ISSN 2395-1621

Secured Approach for Hybrid Cloud De-Duplication

[#1]Miss.Karishma Bagwan, [#2]Miss.Komal Kakkad, [#3]Miss.Sumayya Shaikh,
[#4]Prof. Murkute. P.K

[1]karishmabagwan11@gmail.com
[2]KomalKakkad20@gmail.com
[3]shifa4200@gmail.com

[#123]Department of Computer Engineering
[#4]Prof. Department of Computer Engineering

Al-Ameen College of Engineering
Koregaon-Bhima,Tal-shirur Dist.Pune-412216

ABSTRACT

A hybrid cloud is a combination of public and private clouds bound together by either standardized or proprietary technology that enables data and application portability. Proposed system aiming to efficiently solving the problem of deduplication with differential privileges in cloud computing. A hybrid cloud architecture consisting of a public cloud and a private cloud and the data owners only outsource their data storage by utilizing public cloud while the data operation is managed in private cloud. To make data management scalable in cloud computing, deduplication has been a very well-known technique recently is use. Deduplication reduces your bandwidth requirements, speeds up the data transfers, and it keeps your cloud storage needs to a minimum. Proposed system present several new deduplication constructions supporting authorized duplicate check in hybrid cloud architecture. To maintain the confidentiality of data the convergent encryption technique has been used to encrypt the data before outsourcing. Authorized deduplication system support differential authorization duplicate check. As a proof of concept, a prototype is implemented in authorized duplicate check scheme and conduct test bed experiments using prototype, authorized duplicate check scheme incurs minimal overhead compared to normal operations.

Keywords: **Deduplication, Authorized duplicate check, Hybrid cloud, Covergent encryption.**

ARTICLE INFO

Article History

Received: 19th May 2016

Received in revised form :

20th May 2016

Accepted: 24rd May 2016

Published online :

25th May 2016

I. INTRODUCTION

DupLESS starts with the observation that brute-force cipher text recovery in a CE-type scheme can be default with by using a key server to derive keys, instead of setting keys to be hashes of messages. Access to the KS is preceded by authentication, which stops external attackers. The increased cost slows down brute-force attacks from compromised clients, and now the KS can function as a single point of control for implementing rate-limiting measures. We can expect that by scrupulous choice of rate-limiting policies and parameters, brute-force attacks originating from compromised clients will be rendered less effective, while normal usage will remain unaffected. We start by looking at secret-parameter MLE, an extension to MLE which endows all clients with a system wide secret parameter. The rational here is that if disk is unknown to the attacker, a high level of security can be achieved, but even if diskis leaked, security falls to that of regular MLE. A server-aided MLE scheme then is a transformation where the secret key is restricted to the KS instead of being available to all clients. One simple approach to get server-aided MLE is to use a PRF F, with a secret key K that never leaves the KS. A client would send a hash H of a file to the KS and receive back a message-derived key K, F(K,H). The other steps are as in CE. However, this approach proves unsatisfying from a security perspective. The KS here becomes a single point of failure, violating our goal of compromise resilience: an attacker can obtain hashes of files after gaining access to the KS, and can recover files with brute force attacks. Instead, DupLESS employs an oblivious PRF (OPRF) protocol between the KS and clients, which ensures that the KS learns nothing about the client inputs or the resulting PRF outputs, and that clients learn nothing about the key. In Section 4, we propose a new server-aided MLE scheme DupLESSMLE which combines a CE-type base with the OPRF protocol based on RSA blind-

© 2015, IERJ All Rights Reserved

signatures. Thus, a client, to store a file M, will engage in the RSA OPRF protocol with the KS to compute a message derived key K, then encrypt M with K to produce a cipher text C data. The clients secret key will be used to encrypt K to produce a key encapsulation cipher text C key. Both C key and C data are stored on the SS. Should two clients encrypt the same file, then the message-derived keys and, in turn, C data will be the same building a system around DupLESSMLE requires careful design in order to achieve high performance.

II. RELATED WORK

The Proof of Ownership (PoW) is introduced by Halevi. It is challenge-response protocol enabling a storage server to check whether a requesting entity is the data owner, based on a short value. That is, when a user wants to upload a data file (D) to the cloud, he first computes and sends a hash value hash = H(D) to the storage server. This latter maintains a database of hash values of all received files, and looks up hash. If there is a match found, then D is already outsourced to cloud servers. As such, the cloud tags the cloud user as an owner of data with no need to upload the file to remote storage servers. If there is no math, then the user has to send the file data (D) to the cloud. This client side de duplication, referred to as hash-as-a proof, presents several security challenges, mainly due to the trust of cloud users assumption. This Section presents a security analysis of existing PoW schemes.

Security Analysis:

Despite the significant resource saving advantages, PoW schemes bring several security challenges that may lead to sensitive data. Data confidentiality disclosure hash-as-a-proof schemes (e.g. Dropbox) introduce an important data confidentiality concern, mainly due to the static proof client side generation.

III. PROPOSED ALGORITHM

AES Parameters:

At the start of the Cipher, the input is copied to the State array using the conventions described. After an initial Round Key addition, the State array is transformed by implementing a round function 10, 12, or 14 times (depending on the key length), with the final round differing slightly from the first ***Nr*** 1 rounds. The final State is then copied to the output .The round function is parameterized using a key schedule that consists of a one-dimensional array of four-byte words derived using the Key Expansion routine.

The Cipher is described in the pseudo code. The individual transformations -**SubBytes()**, **ShiftRows()**, **MixColumns()**, and **AddRoundKey()** – process the State and are described in the following subsections. The array **w[]** contains the key schedule.

All ***Nr*** rounds are identical with the exception of the final round, which does not include the **MixColumns()** transformation.

Cipher(byte in[4*Nb], byte out[4*Nb], word w[Nb*(Nr+1)]) begin byte state[4,Nb]

state = in

AddRoundKey

(state, w[0, Nb-1])

for round = 1 step 1 to Nr–1 SubBytes(state)

ShiftRows(state)

MixColumns(state)

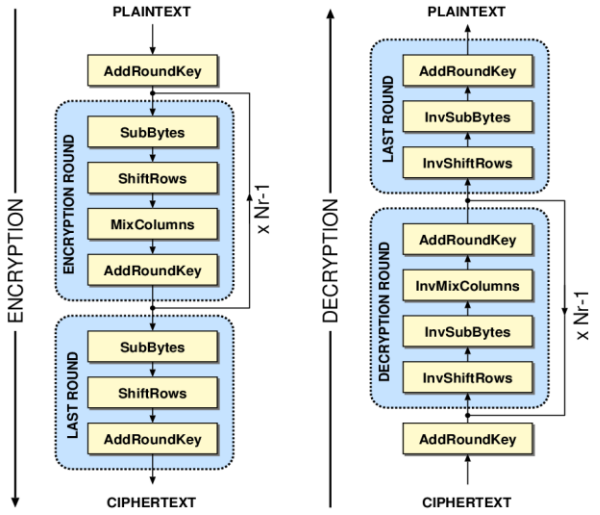
AddRoundKey(state, w[round*Nb, (round+1)*Nb-1])

end for

SubBytes(state) ShiftRows(state)

AddRoundKey(state, w[Nr*Nb, (Nr+1)*Nb-1])

out = state end



Rolling Hash (Rabin-Karp Algorithm)

Objective

If we have text string S and pattern string P, we want to determine whether or not P is found in S, i.e. P is a substring of S.Say P has length L and S has length n. One way to search for P in

1. Hash P to get h(P) O(L)

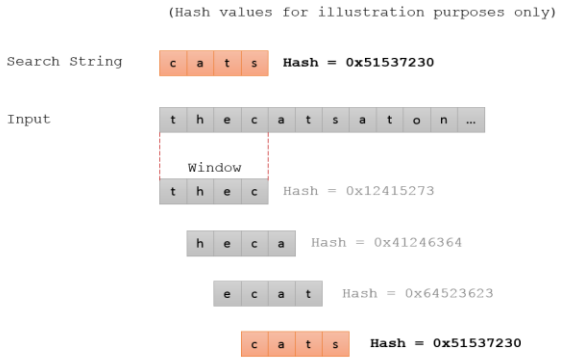
2. Iterate through all length L substrings of S hashing those substrings and comparing toh(P) O(nL)

3. If a substring hash value does match h(P), do a string comparison on that substring and P, stopping if they do match and continuing if they do not. O(L)

This method takes O(nL) time. We can improve on this runtime by using a rolling hash. In step 2. we looked at O(n) substrings independently and took O(L) to hash them all. These substrings however have a lot of overlap. For example,

© 2015, IERJ All Rights Reserved

looking at length 5 substrings of “algorithms”,the first two substrings are “algor” and “lgori”. Wouldn’t it be nice if we could take advantage of the fact that the two substrings share “lgor”, which takes up most of each substring, to save some computation? It turns out we can with rolling hashes.



IV. IMPLEMENTATION

In our system we will be implementing a project that includes the public cloud and the private cloud and also the hybrid cloud which is a combination of the both public cloud and private cloud. In general by if we used the public cloud we can’t provide the security to our private data and hence our private data will be loss. So that we have to provide the security to our data for that we make a use of private cloud also. When we use a private clouds the greater security can be provided. In this system we also provides the data deduplication .which is used to avoid the duplicate copies of data .User can upload and download the files from public cloud but private cloud provides the security for that data .that means only the authorized person can upload and download the files from thepublic cloud for that user generates the key and stored that key onto the private cloud. at the time of downloading user request to the private cloud for key and then access that Particular file.

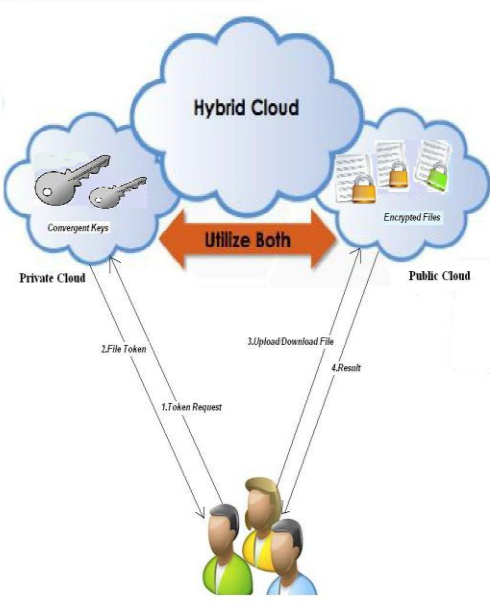


Fig 1. Architecture of Authorized De-duplication

Data User:

A user is an entity that want to access the data or files from S-SCP. User generate the key and store that key in private cloud. In storage system supporting de-duplication, The user only upload unique data but do not upload any duplicate data to save the upload bandwidth, which may be owned by the same user or different users. Each file is protected by convergent encryption key and can access by only authorized person. In our system user must need to register in private cloud for storing token with respective file which are store on public cloud. When he want to access that file he access respective token from private cloud and then access his files from public cloud. token consist of file content F and convergent key KF.

Private Cloud:

In general for providing more security user can use the private cloud instead of public cloud. User store the generated key in private cloud. At the time of downloading system ask the key to download the file. User can not store the secrete key internally. for providing proper file. When user want to access the key he first check authority of user then an then provide key. Protection to key we use private cloud. Private cloud only store the convergent key with respective

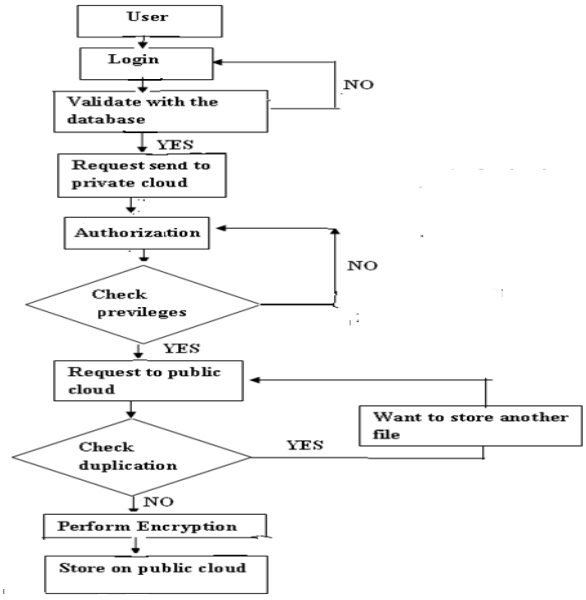


Fig 2. Flow Diagram of Proposed Method

Public Cloud:

Public cloud entity is used for the storage purpose. User upload the files in public cloud. Public cloud is similar as S-CSP. When the user want to download the files from public cloud,it will be ask the key which is generated or stored in private cloud. When the users key is match with files key at that time user can download the file, without key user cannot access the file. Only authorized user can access the file. In public cloud all files are stored in encrypted format. If any chance unauthorized person hack our file, but without the secrete or convergent key he doesn’t access original file. On public cloud there are lots of files are store each user access its respective file if its token matches with S-CSP server token.

© 2015, IERJ All Rights Reserved

V. SIMULATION RESULTS

We proWe did POC (proof of concept) study by taking data stored in personal computer from 7 employees working in group in an organization. In POC we considered few thousand files and analysed space requirement for these data, space saving with file level deduplication and space saving with chunk level deduplication. Space saving from file level deduplication is around 25% whereas chunk deduplication has spacing of around 37% below mentioned graph [Figure 3] demonstrates POC results.

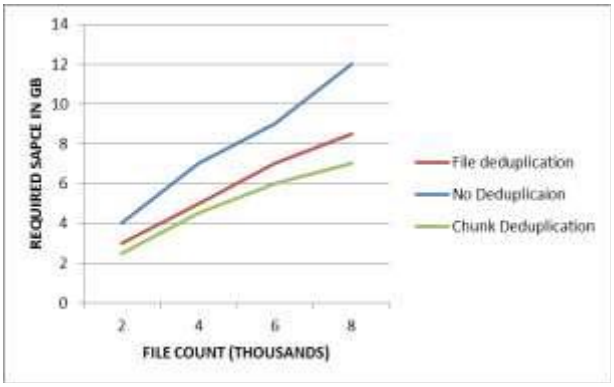
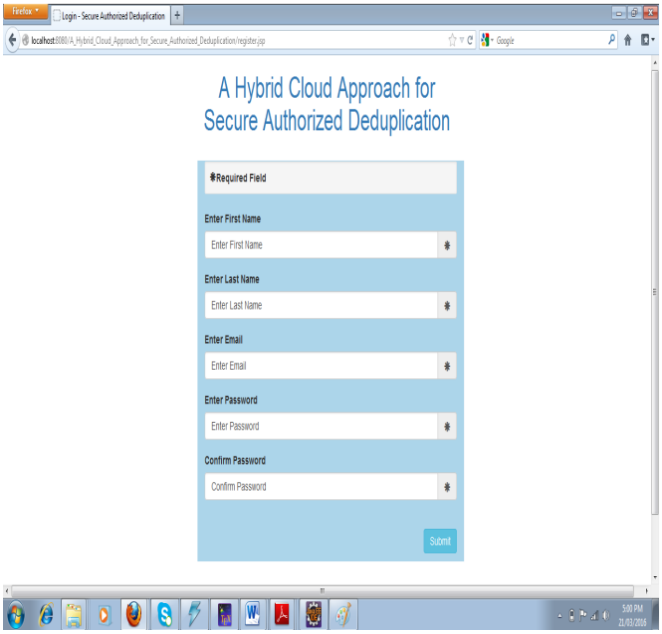
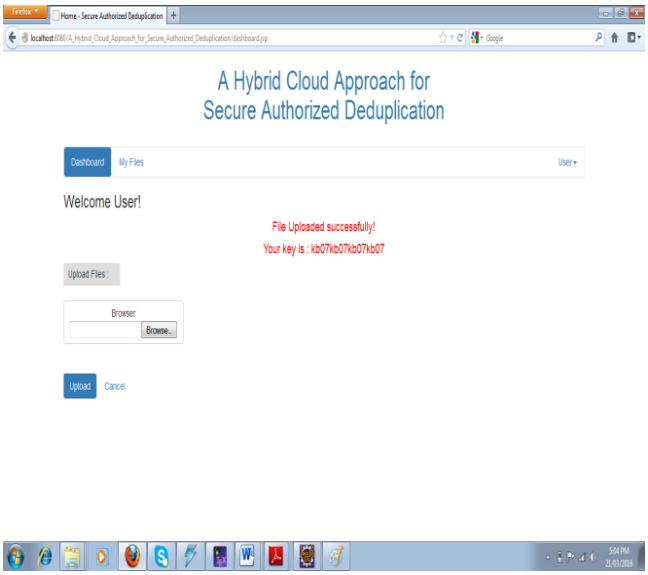


Fig 3. demonstrates POC results

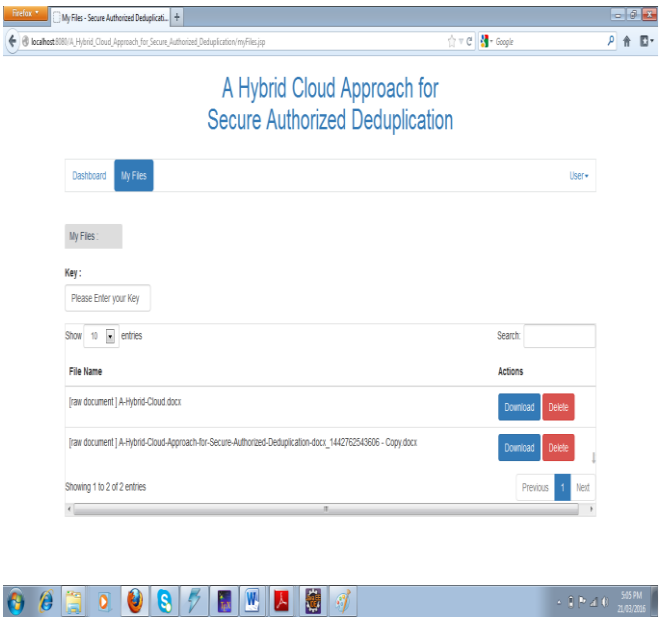
User Registration:
This is user registration window. User need to register once in order to use cloud.



File Upload:
This window allows user to browse file from computer and upload to cloud. When user upload files the system check for duplicate file in database. If file is present then system will maintain reference for existing file and will avoid duplicate. This operation is abstract from user.



File Download:
This window allows user to download previously uploaded files.



VI. CONCLUSION

In this paper we have proposed different techniques about data security, privacy and better confidentiality. Deduplication constructions supporting authorized duplicate check in hybrid cloud architecture, for which the duplicate-check tokens of files are generated by the private cloud server with private keys. Using Hybrid cloud architecture, it provides lot of benefits with the use of both public and private. Clouds and adopting deduplication to store data in the cloud will provide us better storage benefits at lower costs.

REFERENCES

[1] C. K Huang , L.- F Chien , and Y. -J Oyang , " Relevant Term Suggestion in Interactive Web Search Based on Contextual Information in Query Session Logs

© 2015, IERJ All Rights Reserved

, " J. Am. Soc. for Information science and Technology, vol. 54, no. 7, pp.638-649, 2013.

[2] D. Ferraiolo and R. Kuhn. Role-based access controls. In 15th NIST-NCSC National Computer Security Conf., 2013.

[3]GNULibmicrohttpd.http://www.gnu.org/software/libmicrohttpd/.

[4] Ian Clarke, Oskar Sandberg, Brandon Wiley, and Theodore W. Hong "secure cloud backup system with assured deletion and version control. In *ICDCS*, pages 617-624,2013.

[5] 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 of Parallel and Distributed Systems*, 2013.

[6] Jin Li, Yan Kit Li, XiaoFeng Chen, Patrick P.C.Lee,Wenjing Lou: A Hybrid Cloud Approach for Secure Authorized Deduplication. IEEE transactions on parallel and distributed system Vol:PP NO:99 Year 2014.

[7] John Douceur, Atul Adya, William J. Bolosky, Dan Simon and Marvel, Theimer, " Reclaiming space from duplicate files in a serverless distributed file system", 2013. 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] K. Zhang, X. Zhou, Y. Chen, X. Wang, and Y. Ruan (2011), Sedic: privacy ware data intensive computing on hybrid clouds. In Proceedings of the 18th ACM conference on Computer and communications security, vol. 54, no. 7,2014.

[10]M. Bellare, S. Keelveedhi, and T. Ristenpart. Dupless: Serveraided encryption for deduplicated storage. In *USENIX Security Symposium*, 2013.

[11] M. Bellare, S. Keelveedhi, and T. Ristenpart. Message locked encryption and secure deduplication.In *EUROCRYPT*,pages 296–312, 2014.

[12] 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, 2013.

[13] M. Bellare, C. Namprempre, and G. Neven. Security proofs for identity-based identification and signature schemes. J. Cryptology,22(1):1–61, 2012.

© 2015, IERJ All Rights Reserved