the substance of the information duplicate. When key generation and encryption of data is executed, client retains the key and sends the encoded data to the cloud. As the encryption process is defined and cannot be differentiated from the actual information content, indistinguishable information duplicates. And also to avoid unauthorized access, a protected evidence of proprietorship convention (PoW) is expected to provide the assurance that the client to be clear in requesting a same record if the copy is found. In this article we first review the previous solutions and

identify their performance weaknesses. Then we propose an alternative design that achieves cloud server efficiency.

References

[1] Wang, C., Wang, Q., Ren, K., & Lou, W. (2010). Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing. 2010 Proceedings IEEE INFOCOM. doi:10.1109/infcom.2010.5462173

[2] Wang, Q., Wang, C., Ren, K., Lou, W., & Li, J. (2011). Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing. IEEE Transactions on Parallel and Distributed Systems, 22(5), 847–859. doi:10.1109/tpds.2010.183

[3] Shah, Mehul & Swaminathan, Ram & Baker, Mary. (2008). Privacy-Preserving Audit and Extraction of Digital Contents.. IACR Cryptology ePrint Archive. 2008. 186.

[4] Bellare, M., & Neven, G. (2006). Multi-signatures in the plain public-Key model and a general forking lemma. Proceedings of the 13th ACM Conference on Computer and Communications Security - CCS '06. doi:10.1145/1180405.1180453

[5] Merkle, R. C. (1980). Protocols for Public Key Cryptosystems. 1980 IEEE Symposium on Security and Privacy. doi:10.1109/sp.1980.10006