An Efficient & Enhance Merkle Tree for Big Data Deduplication in Cloud

A. S. Suji[1] J. Arun[2] V. Yathavaraj[3]
[1]PG Student [2,3]Assistant Professor
[1,2,3]Department of Computer Science & Engineering
[1,2,3]Maharaja Engineering College, Avinashi, India

Abstract— Cloud computing, as an emerging computing paradigm, enables users to remotely store their data in a cloud, so as to enjoy services on-demand. With rapid development of cloud computing, more and more enterprises will outsource their sensitive data for sharing in a cloud. To keep the shared data confidential against untrusted cloud service providers (CSPs), a natural way is to store only the encrypted data in a cloud. In previous approaches designed efficient and privacy-preserving big data deduplication in cloud storage achieves both privacy-preserving and data availability, in addition takes accountability into consideration to offer better privacy assurances than existing schemes. Proposed approach is to reduce the communication, storage overheads and duplicates search in cloud storage services previous researchers designed data deduplication schemes to protect from resist brute-force attacks or ensure the efficiency and data availability, but not both conditions. In this work propose block index search approach for privacy-preserving big data deduplication to fully protect the duplicate information from disclosure, even by a malicious CSP, without affecting the capability to perform data deduplication and Merkle Tree over encrypted data, in order to derive a unique identifier of outsourced data, this identifier serves to check the availability of the same data in remote cloud servers and it is used to ensure efficient access control in cloud. Extensive security and performance analysis show that the proposed scheme is highly efficient and provably secure.

Key words: Big Data, Deduplication, Cloud Computing

I. Introduction

Cloud computing provides seemingly unlimited “virtualized” resources to users as services across the whole Internet, while hiding platform and implementation details. Today’s cloud service providers offer both highly available storage and massively parallel computing resources at relatively low costs Data deduplication is a specialized data compression technique for eliminating duplicate copies of repeating data in storage. The technique is used to improve storage utilization and can also be applied to network data transfers to reduce the number of bytes that must be sent. Instead of keeping multiple data copies 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 place at either the file level or the block level. For file-level deduplication, it eliminates duplicate copies of the same file. Deduplication can also take place at the block level, which eliminates duplicate blocks of data that occur in non-identical files. 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. Data deduplication is one of important data compression techniques for eliminating duplicate copies of repeating data, and has been widely used in cloud storage to reduce the amount of storage space and save bandwidth. To protect the confidentiality of sensitive data while supporting deduplication. R-MLE2 (Dynamic) scheme seeks to improve the efficiency of duplicate ciphertext identification. However, the scheme suffers from brute-force attacks. Another efficient secure deduplication scheme SecDep to resist brute-force attacks. However, this scheme only deals with small-sized data, and is not suitable for big data deduplication. Another work a scheme to deduplicate encrypted big data stored in the cloud based on ownership challenge and proxy re-encryption. Although this scheme is efficient, it is vulnerable to brute-force attacks. In this work propose block index search approach for privacy-preserving big data deduplication to fully protect the duplicate information from disclosure, even by a malicious CSP, without affecting the capability to perform data deduplication and Merkle Tree over encrypted data, in order to derive a unique identifier of outsourced data, this identifier serves to check the availability of the same data in remote cloud servers and it is used to ensure efficient access control in cloud. It is constructed as a binary tree where the leaves in the are the hashes of authentic data values. The verifier with the authentic hr requests and requires the authentication of the received blocks. The provider provides the verifier with the auxiliary authentication information. The verifier can then verify and then checking if the calculated hr is the same as the authentic one. It is commonly used to authenticate the values of data blocks. An augmented inverted index structure called Block Index search early termination. A Block index search augments the commonly used inverted index structure, where for each distinct term t we store a sorted list of the IDs of those keywords where t occurs, with upper-bound values for blocks of these IDs. Extensive security and performance analysis show that the proposed scheme is highly efficient and provably secure.

II. Methodology

A. Creation of Cloud Environment

The initially the basic network model for the cloud data storage is developed in this module. Three different network entities can be identified as follows :Client: an entity, which has large data files to be stored in the cloud and relies on the cloud for data maintenance and computation, can be either individual consumers or organizations; Cloud Storage Server (CSS): an entity, which is managed by Cloud Service Provider (CSP), has significant storage space and computation resource to maintain the clients’ data; Data owner : an entity, which has expertise and capabilities that clients do not have, is trusted to assess and expose risk of cloud storage services on behalf of the clients upon request.

In the cloud paradigm, by putting the large data files on the remote servers, the clients can be relieved of the burden of storage and computation. KDC: The trusted KDC is tasked with the distribution and management of private keys for the system.

B. Key Distribution Center

Let G and G1 be two cyclic groups of prime order p and E (G*G1) = G2 be a map. G is a bilinear group if all the operations involved above are efficiently computable. Many classes of elliptic curves feature bilinear groups. KDC takes a security parameter k as input, and outputs a 5-tuple (N, g, G, G_t, e) by running the composite bilinear parameter generator algorithm Gen (k). Then, it selects four random numbers s, t, a, b €Z_N, and computes Ya, Yb. Then KDC chooses three cryptographic hash functions H1: $\{0, 1\}^* \dashrightarrow \{0,1\}^n$, H2: $\{0, 1\}^* \dashrightarrow Z^*_p$, H3: G → $\{0, 1\}^*$, where n is the bit length of symmetric key. Finally, KDC sends s and t to all members in domains A and B, respectively, and sends yA and yB to the CSP by secure channel. KDC publishes parameters pp = (N,g,G,G_T ,e, $e(g,g)^s$ $e(g,g)^t$, h1,h2,h3).

C. Merkle Tree based Block index Search

A Merkle Tree is a well-studied authentication structure, which is intended to efficiently and securely prove that a set of elements are undamaged and unaltered. It is constructed as a binary tree where the leaves in the hashes of authentic data values. The verifier with the authentic hr requests and requires the authentication of the received blocks. The provider provides the verifier with the auxiliary authentication information. The verifier can then verify and then checking if the calculated hr is the same as the authentic one. It is commonly used to authenticate the values of data blocks. An augmented inverted index structure called Block Index search early termination. A Block index search augments the commonly used inverted index structure, where for each distinct term t we store a sorted list of the IDs of those keywords where t occurs, with upper-bound values for blocks of these IDs. That is, for every say IDs of keywords containing a term t, store the maximum term-wise score of any of these keywords with respect to t [14].

D. Data Uploading & Downloading

An entity, which has large data files to be stored in the cloud and relies on the cloud for data maintenance and computation, can be either individual consumers or organizations; Users rely on the CS for cloud data storage and maintenance. They may also dynamically interact with the CS to access and update their stored data for various application purposes. The data owner first encrypts the data's, and then stored the cloud server. Cloud server is considered as "honest-but-curious" in our model, which is consistent with the most related works on searchable encryption. Specifically, cloud server acts in an "honest" fashion and correctly follows the designated protocol specification. The cloud server collects the some different encrypted documents. In this process encrypted data to store with cloud server. Search result should be ranked by cloud. The cloud server according to some ranking criteria. The cloud server is responsible to search the index I and return the corresponding set of encrypted documents. Anyone user wants to request the data and then server able to send the user request. The retrieve the data from server to user. Searching index/rank calculation On the one hand, to meet the effective data retrieval need, large amount of documents demand cloud server to perform result relevance ranking, instead of returning undifferentiated result. Ranked search can also elegantly eliminate unnecessary network traffic by sending back only the most relevant data. The server response to users request with the rank and index estimation while search a document in cloud. The client request for our needs send to cloud server. Specifically, cloud server acts in an "honest" fashion and correctly follows the designated protocol specification. All encrypted key data to send to admin of the cloud server, the admin to take the encrypted data. Then all the data to store in cloud server. 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. The access control mechanism is employed to manage decryption capabilities given to users.

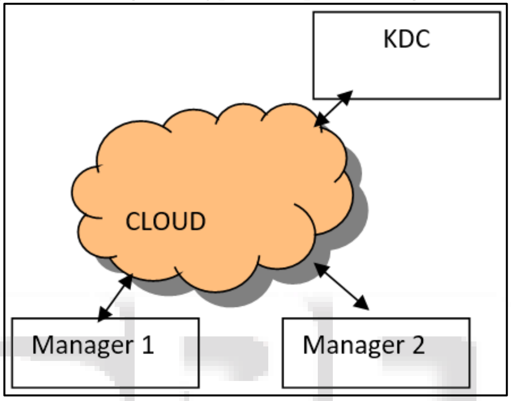

Fig. 1: System Model

III. Related works

In recent times, secure data deduplication has been studied by the research community [6][8] [9] [10]. Proxy Re-Encryption, is a cryptosystem that produces identical ciphertexts from identical plaintext files and has been widely applied in secure data deduplication [7], [6]. Bellare et al. [5] formalize a new cryptographic primitive, message locked encryption (MLE), to improve the security of CE. The use of a deterministic and message-dependent symmetric key [4], these approaches suffer from inherent security limitations shown in [2]. To enhance the security of deduplication and ensure data confidentiality, Keelveedhi et al. [4] explained how one can ensure data confidentiality by transforming the predictable message into an unpredictable message. In their system, a key server is introduced to generate the file tag for duplication check. However, the third party server suffers from the single point of failure. Thus, Liu et al. proposed the first secure cross-user deduplication scheme that supports client-side encryption without requiring any additional independent servers.

IV. Experimental setup & results

In order to evaluate the performance at the proposed model have used the following performance metrics namely computation time complexity and data integrity. At each time computed the average time for uploading and downloading the encrypted file, the results of average time computation to upload and download data file in the cloud database. From this analysis can know that the proposed system having the better performance than the existing system.

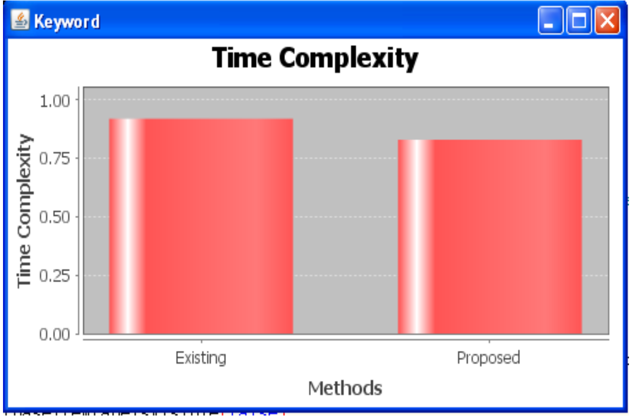

Fig 2: Time Complexity Performance Graph

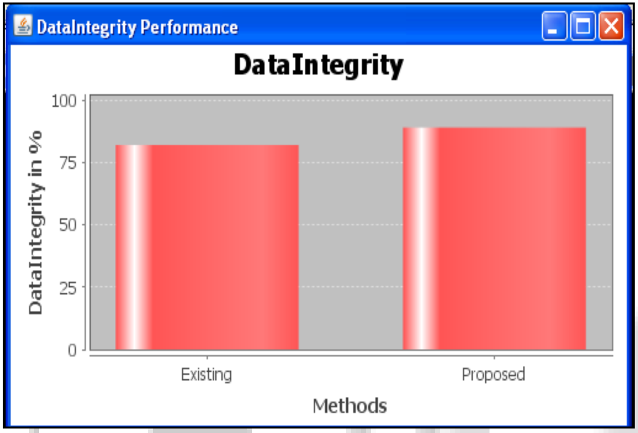

Fig. 3: Data Integrity Performance Graph

V. Conclusion

In this work propose block index search approach for privacy-preserving big data deduplication to fully protect the duplicate information from disclosure, even by a malicious CSP, without affecting the capability to perform data deduplication and Merkle Tree over encrypted data, in order to derive a unique identifier of outsourced data, this identifier serves to check the availability of the same data in remote cloud servers and it is used to ensure efficient access control in cloud. The performance at the proposed model has used the following performance metrics namely computation time complexity and data integrity. Based on the comparison and the results from the experiment show the proposed approach works better.

References

[1] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, “Improved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage”, Proc. Network and Distributed Systems Security Symp. (NDSS), pp. 29-43, 2006.

[2] Danny Harnik, Benny Pinkas, Alexandra Shulman-Peleg,“Side channels in cloud services, the case of de duplication in cloud storage”, IEEE Security and Privacy Magazine, special issue of Cloud Security, Vol. 8, No. 2, pp. 40-47, 2010.

[3] J. Paulo and J. Pereira, “A survey and classification of storage de duplication systems”, ACM Comput. Surv., vol. 47, no. 1, pp. 11:1– 11:30, 2014.

[4] S. Keelveedhi, M. Bellare, and T. Ristenpart, “Dupless: Server-aided encryption for de duplicated storage”, in Proceedings of the 22th USENIX Security Symposium, Washington, DC, USA, August 14-16, 2013, pp. 179–194.

[5] M. Bellare, S. Keelveedhi, and T. Ristenpart, “Message-locked encryption and secure de duplication”, in Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013, pp. 296–312.

[6] T. Jiang, X. Chen, Q. Wu, J. Ma, W. Susilo, and W. Lou, “Secure and efficient cloud data de duplication with randomized tag”, IEEE Trans. Information Forensics and Security, vol. PP, no. 99, pp. 1–1, 2016.

[7] Y. Zhou, D. Feng, W. Xia, M. Fu, F. Huang, Y. Zhang, and C. Li, “Secdep: A user-aware efficient fine-grained secure de-duplication scheme with multi-level key management”, in IEEE 31st Symposium on Mass Storage Systems and Technologies, MSST 2015, Santa Clara, CA, USA, May 30 - June 5, 2015, 2015, pp. 1–14.

[8] R. D. Pietro and A. Sorniotti, “Boosting efficiency and security in proof of ownership for deduplication,” in 7th ACM Symposium on Information, Compuer and Communications Security, ASIACCS ’12, Seoul, Korea, May 2-4, 2012, pp. 81–82.

[9] S. Halevi, D. Harnik, B. Pinkas, and A. Shulman-Peleg, “Proofs of ownership in remote storage systems”, IACR Cryptology ePrint Archive, vol. 2, pp. 207, 2011.

[10] N. Kaaniche and M. Laurent, “A secure client side deduplication scheme in cloud storage environments,” in 6th International Conference on New Technologies, Mobility and Security, NTMS 2014, Dubai, United Arab Emirates, March 30 - April 2, 2014.

[11] Singhal, “Modern Information Retrieval: A Brief Overview,” IEEE Data Eng. Bull., vol. 24, no. 4, pp. 35-43, 2001.

[12] E.-C. Chang and J. Xu, “Remote Integrity Check with DishonestStorage Server,” Proc. 13th European Symp.Research in ComputerSecurity (ESORICS ’08), pp. 223-237, 2008.

[13] Juels and B.S. Kaliski Jr., “Pors: Proofs of Retrievability for Large Files,” Proc. ACM Conf. Computer and Comm. Security, P. Ning, S.D.C. di Vimercati, and P.F. Syverson, eds., pp. 584-597, 2007.

[14] V.Eswaramoorthy, P.Sengottuvelan and N.Kuppuswamy, “Fuzzy Logic based Improved Lion Optimization Algorithm Centered Routing Protocol for Mobile Adhoc Network”, Asian Journal of Research in Social Sciences and Humanities, vol.7, No.7, pp. 934-949.