A Survey on Different approaches for Secure Deduplication of Encrypted data in cloud

[1]Pushpa Shavi, [2]Jayanthi M G
[1]M.Tech Student, Dept of CSE, [2]Associate Professor,Dept of CSE
Cambridge Institute of Technology, Bangalore, Karnataka

Abstract - As Big Data cloud storage servers are becoming popular the shortage of disk space in the cloud becomes a problem. Data deduplication is a method to control the explosion growth of data on the cloud and most of the storage providers are finding more secure and efficient methods for their sensitive data.The following paper reviews on different methods of secure deduplication of encrypted data in cloud. The main motive of this review is to suggest a suitable mechanism for eliminating duplicate copies of identical data in order to save storage space and network bandwidth in cloud. By the end of the paper we will be summarizing the advanced methods that are used for secure deduplication of encrypted data in cloud.

Keywords: ABE, Storage, Deduplication, Ciphertext, ClouDedup, SecDep.

I.INTRODUCTION

With the potentially infinite storage space offered by cloud providers, users tend to use as much space as they can and vendors constantly look for techniques aimed to minimize redundant data and maximize space savings.The simple idea behind deduplication is to store duplicate data (either files or blocks) only once. Therefore, if a user wants to upload a file (block) which is already stored, the cloud provider will add the user to the owner list of that file (block). Deduplication has proved to achieve high space and cost savings and many cloud storage providers are currently adopting it. Deduplication can reduce storage needs by up to 90-95% for backup applications [11] and up to 68% in standard filesystems.

Along with low ownership costs and flexibility, users require the protection of their data and confidentiality guarantees through encryption. Unfortunately, deduplication and encryption are two conflicting technologies. While the aim of deduplication is to detect identical data segments and store them only once, the result of encryption is to make two identical data segments indistinguishable after being encrypted.This means that if data are encrypted by users in a standard way, the cloud storage provider cannot apply deduplication since two identical data segments will be different after encryption. On the other hand, if data are not encrypted by users, confidentiality cannot be guaranteed and data are not protected against curious cloud storage providers.As a result a suitable mechanism for eliminating duplicate copies of identical data in order to save storage space and network bandwidth in cloud.in this paper different methods are mentioned for secure deduplication of encrypted data in cloud.

1.1 Our Contributions

In this paper, we present an attribute-based storage system which employs ciphertext-policy attribute-based encryption (CP-ABE) and supports secure deduplication. Our main contributions can be summarized as follows.

Discussion on related works

The system is the first that achieves the stan- dard notion of semantic security for data confiden- tiality in attribute-based deduplication systems by resorting to the hybrid cloud architecture [12].

A methodology to modify a ciphertext over one access policy into ciphertexts of the same plaintext but under any other access policies without revealing the underlying plaintext. This technique might be of independent interest in addition to the application in the proposed storage system.

An approach based on two cryp- tographic primitives, including a zero-knowledge proof of knowledge [13] and a commitment scheme ..

II.RELATED WORKS

Table 1 : The different approaches for secure deduplication of Encrypted data in cloud.

Secure Deduplication Approach	Characteristics and Method	References
Ciphertext-Policy Attribute-Based Encryption	In this technique, Encrypted data can be kept confidential even if the storage server is untrusted and secure against collusion attacks. Allows for a new type of encrypted access control where user's private keys are specified by a set of attributes and a party encrypting data can specify a policy	[1],[2]

	over these attributes specifying which users are able to decrypt. In this approach ,private keys will be identified with a set S of descriptive attributes. A party that wishes to encrypt a message will specify through an access tree structure a policy that private keys must satisfy in order to decrypt. Each interior node of the tree is a threshold gate and the leaves are associated with attributes. The encryption algorithm encrypts a message M under the tree access structure . Decryption procedure as a recursive algorithm.	
ClouDedup	Two main operations of ClouDedup: storage and retrieval. During the storage procedure, a user uploads a file to the system. Enables block-level deduplication at the same time. Showed that it is worth performing block-level deduplication instead of filelevel deduplication. Additional layers of encryption are added by the server and the optional HSM. Secret keys can be generated in a hardware dependent way by the device itself and do not need to be shared with anyone else.	[3],[4]
SecDep	A User-Aware Efficient Fine-Grained Secure Deduplication Scheme with Multi-Level Key Management. SecDep consists of Users,a Storage Provider (SP), and Distributed Key Servers (DKS). User. A user is an entity who wants to upload data to (download data from) the Storage Provider (SP). Storage Provider (SP). The storage providers mainly offer computation and storage services. Distributed Key Servers (DKS). The DKS is built on aquorum of key servers via Shamir Secret Sharing Scheme(SSSS) to ensure security of keys. When a user wants to access the DKS and the SP, his/her passwords and credentials should be verified at first. Filelevel keys are securely divided into share-level keys via Shamir Secret Sharing Scheme (SSSS). User Aware Convergent Encryption algorithm (UACE) and Multi Level Key management (MLK) approaches. (1) UACE uses server-aided HCE at file-level and user-aided CE at chunk level to resist brute-force attacks. (2) MLK encrypts chunk-level keys by file level key, which avoids key space increasing with the number of sharing users. Evaluation results suggest that MLK reduces 59.5-63.6% and 34.8-96.6% of key space overheads.	[5],[6]
Deduplication based on Verifiable Hash Convergent Group Signcryption	Introduces the secure VHCGS framework and then construct a scheme with two sub-protocols for the deduplication process. Upload protocol to store a new ciphertext at the storage server. Download protocol red by which the client can restore a ciphertext by verifying ownerships. The proxy(the trusted third party) can verify the ciphertext and the signature without recovering the original message. VHCGS is composed of three protocols: a setup protocol, an upload protocol, and a download protocol. VHCGS ensures both message security and tag consistency as well as the bandwidth efficiency of the group user and cloud storage server.	[7],[8]
Decentralized Server-aided Encryption for Secure Deduplication	The key idea is to construct an inter-KS deduplication algorithm, by which the CSP can check whether two ciphertexts encrypted under the different secret keys of different KSs (within a tenant or across tenants) have the identical plaintext or not. By utilizing a blind signature scheme, the inter-KS deduplication algorithm realizes cross-tenant data deduplication without revealing any sensitive information except the ciphertext itself. System model consists of user,key server and Cloud storage service provider	[9],[10]

	(CSP). User (U): This is a client who owns data (or files 1), and wishes to outsource the data into the cloud storage for the purpose of backup or file-sharing. Key server (KS): This is an entity responsible for generating a convergent key and a corresponding tag for each request from U. Cloud storage service provider (CSP): This is an entity that offers cloud storage services. Uploading of F will be completed through the following sub-operations: (1) key and tag generation, (2)encryption of F , and (3) intra-KS deduplication. U interacts with the CSP to download an outsourced file F . Offers flexibility in managing a KS to tenants, while at the same time allowing a CSP to perform cross-tenant deduplication over encrypted data	

III.PROPOSED WORK.

An attribute-based storage system with secure deduplication in a hybrid cloud setting, where a private cloud is responsible for duplicate detection and a public cloud manages the storage. It has two advantages. Firstly, it can be used to confidentially share data with users by specifying access policies rather than sharing decryption keys. Secondly, it achieves the standard notion of semantic security for data confidentiality while existing systems only achieve it by defining a weaker security notion.

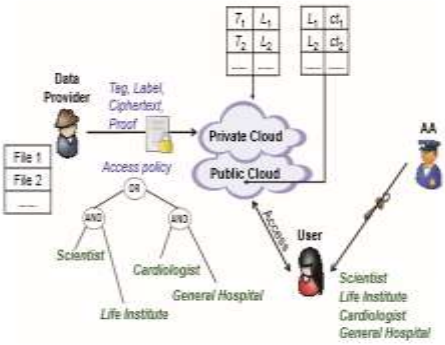

Fig. 1: System architecture of attribute-based storage with secure deduplication.

The architecture of our attribute-based storage system with secure deduplication is shown in Fig. 2 in which four entities are involved: data providers, attribute authority (AA), cloud and users. A data provider wants to outsource his/her data to the cloud and share it with users possessing certain credentials. The AA issues every user a decryption key associated with his/her set of attributes. The cloud consists of a public cloud which is in charge of data storage and a private cloud which performs certain computation such as tag checking. When sending a file storage request, each data provider firstly creates a tag T and a label L associated with the data, and then encrypts the data under an access structure over a set of attributes. Also, each data provider generates a proof pf on the relationship of the tag T , the label L and the encrypted message ct3 , but this proof will not be stored anywhere in the cloud and is only used during the checking phase for any newly generated storage request. After receiving a storage request, the private cloud first checks the validity of the proof pf, and then tests the equality of the new tag T with existing tags in the system. If there is no match for this new tag T , the private cloud adds the tag T and the label L to a tag-label list, and forwards the label and the encrypted data, (L, ct) to the public cloud for storage. Otherwise, let ct0 be the ciphertext whose tag matches the new tag and L0 be the label associated with ct0 , and then the private cloud executes as follows.

- If the access policy in ct is a subset of that in ct^0, the private cloud simply discards the new storage request; else, if the access policy in ct^0 is a subset of that in ct, the private cloud asks the public cloud to replace the stored pair (L^0, ct^0) with the new pair (L, ct) where $L = L^0$.
- If the access policies in ct and ct^0 are not mutually contained, the private cloud runs the ciphertext re-generation algorithm to yield a new ciphertext for the same underlying plaintext file and associated with an access structure which is the union of the two access.

IV. CONCLUSION

In this paper, we have designed a scheme that supports secure deduplication of encrypted data in cloud.Attribute-based encryption (ABE) has been widely used in cloud computing where data providers outsource their encrypted data to the cloud and can share the data with users possessing specified credentials. On the other hand, deduplication is an important technique

to save the storage space and network bandwidth, which eliminates duplicate copies of identical data. However, the standard ABE systems do not support secure deduplication, which makes them costly to be applied in some commercial storage services. In this paper, we presented a novel approach to realize an attribute-based storage system supporting secure deduplication. Our storage system is built under a hybrid cloud architecture, where a private cloud manipulates the computation and a public cloud manages the storage. The private cloud is provided with a trapdoor key associated with the ciphertext over one access policy into ciphertexts of the same plaintext under any other access policies without being aware of the underlying plaintext. After receiving a storage request, the private cloud first checks the validity of the uploaded item through the attached proof. If the proof is valid, the private cloud runs a tag matching algorithm to see whether the same data underlying the ciphertext has been stored. If so, whenever it is necessary, it regenerates the ciphertext into a ciphertext of the same plaintext over an access policy which is the union set of both access policies. The proposed storage system enjoys two major advantages. Firstly, it can be used to confidentially share data with other users by specifying an access policy rather than sharing the decryption key. Secondly, it achieves the standard notion of semantic security while existing deduplication schemes only achieve it under a weaker security notion.

REFERENCES

[1] A. Beimel. Secure Schemes for Secret Sharing and Key Distribution. PhD thesis, Israel Institute of Technology, Technion, Haifa, Israel, 1996.

[2] M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In ACM conference on Computer and Communications Security (ACM CCS), pages 62–73, 1993.

[3] Chuanyi Liu, Xiaojian Liu, and Lei Wan. Policy-based deduplication in secure cloud storage. In Trustworthy Computing and Services,pages 250–262. Springer, 2013.

[4] Luis Marques and Carlos J Costa. Secure deduplication on mobile devices. In Proceedings of the 2011 Workshop on Open Source and Design of Communication, pages 19–26. ACM, 2011.

[5] M. Bellare, S. Keelveedhi, and T. Ristenpart, "Dupless: server-aided encryption for deduplicated storage," in Proceedings of the 22nd USENIX Security Symposium. Washington, DC, USA USENIX Association, August 2013, pp. 1–16.

[6] J. R. Douceur, A. Adya, W. J. Bolosky, and et al, "Reclaiming space from duplicate files in a serverless distributed filesystem," in Proceedings of the 22nd International Conference on Distributed Computing Systems. Vienna, Austria: IEEE Computer Society Press, July 2002, pp. 617–624.]

[7] S. Halevi, D. Harnik, B. Pinkas, A. Shulman-Peleg, "Proofs of ownership in remote storage systems", ACM Conference on Computer and Communications Security 2011, pp. 491-500, 2011.

[8] P. Gokulraj, K. Kiruthika-Devi, "Revocation and security based ownership deduplication of convergent key creating in cloud", International Journal of Innovative Research in Science, Engineering and Technology,vol.3, issue 10, pp. 16527-16533, 2014

[9]Durao, J. F. S. Carvalho, A. Fonseka, and V. C. Garcia, "A systematic review on cloud computing," The Journal of Supercomputing,vol. 68, no. 3, pp. 1321–1346, 2014.

[10] P. Mell and T. Grance, "The NIST definition cloud computing," National Institute of Standards and Technology of, Information Technology Laboratory, Tech. Rep., 2009.

[11] K. R. Choo, J. Domingo-Ferrer, and L. Zhang, "Cloud cryptog- raphy: Theory, practice and future research directions," Future Generation Comp. Syst., vol. 62, pp. 51–53, 2016.

[12] K. R. Choo, M. Herman, M. Iorga, and B. Martini, "Cloud foren- sics: State-of-the-art and future directions," Digital Investigation, vol. 18, pp. 77–78, 2016.

[13] B. Zhu, K. Li, and R. H. Patterson, "Avoiding the disk bottleneck in the data domain deduplication file system," in 6th USENIX Conference on File and Storage Technologies, FAST 2008, February 26-29, 2008, San Jose, CA, USA. USENIX, 2008, pp. 269–282.