



A Secure Hybrid Cloud Approach to Avoid Deduplication

Waghamare Amol Arjun

Department of computer engineering (KJCOEMR)
Kj college of engineering management and research pune, Pune University, India
waghamareamol92@gmail.com

Prof. Mr. M. K. Mokashi

Department of computer engineering (KJCOEMR)
Kj college of engineering management and research pune, Pune University, India
findmeasmandar@gmail.com

Abstract— 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, the convergent encryption technique has been proposed to encrypt the data before outsourcing. . To better protect data security, this paper makes 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. We also present several new deduplication constructions supporting authorized duplicate check in hybrid cloud architecture. Security analysis demonstrates that our scheme is secure in terms of the definitions specified in the proposed security model. As a proof of concept, we implement a prototype of our proposed authorized duplicate check scheme and conduct testbed experiments using our prototype. We show that our proposed authorized duplicate check scheme incurs minimal overhead compared to normal operations.

Keywords- authorized duplicate check, confidentiality, hybrid cloud

I. INTRODUCTION

Cloud computing provides seemingly unlimited “virtualized” resources to users as services across the whole Internet, while hiding platform and implementation de-tails. Today’s cloud service providers offer both highly available storage and massively parallel computing re-sources at relatively low costs. As cloud computing becomes prevalent, an increasing amount of data is being stored in the cloud and shared by users with specified *privileges*, which define the access rights of the stored data. One critical challenge of cloud storage services is the management of the ever-increasing volume of data.

To make data management scalable in cloud computing, deduplication [17] has been a well-known technique and has attracted more and more attention recently. 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. 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 ciphertexts, making deduplication impossible. Convergent encryption [8] has been proposed to enforce data confidentiality while making deduplication feasible. It en-encrypts/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 [11] 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 ciphertexts and the proof of ownership prevents the unauthorized user to access the file.

II. EXISTING SYSTEM

Data deduplication system, the private cloud is involved as a proxy to allow data owner/user to security perform duplicate check with differential privileges.

Such architecture is practical and has attracted much attention from researchers.

The data owners only outsource their data storage by utilizing public cloud while the data operation is managed in private cloud

DISADVANTAGES OF EXISTING SYSTEM:

Traditional encryption, while providing data confidentiality, is incompatible with data deduplication.

Identical data copies of different users will lead to different ciphertexts, making deduplication impossible

III. PROPOSED SYSTEM

In this paper, we enhance our system in security. Specifically, we present an advanced scheme to support stronger security by encrypting the file with differential privilege keys. In this way, the users without corresponding privileges cannot perform the duplicate check. Furthermore, such unauthorized users cannot decrypt the cipher text even collude with the S-CSP. Security analysis demonstrates that our system is secure in terms of the definitions specified in the proposed security model.

ADVANTAGES OF PROPOSED SYSTEM:

The user is only allowed to perform the duplicate check for files marked with the corresponding privileges.

We present an advanced scheme to support stronger security by encrypting the file with differential privilege keys.

Reduce the storage size of the tags for integrity check. To enhance the security of deduplication and protect the data confidentiality

IV. IMPLEMENTATION

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

The implementation stage involves careful planning, investigation of the existing system and its constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

This project contains two main module:

1) User:

This perform following functions:

1. select a file :

In this we first select a file and generate file Tag by using SHA-1 algorithm .In this module we ask the user to make file private or shared and then make request to the private server. i.e admin to generate token. But before sending request to the admin first we check file is duplicate or not.

2. View file:In this we see all the file that selected by user.

3. Upload file to Cloud: In this module we upload the file that we are selected.
4. Download file: In this it is possible to download a file that is uploaded by ourselves and shared by other users.

2) Admin:

In this module, an admin acts as a private server which handles requests coming from users. This performs two functions: TokenGen(Tag, UserID) - It loads the associated privilege keys of the user and generates the token with HMAC-SHA-1 algorithm.

ShareTokenGen(Tag, {Priv.}) - It generates the share token with the corresponding privilege keys of the sharing privilege set with HMAC-SHA-1 algorithm.

Our implementation of the Storage Server provides deduplication and data storage with the following handlers and maintains a map between existing files and associated tokens with a Hash Map.

DupCheck(Token) - It searches the File to Token Map for duplicates; and

FileStore(FileID, File, Token) - It stores the File on Disk and updates the Mapping.

V. CONCLUSION

In this paper, the notion of authorized data deduplication was proposed to protect data security by including differential privileges of users in the duplicate check. We also presented several new deduplication constructions supporting authorized duplicate checks in hybrid cloud architecture, in which the duplicate-check tokens of files are generated by the private cloud server with private keys. Security analysis demonstrates that our schemes are secure in terms of insider and outsider attacks specified in the proposed security model. As a proof of concept, we implemented a prototype of our proposed authorized duplicate check scheme and conducted testbed experiments on our prototype. We showed that our authorized duplicate check scheme incurs minimal overhead compared to convergent encryption and network transfer.

REFERENCES

- [1] OpenSSL Project. <http://www.openssl.org/>.
- [2] P. Anderson and L. Zhang. Fast and secure laptop backups with encrypted de-duplication. In *Proc. of USENIX LISA*, 2010.
- [3] M. Bellare, S. Keelveedhi, and T. Ristenpart. Dupless: Server-aided encryption for deduplicated storage. In *USENIX Security Symposium*, 2013.
- [4] M. Bellare, S. Keelveedhi, and T. Ristenpart. Message-locked encryption and secure deduplication. In *EUROCRYPT*, pages 296–312, 2013.
- [5] M. Bellare, C. Namprempe, and G. Neven. Security proofs for identity-based identification and signature schemes. *J. Cryptology*, 22(1):1–61, 2009.
- [6] M. Bellare and A. Palacio. Gq and schnorr identification schemes: Proofs of security against impersonation under active and concurrent attacks. In *CRYPTO*, pages 162–177, 2002.
- [7] 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.
- [8] J. R. Douceur, A. Adya, W. J. Bolosky, D. Simon, and M. Theimer. Reclaiming space from duplicate files in a serverless distributed file system. In *ICDCS*, pages 617–624, 2002.
- [9] D. Ferraiolo and R. Kuhn. Role-based access controls. In *15th NIST-NCSC National Computer Security Conf.*, 1992.
- [10] GNU Libmicrohttpd. <http://www.gnu.org/software/libmicrohttpd/>.
- [11] 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.
- [12] 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*, 2013.
- [13] libcurl. <http://curl.haxx.se/libcurl/>.
- [14] C. Ng and P. Lee. Revdedup: A reverse deduplication storage system optimized for reads to latest backups. In *Proc. of APSYS*, Apr 2013.
- [15] 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.
- [16] R. D. Pietro and A. Sorniotti. Boosting efficiency and security in proof of ownership for deduplication. In H. Y. Youm and Y. Won, editors, *ACM Symposium on Information, Computer and Communications Security*, pages 81–82. ACM, 2012.
- [17] S. Quinlan and S. Dorward. Venti: a new approach to archival storage. In *Proc. USENIX FAST*, Jan 2002.
- [18] A. Rahumed, H. C. H. Chen, Y. Tang, P. P. C. Lee, and J. C. S. Lui. A secure cloud backup system with assured deletion and version control. In *3rd International Workshop on Security in Cloud Computing*, 2011.
- [19] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman. Role-based access control models. *IEEE Computer*, 29:38–47, Feb 1996.
- [20] J. Stanek, A. Sorniotti, E. Androulaki, and L. Kencl. A secure data deduplication scheme for cloud storage. In *Technical Report*, 2013.
- [21] M. W. Storer, K. Greenan, D. D. E. Long, and E. L. Miller. Secure data deduplication. In *Proc. of StorageSS*, 2008.
- [22] Z. Wilcox-O’Hearn and B. Warner. Tahoe: the least-authority filesystem. In *Proc. of ACM StorageSS*, 2008.
- [23] J. Xu, E.-C. Chang, and J. Zhou. Weak leakage-resilient client-side deduplication of encrypted data in cloud storage. In *ASIACCS*, pages 195–206, 2013.
- [24] J. Yuan and S. Yu. Secure and constant cost public cloud storage auditing with deduplication. *IACR Cryptology ePrint Archive*, 2013:149, 2013.
- [25] K. Zhang, X. Zhou, Y. Chen, X. Wang, and Y. Ruan. Sedic: privacy-aware data intensive computing on hybrid clouds. In *Proceedings of the 18th ACM conference on Computer and communications security*, CCS’11, pages 515–526, New York, NY, USA, 2011. ACM.

ACKNOWLEDGEMENT

This is a great pleasure & immense satisfaction to express my deepest sense of gratitude & thanks to everyone who has directly or indirectly helped me in my project work successfully. I express my gratitude towards project guide **Prof. M. K. Mokashi**, and **Prof. Mr. Mhetre** Head of Department of computer engineering KJ college of engineering management and research pune who guided & encouraged me in my project work in scheduled time. I would like to thank our **Principal Dr. Wagh S. J.**, for allowing us to pursue my project in this institute.

Thank you,
Amol Waghmare.