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

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 3, March 2017

A Hybrid Cloud for Secure Authorized Deduplication with Multi-Keyword Ranked Search over Encrypted Data

Sucheta B. Patil [1], Mayank Bhatt [2]

M.Tech Scholar, Computer Science & Engg, LNCTS (RIT), Indore, India [1]

Assistant Professor, Computer Science & Engg, LNCTS (RIT), Indore, India [2]

ABSTRACT: Data deduplication is one of important data compression techniques for remove 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 private sensitive data while supporting deduplication, the convergent encryption technique has been proposed to encrypt the data before outsourcing.This is different from the universal duplication systems. The discriminatoryprivileges of users are further considered in duplicate check apart from the data itself. In hybrid cloud architecture authorized duplicate check supported by various new duplication constructions.This paper also presents a secure multi-keyword ranked search scheme over encrypted cloud data, which individually supports aggressive update operations like deletion and insertion of documents. Based on the definitions identify in the proposed security model, our scheme is secure. Proof of the concept implemented in this paper by conducting test-bed experiments.

KEYWORDS: Deduplication, authorized duplicate check, confidentiality, hybrid cloud, multi-keyword.

I. INTRODUCTION

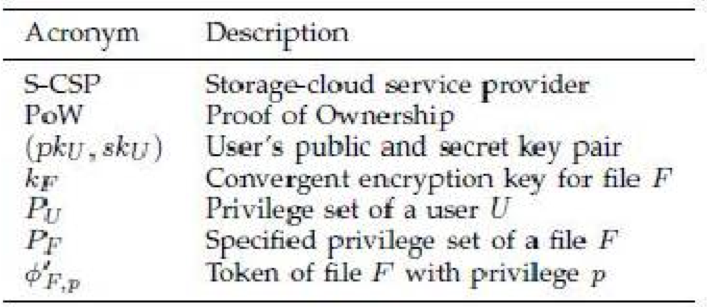
Cloud computing provides apparently unlimited “virtualized” resources to users as services across the wholeInternet, while hiding platform and implementation details. Today’s cloud service providers offer both highly available storage and largely parallel computing resources at relatively low costs. As cloud computing becomes usual, 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 negative challenge of cloud storage services is the management of the ever-increasing volume of data. To make data management scalable in cloud computing, deduplicationhas been a well-known technique and has attracted more and more observation 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 unneeded data by keeping only one physical copy and referring other redundant data to that copy. Deduplication can also take place at the blocklevel, which eliminates duplicate blocks of data that occur in distinct files.Although data de-duplication brings a lot of advantages, security and privacy concerns arise as users sensitive data are susceptible to both the insider and outsider strikes, When compares the traditional encryption with data duplication. This convergent encrypt provides one convergent key to encrypt/decrypt the data, which is obtained by calculate the cryptographic hash value of the content of the data copy. After completion of key generation and data encryption, users keep the keys and send the cipher text to the cloud. Since the encryption operation is deterministic and is derived from the data content, identical data copies generate the same convergent key and hence the same cipher text.Traditional encryption requires different users to encrypt their data with their own keys by which identical data copies of different users will lead to different cipher texts, making deduplication impossible. Convergent encryption [4] has been proposed to impose 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.

DOI: 10.15680/IJIRCCE.2017. 0503303

Whenever the key is generated users keep the keys and send the cipher text to the cloud. In order 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. Hence convergent encryption allows the cloud to perform deduplication on the ciphertexts and the proof of ownership preventsthe unauthorized user to access the file. To preserve data privacy and struggle unwanted accesses in the cloud and away from, sensitive data, for example, emails, personal health records, photo albums, videos, land documents, financial transactions, and so on, may have to be encrypted by data holder before outsourcing to the business public cloud; on the other hand, obsoletes the traditional data use service based on plaintext keyword search. The minor solution of downloading all the information and decrypting nearby is clearly impossible, due to the enormous amount of bandwidth cost in cloud scale systems. Furthermore, excepting the local storage management, storing data into the cloud supplies no purpose except they can be simply searched and operated.

1.1 Contribution

For solving the problems of deduplication we consider a hybrid cloud architecture consisting of a public cloud and a private cloud. Different privilege levels have been allocated to securely perform duplicate check in private cloud. A new deduplication system supporting differential duplicate check is proposed under this hybrid cloud architecture where the S-CSP resides in the public cloud.



Acronym	Description
S-CSP	Storage-cloud service provider
PoW	Proof of Ownership
(pk_U, sk_U)	User's public and secret key pair
k_F	Convergent encryption key for file F
P_U	Privilege set of a user U
P_F	Specified privilege set of a file F
$\phi'_{F,p}$	Token of file F with privilege p

TABLE 1
Notations Used in This Paper

The user is only permit to perform the duplicate check for files marked with the corresponding privileges. Furthermore, we boost 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 execute the duplicate check. Furthermore, such unauthorized users cannot decrypt the ciphertext even collude with the S-CSP. Finally, we implement a prototype of the proposed authorized duplicate check and conduct testbed experiments to evaluate the overhead of the prototype. We show that the above is minimal compared to the normal convergent encryption and file upload operations.

II. PRELIMINARIES

In this section, we first define the notations used in this paper, study some secure primitives used in our secure deduplication.

Symmetric encryption. Symmetric encryption uses a common secret key *to* encrypt and decrypt information.
A symmetric encryption scheme consists of three Primitive functions:

- KeyGenSE(1_) -> *κ* is the key generation algorithm that generates *κ* using security parameter 1;
- DecSE(*κ*,*C*) -> *M* is the symmetric decryption algorithm that takes the secret *κ* and ciphertext*C* and

then outputs the original message *M*.

- EncSE(κ,M) -> C is the symmetric encryption algorithm that takes the secret κ and message M and

then outputs the ciphertext$C$.

Convergent encryption. Convergent encryption [3], [4] provides data confidentiality in deduplication. A data owner derives a convergent key from each original data copy and encrypts the data copy with the convergent key.

- KeyGenCE(M)-> K is the key generation algorithm that maps a data copy M to a convergent key K
- EncCE(K,M)-> C is the symmetric encryption algorithm that takes both the convergent key K and the data copy M asinputs and then outputs a ciphertext$C$.
- DecCE(K,C) -> 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

- TagGen(M)-> T(M) is the tag generation algorithm that plan the original data copy M and outputs a tag T(M).
- DecSE(κ,C)-> M is the symmetric decryption algorithm that takes the secret κ and ciphertext$C$ and

then outputs the original message M.

Proof of ownership. The notion of proof of ownership (PoW) [2] enables users to prove their ownership of data copies to the storage server. However PoW is implemented as an interactive algorithm (denoted by PoW) run by a prover (i.e., user) and a verifier (i.e.storage server).

Identification Protocol. An identification protocol 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.

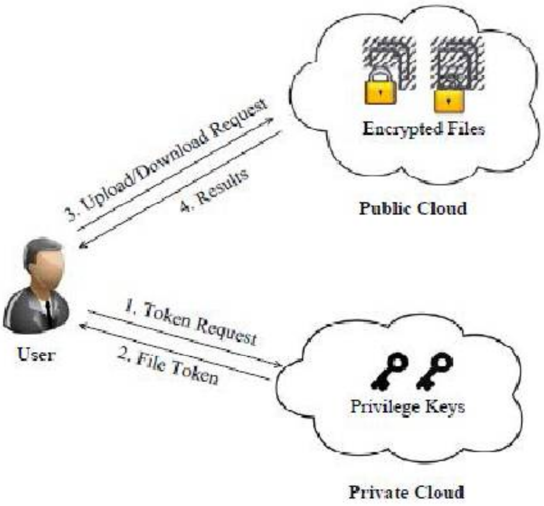


Fig. 1. Architecture for Authorized Deduplication

III. SYSTEM MODEL

3.1 Hybrid Architecture for Secure Deduplication

At a high level, our setting of interest is an enterprise network, consisting of a group of affiliated clients (for example, employees of a company) who will use the S-CSP and store data with deduplication technique. In this setting, deduplication can be frequently used in these settings for data backup and disaster recovery applications while much reducing storage space. Such systems are widespread and are often more suitable to user file backup and synchronization applications than richer storage abstractions. There are three entities defined in our system, that is, *users*, *private cloud* and S-CSP in *public cloud* as shown in Fig. 1. The S-CSP performs deduplication by checking if the contents of two files are the same and stores only one of them. Each privilege is represented in the form of a short message called *token*. Each file is associated with some *file tokens*, which denote the tag with specified privileges. Role of the private cloud server will be explained in the paper. block-leveldeduplication can be easily deduced from file-level deduplication, which is similar. Specifically, to upload a file, a user first performs the file-level duplicate check. If the file is a duplicate, then all its blocks must be duplicates as well.

- *S-CSP*->This is an entity that provides a data storage service in public cloud. The S-CSP provides the data outsourcingservice and stores data on behalf of the users.
- *Data Users*->A user is an entity that wants to outsource data storage to the S-CSP and access the data later. In a storage system supporting deduplication, the user only uploads unique data but does not upload any duplicate data to save the upload bandwidth, which may be owned by the same use or different users. Every single file is protected with the convergent encryption key and privilege keys to realize the authorized deduplication with differential privileges.
- *Private Cloud*->Compared with the traditional deduplication architecture in cloud computing, this is a new entityintroduced for facilitating user's secure usage of cloud service. Private Keys are managed by private cloud in order to give them privileges as per their designation.

3.2 Design Goals

We have proposed a new deduplication system for the following:

- Differential Authorization->Each authorized user is able to get his/her individual token of his file to perform duplicate check based on his privileges. Under this assumption, any user cannot generate a token for duplicate check out of his privileges or without the aid from the private cloud server.
- Authorized Duplicate Check->Authorized user is able to use his/her individual private keys to generate query for certain file and the privileges he/she owned with the help of private cloud, while the public cloud performs duplicate check directly and tells the user if there is any duplicate.
- Unforgeability of file token/duplicate-check token->Unauthorized users without appropriate privileges or file should be prevented from getting or generating the file tokens for duplicate check of any file stored at the S-CSP. The duplicate check token of users should be issued from the private cloud server in our scheme

Multi-keyword ranked search use a secure, efficient and dynamic search scheme which supports not only the accurate multi-keyword ranked search but also the dynamic deletion and insertion of documents. It construct a special keyword balanced binary tree as the index, and propose a "Greedy Depth-first Search" algorithm to obtain better efficiency than linear search. In addition, the parallel search process can be carried out to further reduce the time cost. The security of the scheme is protected against two threat models by using the secure kNN algorithm model are combined in the index construction and query generation to provide multi-keyword ranked search. The secure kNN algorithm is used to encrypt the index and query vectors, and meanwhile ensure accurate relevance score calculation between encrypted index and query vectors. To oppose different attacks in different threat models, we construct two secure search schemes: the basic dynamic multi-keyword ranked search (BDMRS) scheme in the known ciphertext model, and the enhanced dynamic multi-keyword ranked search (EDMRS) scheme in the known background model[7].

IV. SECURE DEDUPLICATION SYSTEMS

Main Idea -To support authorized deduplication, the tag of a file F will be determined by the file F and the privilege. To show the difference with traditional notation of tag, we call it file token instead. To support authorized access, a secret key kpwill be bounded with a privilege p to generate a file token.

4.1 A First Attempt

Before introducing our construction of differential deduplication, we present a straightforward attempt with the technique of token generation TagGen(*F, kp*)above to design such a deduplication system. The main idea of this basic construction is to issue corresponding privilege keys to each user, who will compute the file tokens and perform the duplicate check based on the privilege keys and files. In more details, suppose that there are *N* users in the system and the privileges in the universe is defined as *P = (p*1*, . . . ,ps).* For each privilege *p* in *P*, a private key *kp*will be selected. For a user *U* with a set of privileges *PU*, he will be assigned the set of keys *(kpi)pi∈PU.*

File Uploading- Suppose that a data owner U with privilege set PU wants to upload and share a file F with users who have the privilege set PF = fpjg. The user computes and sends S-CSP the file token ϕ′ F;p= TagGen(F, kp) for all p 2 PF .

• If a duplicate is found by the S-CSP, the user proceeds proof of ownership of this file with the S-CSP. If the proof is passed, the user will be assigned a pointer, which allows him to access the file.

• Otherwise, if no duplicate is found, the user computes the encrypted file CF = EncCE(kF , F) with the convergent key kF= KeyGenCE(F) and uploads (CF , fϕ′ F;p g) to the cloud server. The convergent key kFis stored by the user locally.

File Retrieving- Suppose a user wants to download a file F. It first sends a request and the file name to the S-CSP. Upon receiving the request and file name, the S-CSP will check whether the user is eligible to download F. If failed, the S-CSP sends back an abort signal to the user to indicate the download failure. Otherwise, the S-CSP returns the corresponding ciphertextCF.

4.2 Proposed System Description

We have introduced hybrid cloud architecture in our proposed deduplication system. The private keys for privileges will not be issued to users directly, which will be kept and managed by the private cloud server instead. In this way, the users cannot share these private keys of privileges in this proposed construction, which means that it can prevent the privilege key sharing among users in the above straightforward construction. To get a file token, the user needs to send a request to the private cloud server. The intuition of this construction can be described as follows. To perform the duplicate check for some file, the user needs to get the file token from the private cloud server. The private cloud server will also check the user's identity before issuing the corresponding file token to the user. The authorized duplicate check for this file can be performed by the user with the public cloud before uploading this file. Based on the results of duplicate check, the user either uploads this file or runs PoW. Before giving our construction of the deduplication system, we define a binary relation R = *f*((*p, p′*)*g* as follows. Given two privileges*p* and *p′* , we say that *p* matches *p′* if and only if R(*p, p′*) = 1.

System Setup. An identification protocol _ = (Proof, Verify) is also defined, where Proof and Verify are the proof and verification algorithm respectively. Furthermore, each user *U* is assumed to have a secret key *skU*to perform the identification with servers. Assume that user *U* has the privilege set *PU*. It also initializes a PoW protocol POW for the file ownership proof. The private cloud server will maintain a table which stores each user's public information *pkU*and its corresponding privilege set *PU*. The file storage system for the storage server is set.

File Uploading. Suppose that a data owner wants to upload and share a file *F* with users whose privilege belongs to the set *PF = fpjg*. The data owner needs interact with the private cloud before performing duplicate check with the S-CSP. Data ownerperforms an identification to prove its identity with private key *skU*. If it is passed, the private cloud server will find thecorresponding privileges *PU* of the user from its stored table list. The user computes and sends the file tag *ϕF*= TagGen(*F*) tothe private cloud server, who will return *fϕ′ F;p_* = TagGen(*ϕF, kp_*)*g*back to the user for all *p_* satisfying R(*p, p_*) = 1 and *p 2PU*. Then, the user will interact and send the file token *fϕ′ F;p_g* to the S-CSP.

• If a file duplicate is found, the user needs to run the PoW protocol POW with the S-CSP to prove the file ownership. If the proof is passed, the user will be provided a pointer for the file. Furthermore, a proof from the S-CSP will be returned, which could be a signature on *fϕ′ F;p_g, pkU*and a time stamp. The user sends the privilege set *PF = fpjg*for the file *F* as well as the proof to the private cloud server. Upon receiving the request, the private cloud server first verifies the proof from the S-CSP. If it is passed, the private cloud server computes *fϕ′ F;p_* = TagGen(*ϕF, kp_*)*g* for all *p_* satisfying R(*p, p_*) = 1 for each *p 2 PF -PU*, which will be returned to the user. The user also uploads these tokens of the file *F* to the private cloud server.Otherwise, if no duplicate is found, a proof from the S-CSP will be returned, which is also a signature on $f\phi'_{F;p\_}g$, $pk_U$and a time stamp. The user sends the privilege set $P_F = fp_jg$for the file F as well as the proof to the private cloud server. Upon receiving the request, the private cloud server first verifies the proof from the S-CSP. If it is passed, the private cloud server computes $f\phi'_{F;p\_} =$ TagGen(ϕ_F, $k_{p\_}$)g for all $p\_$ satisfying R(p, $p\_$) = 1 and p 2 P_F . Finally, the user computes the encrypted file C_F = Enc$_{CE}$(k_F, F) with the convergent key k_F= KeyGen$_{CE}$(F) and uploads fC_F, $f\phi'_{F;p_}Gg$with privilege P_F .

File Retrieving. The user downloads his files in the same way as the deduplication system in Section 4.1. That is, the user can recover the original file with the convergent key $k_F$after receiving the encrypted data from the S-CSP.

4.3 Further Enhancement

We design and implement a new system which could protect the security for predicatable message. The *mainidea* of our technique is that the novel encryption key generation algorithm. For simplicity, we will use the hash functions to define the tag generation functions and convergent keys in this section. In traditional convergent encryption, to support duplicate check, the key is derived from the file F by using some cryptographic hash function $kF= H(F)$. To avoid the deterministic key generation, the encryption key $kF$for file F in our system will be generated with the aid of the private key cloud server with privilege key kp. The encryption key can be viewed as the form of $kF;p= H0(H(F), kp) \oplus H2(F)$, where $H0$,H and $H2$ are all cryptographic hash functions. The file F is encrypted with another key k, while k will be encrypted with $kF;p$. In this way, both the private cloud server and S-CSP cannot decrypt the ciphertext. Furthermore, it is semantically secure to the S-CSP based on the security of symmetric encryption.

V. IMPLEMENTATION

Proposed cloud storage systems that offer privacy, reliability and authentication of client data against a UN trusted cloud provider. This OTP used to see data in cloud and it can be used once only in a time, when you search a file and want to see the file, the OTP will send to the email or to the phone and getting the OTP use the OTP to utilize the file . Presently in the existing system the cloud server hosts third-party data storage and get back services. As information may have sensitive information, the cloud servers cannot be fully hand over in protecting data. For this cause, outsourced files must be encrypted. Any type of data leakage that would involve data privacy is considered as undesirable[7]. To enable ranked searchable symmetric encryption for effective utilization of outsourced and encrypted cloud data under the aforementioned model, our system design should achieve the following security and performance guarantee.Specifically, we have the following goals: i) Ranked keyword search: to explore different mechanisms for designing effective ranked search schemes based on the existing searchable encryption framework; ii) Security guarantee: to prevent cloud server from learning the plaintext of either the data files or the searched keywords, and achieve the “asstrong- as-possible” security strength compared to existing searchable encryption schemes; iii) Efficiency: above goals should be achieved with minimum communication and computation overhead[8].

A Private Server program is used to model the private cloud which manages the private keys and handles the file token computation. A Storage Serverprogram is used to model the S-CSP which stores and deduplicate files.

Our implementation of the **Client** provides the following function calls to support token generation and deduplication along the file upload process.

- FileTag(File) - It computes SHA-1 hash of the File as File Tag;
- TokenReq(Tag, UserID) - It requests the Private Server for File Token generation with the File Tag and User ID;
- DupCheckReq(Token) - It requests the Storage Server for Duplicate Check of the File by sending the file token received rom private server;
- ShareTokenReq(Tag, {Priv.}) - It requests the Private Server to generate the Share File Token with the File Tag and Target Sharing Privilege Set;
- FileEncrypt(File) - It encrypts the File with Convergent Encryption using 256-bit AES algorithm in cipher block chaining (CBC) mode, where the convergent key is from SHA-256 Hashing of the file;
- FileUploadReq(FileID, File, Token) – It uploads the File Data to the Storage Server if the file is Unique and updates the File Token stored. Our implementation of the **Private Server** includes corresponding request handlers for the token generation and maintains a key storage with Hash Map.

- TokenGen(Tag, UserID) - It loads the associated privilege keys of the user and generate the token with HMAC-SHA-1 algorithm.

Also Proposed work is based on a secure tree-based search scheme over the encrypted cloud data, which supports multi-keyword ranked search and dynamic operation on the document collection. Specifically, the vector space model and the broadly-used “term frequency (TF) × inverse document frequency (IDF)” model are combined in the index construction and query generation to provide multi-keyword ranked search. The secure kNN algorithm is used to encrypt the index and query vectors, and meanwhile ensure accurate relevance score calculation between encrypted index and query vectors. To oppose different attacks in different threat models, we construct two secure search schemes: the basic dynamic multi-keyword ranked search (BDMRS) scheme in the known ciphertext model, and the enhanced dynamic multi-keyword ranked search (EDMRS) scheme in the known background model.

VI. CONCLUSION

In this paper, we presented 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. Also In this paper, a secure, efficient and dynamic search scheme is proposed, which supports not only the accurate multi-keyword ranked search but also the dynamic deletion and insertion of documents. We construct a special keyword balanced binary tree as the index, and propose a “Greedy Depth-first Search” algorithm to obtain better efficiency than linear search. In addition, the parallel search process can be carried out to further reduce the time cost. The security of the scheme is protected against two threat models by using the secure kNN algorithm. Experimental results demonstrate the efficiency of our proposed scheme.

REFERENCES

[1] Jin Li, Yan Kit Li, Xiaofeng Chen, Patrick P. C. Lee, Wenjing Lou “ A Hybrid Cloud Approach for Secure Authorized De-duplication” in vol: pp no-99, IEEE, 2014

[2] 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 andCommunications Security, pages 491–500. ACM, 2011.

[3] M. Bellare, S. Keelveedhi, and T. Ristenpart. Message-locked encryption and secure deduplication. In EUROCRYPT, pages 296– 312, 2013.

[4] 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.

[5] B. Wang, S. Yu, W. Lou, and Y. T. Hou, “Privacy-preserving multi-keyword fuzzy search over encrypted data in the cloud,” in IEEEINFOCOM, 2014.

[6] Zhihua Xia, Member, Xinhui Wang, Xingming Sun and Qian Wang, Member, IEEE, “A Secure and Dynamic Multi-keyword Ranked Search Scheme over Encrypted Cloud Data”, IEEE transactions on Parallel and Distributed systems,2015

[7] C. Orencik, M. Kantarcioglu, and E. Savas, “A practical and secure multi-keyword search method over encrypted cloud data,” in Cloud Computing (CLOUD), 2013 IEEE Sixth International Conference on. IEEE, 2013.

[8] C. Wang, N. Cao, K. Ren, and W. Lou, “Enabling secure and efficient ranked keyword search over outsourced cloud data,” IEEE Transactions on Parallel and Distributed Systems, vol. 23,2015.

[9] W. Zhang, S. Xiao, Y. Lin, T. Zhou, and S. Zhou, “Secure ranked multi-keyword search for multiple data owners in cloud computing,” in Dependable Systems and Networks (DSN), 2014 44th AnnualIEEE/IFIP International Conference on. IEEE, 2014.

[10] J. Yuan and S. Yu. Secure and constant cost public cloud storage auditing with deduplication. IACR Cryptology ePrint Archive, 2013:149, 2013.