



# **A Hybrid Cloud Approach for Secure Authorized Deduplication**

G.Prashanthi, Z.Shobarani

M.Tech (CSE), Department of CSE, AITS, Rajampet, Kadapa(dist), Andhra Pradesh, India

Asst.Professor, Department of CSE, JNTUA University, Ananthapur. AITS, Rajampet, Kadapa(dist), Andhra Pradesh, India

**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. 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 a hybrid cloud architecture.

To better protect data security, this paper makes the first attempt to formally address the problem of authorized data deduplication. 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:** Deduplication, Hybrid cloud, convergent keys, Authorization, Duplicate Check, Confidentiality .

## **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. 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 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.

Cloud computing is an emerging service model that provides computation and storage resources on the Internet. One attractive functionality that cloud computing can offer is cloud storage. Individuals and enterprises are often required to remotely archive their data to avoid any information loss in case there are any hardware/software failures or unforeseen disasters. Instead of purchasing the needed storage media to keep data backups, individuals and enterprises can simply outsource their data backup services to the cloud service providers, which provide the necessary storage resources to host the data backups. While cloud storage is attractive, how to provide security guarantees for outsourced data becomes a rising concern. One major security challenge is to provide the property of assured deletion, i.e., data files are permanently inaccessible upon requests of deletion. Keeping data backups permanently is undesirable, as



**International Journal of Innovative Research in Computer and Communication Engineering**

**An ISO 3297: 2007 Certified Organization**

**Vol.3, Special Issue 4, April 2015**

**National Conference On Emerging Trends in Information, Digital & Embedded Systems (NC'e-TIDES -15)**

**Organized by**

**Dept. of ECE, Annamacharya Institute Of Technology & Sciences, Rajampet, Andhra Pradesh-516126, India held on 28<sup>th</sup> February 2015**

sensitive information may be exposed in the future because of data breach or erroneous management of cloud operators. Thus, to avoid liabilities, enterprises and government agencies usually keep their backups for a finite number of years and request to delete (or destroy) the backups afterwards. For example, the US Congress is formulating the Internet Data Retention legislation in asking ISPs to retain data for two years, while in United Kingdom, companies are required to retain wages and salary records for six years.

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 [1] has been proposed to enforce data confidentiality while making deduplication feasible. It 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 [2] 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. RELATED WORK

However, previous deduplication systems cannot support differential authorization duplicate check, which is important in many applications. In such an authorized deduplication system, each user is issued a set of privileges during system initialization. Each file uploaded to the cloud is also bounded by a set of privileges to specify which kind of users is allowed to perform the duplicate check and access the files. Before submitting his duplicate check request for some file, the user needs to take this file and his own privileges as inputs. The user is able to find a duplicate for this file if and only if there is a copy of this file and a matched privilege stored in cloud. For example, in a company, many different privileges will be assigned to employees. In order to save cost and efficiently management, the data will be moved to the storage server provider (S-CSP) in the public cloud with specified privileges and the deduplication technique will be applied to store only one copy of the same file. Because of privacy consideration, some files will be encrypted and allowed the duplicate check by employees with specified privileges to realize the access control. Traditional deduplication systems based on convergent encryption, although providing confidentiality to some extent, do not support the duplicate check with differential privileges. In other words, no differential privileges have been considered in the deduplication based on convergent encryption technique. It seems to be contradicted if we want to realize both deduplication and differential authorization duplicate check at the same time

### A. Symmetric Encryption

Symmetric encryption uses a common secret key  $\kappa$  to encrypt and decrypt information. A symmetric encryption scheme consists of three primitive functions:

$\text{KeyGenSE}(1\lambda) = \kappa$  is the key generation algorithm that generates  $\kappa$  using security parameter  $1\lambda$ .

$\text{EncSE}(\kappa, M) = C$  is the symmetric encryption algorithm that takes the secret  $\kappa$  and message  $M$  and then outputs the ciphertext  $C$ .

$\text{DecSE}(\kappa, C) = M$  is the symmetric decryption algorithm that takes the secret  $\kappa$  and ciphertext  $C$  and then outputs the original message  $M$ .



### **B. Convergent Encryption**

Convergent encryption [1], [3] provides data confidentiality in deduplication. A user (or data owner) derives a convergent key from each original data copy and encrypts the data copy with the convergent key. In addition, the user also derives a tag for the data copy, such that the tag will be used to detect duplicates. Here, we assume that the tag correctness property [3] holds, i.e., if two data copies are the same, then their tags are the same. To detect duplicates, the user first sends the tag to the server side to check if the identical copy has been already stored. Note that both the convergent key and the tag are independently derived, and the tag cannot be used to deduce the convergent key and compromise data confidentiality. Both the encrypted data copy and its corresponding tag will be stored on the server side

### **C. Proof of Ownership**

The notion of proof of ownership (PoW) [2] enables users to prove their ownership of data copies to the storage server. Specifically, PoW is implemented as an interactive algorithm (denoted by PoW) run by a prover (i.e., user) and a verifier (i.e., storage server). The verifier derives a short value ( $M$ ) from a data copy  $M$ . To prove the ownership of the data copy  $M$ , the prover needs to send  $\mathbb{Z}'$  to the verifier such that  $\mathbb{Z}' = (M)$ . The formal security definition for PoW roughly follows the threat model in a content distribution network, where an attacker does not know the entire file, but has accomplices who have the file. The accomplices follow the “bounded retrieval model”, such that they can help the attacker obtain the file, subject to the constraint that they must send fewer bits than the initial min-entropy of the file to the attacker [2].

### **D. Identification Protocol**

An identification protocol  $\Pi$  can be described with two phases: Proof and Verify. In the stage of Proof, a prover/user  $U$  can demonstrate his identity to a verifier by performing some identification proof related to his identity. The input of the prover/user is his private key  $sk_U$  that is sensitive information such as private key of a public key in his certificate or credit card number etc. that he would not like to share with the other users. The verifier performs the verification with input of public information  $pk_U$  related to  $sk_U$ . At the conclusion of the protocol, the verifier outputs either accept or reject to denote whether the proof is passed or not. There are many efficient identification protocols in literature, including certificate-based, identity-based identification etc. [4], [5].

## **III. PROPOSED ALGORITHM**

A convergent encryption scheme can be defined with four primitive functions:

- $\text{KeyGen}_{\text{CE}}(M) \rightarrow K$  is the key generation algorithm that maps a data copy  $M$  to a convergent key  $K$ ;
- $\text{Enc}_{\text{CE}}(K, M) \rightarrow C$  is the symmetric encryption algorithm that takes both the convergent key  $K$  and the data copy  $M$  as inputs and then outputs a ciphertext  $C$ ;
- $\text{Dec}_{\text{CE}}(K, C) \rightarrow M$  is the decryption algorithm that takes both the ciphertext  $C$  and the convergent key  $K$  as inputs and then outputs the original data copy  $M$ ; and
- $\text{TagGen}(M) \rightarrow T(M)$  is the tag generation algorithm that maps the original data copy  $M$  and outputs a tag  $T(M)$ .

The notion of proof of ownership (PoW) [11] enables users to prove their ownership of data copies to the storage server. Specifically, PoW is implemented as an interactive algorithm (denoted by PoW)

The verifier derives a short value  $\phi(M)$  from a data copy  $M$ . To prove the ownership of the data copy  $M$ , the prover needs to send  $\phi$  to the verifier such that  $\phi = \phi(M)$ .



## International Journal of Innovative Research in Computer and Communication Engineering

An ISO 3297: 2007 Certified Organization

Vol.3, Special Issue 4, April 2015

National Conference On Emerging Trends in Information, Digital & Embedded Systems (NC'e-TIDES -15)

Organized by

Dept. of ECE, Annamacharya Institute Of Technology & Sciences, Rajampet, Andhra Pradesh-516126, India held on 28<sup>th</sup> February 2015

### IV. PSEUDO CODE

Step1: Calculate the two convergent key values

Step2: Compare the two keys and files gets accessed.

Step3: Apply deduplication to eradicate the duplicate values.

Step4: If any other than the duplicates it will be checked once again and make the data unique.

Step5: That data will be unique and also more confidential the authorized can access and data is stored.

### V. SIMULATION RESULTS

The simulation can be takes place with the help of 3 process the process one is checking the person whether he is authorized or not the data is trusted or not. And the process 2 is confidential make to keep in public or not and the process 3 is removing the duplicate and making the data unique. And again it is going to check the data having any duplicate values and that will be keep stored in the cloud and that data is encrypted with the help of convergent keys values. And also the data size will be compressed and stored in the form of blocks.

### IV. CONCLUSION AND FUTURE WORK

Several new deduplication constructions supporting authorized duplicate check 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 conduct 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*,
- [5] M. Bellare, C. Namprempre, 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*, 2002.
- [7] S. Bugiel, S. Nurnberger, A. Sadeghi, and T. Schneider. Twinclouds: 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*,
- [9] D. Ferraiolo and R. Kuhn. Role-based access controls. In *15<sup>th</sup> 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*. 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*, . 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*. 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*.



ISSN(Online): 2320-9801  
ISSN (Print): 2320-9798

**International Journal of Innovative Research in Computer and Communication Engineering**

**An ISO 3297: 2007 Certified Organization**

**Vol.3, Special Issue 4, April 2015**

**National Conference On Emerging Trends in Information, Digital & Embedded Systems (NC'e-TIDES -15)**

**Organized by**

**Dept. of ECE, Annamacharya Institute Of Technology & Sciences, Rajampet, Andhra Pradesh-516126, India held on 28<sup>th</sup> February 2015**

### **BIOGRAPHY**

**Gangireddy Prashanthi** is a Research Assistant in the Computer Science & Engineering Department, College of Annamacharya Institute of Technology and Science, JNTUA University. She received Master of Technology post graduation in 2015 from JNTUA, Ananthapur, India. Her research interests are networking security, Symmetric and convergent Algorithms etc.

**Z.Shobha Rani** is a Research Assistant professor in the Computer Science & Engineering Department, Master of Technology in sreedeviwomens Engineering college, Gandipet, Hyderabad in 2012 Bachelor of Technology in Annamacharya Institute of Technology and Science in 2005, from JNTU Ananthapur University, India. Her research interests are Databases, Networking And Data mining.