INTERNATIONAL JOURNAL OF RESEARCH
In Advanced Engineering Technologies

Volume 5, Issue 3 SEP 2016

A Framework to Support File-level and Fine-grained Block-level Data De-duplication

[1]BASHETTI.KRUTHIKA, 2MANASA

[1] M. Tech Student, Department of CSE, A.M.R. Institute of Technology, Village Mavala,
Mandal Adilabad, District Adilabad, Telangana, India

[2] HOD, Department of CSE, A.M.R. Institute of Technology, Village Mavala,
Mandal Adilabad, District Adilabad, Telangana, India

ABSTRACT— ***De-duplicate copies of data can be eliminated by using data de-duplication technique .Data de-duplication technique is very useful in cloud storage to reduce the storage space and upload bandwidth. However, there is only one of copy for each file stored in cloud even if such a file is used by a large number of users. As a result, de-duplication system improves storage utilization while decreasing the reliability. Furthermore, the challenge of privacy for confidential data also arises when they are outsourced by users to cloud. Concentrate to enforce above security problems, this paper first introduce the notion of distributed reliable de-duplication system. We propose new distributed de-duplication systems with higher reliability in which the data parts are distributed across multiple cloud servers. The security requirements of data confidentiality and tag consistency are also achieved by a deterministic secret sharing scheme in distributed storage systems. Instead of using convergent encryption, as in previous de-duplication system; Security analysis says that our proposed de-duplication systems are secure in terms of the definitions specified in the proposed security model. As a proof of concept, we implement the proposed systems and demonstrate that the overhead is very limited in realistic environments.***

Index Terms— De-duplication, convergent encryption, confidentiality.

I. INTRODUCTION

We implement new distributed de-duplication system that has a lot of reliableness. In that information chunks are distributed across multiple numbers of cloud servers. De-duplication technique will save the memory area for the cloud storage service providers; it reduces the honesty of the system. Security analyses indicate that our de-duplication systems are secure in terms of the definitions specified in this security model. As an indication of conception, we tend to implement the projected systems that indicate the non-heritable aerial is extremely restricted in actual environments. De-duplication method improves storage utilization & it saves storage space that is why it's helpful in trade likewise as in educational. It is helpful in such application that has high de-duplication quantitative relation like as deposit storage system. Moreover, for the information privacy challenge is additionally arises a lot of. The lots of sensitive knowledge are decentralized by the users to cloud. Cryptography have been sometimes utilized, for to produce protection confidentiality before the decentralized knowledge into cloud. Most business storage No of service suppliers are oppose to use encoding over the info as a result of it's not possible to create de-duplication. The reason of that's the traditional encoding mechanism. During which as well as the general public key encoding and radial key encoding have require number of users to cipher their knowledge with own key. For the results of similar knowledge copy of the quantity of users can lead to the different information has been encrypted. To solve the issues of confidentiality and de-duplication, for finding the problem of de-duplication we tend to implement notation of the oblique encoding.

II. RELATED WORK

The Farsite distributed file system provides accessibility by replicating every file onto multiple desktop computers. Since this replication consumes significant space for storing, it's necessary to reclaim used space wherever possible. Measurement of over five hundred desktop file systems shows that just about half all consumed space is occupied by

duplicate files. We tend to present a mechanism to reclaim space from this incidental duplication to make it available for controlled file replication. Our mechanism includes 1) convergent encoding, that permits duplicate files to united into the space of a single file, although the files are encrypted with totally different users' keys, and 2) dish, a Self Arranging, Lossy, Associative database for aggregating file content and placement information in an exceedingly decentralized, scalable, fault-tolerant manner. Large-scale simulation experiments present that the duplicate-file coalescing system is ascendible, highly effective, and fault-tolerant

Cloud storage service suppliers like Dropbox, Mozy, and others perform de-duplication to avoid wasting area by solely storing one copy of every file uploaded. Ought to purchasers conventionally cipher their files, however, savings are lost. Message-locked encoding (the most outstanding manifestation of that is convergent encryption) resolves this tension. However it's inherently subject to tenacity attacks which will recover files falling into an acknowledged set. We propose an design that gives secure deduplicated storage resisting brute-force attacks, and are aware of it in an exceedingly system referred to as Dupless. In Dupless, shoppers encipher below message-based keys obtained from a key-server via an oblivious PRF protocol. It allows purchasers to store encrypted knowledge with an existing service, has the service perform de-duplication on their favor, and however achieves robust confidentiality guarantees. we tend to show that encoding for deduplicated storage are able to do performance and area savings on the point of that of victimization the storage service with plaintext knowledge .

An data dispersion rule (IDA) is developed that breaks a file F of length L = (F (into n items F,, 1 5 i 5n, every of length (F, 1 = L/m, in order that each m items do for reconstructing F[9]. Dispersion and reconstruction are computationally economical. The ad of the lengths (F, 1 is (n/m). L. Since n/m is often chosen to be about to I, the IDA is space eflicient. IDA has various applications to secure and reliable storage of data in pc networks and even on single disks, to fault-tolerant and economical transmission of data in networks, and to communications between processors in parallel computers. For the latter drawback incontrovertibly time-efficient and extremely fault tolerant routing on the n-cube is achieved, victimization simply constant size buffers. Classes and Subject Descriptors: [Coding and data Theory]: non-secret cryptography schemes in cloud storage. Although convergent cryptography has been extensively adopted for secure de-duplication, a critical issue of creating convergent cryptography sensible is to expeditiously and faithfully manage a large range of convergent keys. This paper makes the primary arrange to formally address the matter of achieving economical and reliable key management in secure de-duplication. We have a tendency to first introduce a baseline approach during which every user keeps a freelance master key for encrypting the merging keys and outsourcing them to the cloud. However, such a baseline key management theme generates a massive range of keys with the increasing range of users and needs users to dedicatedly shield the master keys. to the present finish, we have a tendency to propose Dekey, a replacement construction during which users don't need to manage any keys on their own however instead firmly distribute the convergent key shares across multiple servers. Security analysis demonstrates that Dekey is more secure in terms of the definitions per the proposed security model. As a symptom of conception, we have a tendency to implement Dekey exploitation the Ramp secret sharing theme and demonstrate that Dekey incurs restricted overhead in realistic environments.

Cloud storage systems have become more and more popular. A promising technology that keeps their price down is de-duplication, that stores solely one copy of continuation information. Client-side de-duplication tries to spot de-duplication opportunities already at the consumer and save the information measure of uploading copies of existing files to the server. During this work we tend to determine attacks that exploit client-side de-duplication, permitting an attacker to achieve access to arbitrary-size files of different users supported terribly tiny hash signatures of those files. Additional specifically, an attacker who knows the hash signature of a file will convert the storage service that it owns that file; thus the server lets the attacker transfer the whole

file. (In parallel to our work, a set of those attacks was recently introduced within the wild with respect to the Dropbox file synchronization service.) To overcome such attacks, we tend to introduce the notion of proofs-of-ownership (PoWs), that lets a consumer expeditiously encourage a server that that the consumer holds a file, instead of just some short info concerning it. We tend to formalize the thought of proof-of-ownership, beneath rigorous security definitions, and rigorous efficiency needs of petabyte scale storage systems. We have a tendency to then present solutions based mostly on Merkle trees and specific encodings, and analyze their security. We enforced one variant of the theme. Our performance measurements indicate that the theme incurs solely a tiny low overhead compared to naive client-side de-duplication.

III. FRAME WORK

To protect non-public knowledge the key sharing technique is employed that is similar to distributed storage systems .In this paper the key sharing technique is employed for cover of personal knowledge. In detail a file is divide and cipher into sections by victimization secret sharing technique. These sections are distributed over several independent storage servers. A cryptanalysis hash price of the content will be calculated and send to storage server because the mark of the fragment stored at every server. only the information user who initial transfer the information is needed to calculate and distribute such secret shares and following users own same knowledge copy don't got to calculate and stores these shares. Retrieve knowledge copies owner must access a minimum range of storage server by a validation and acquire the key shares to change the information. In totally different way, the approved uses can access the key shares knowledge copy. Another distinguishable feature of our proposal is that data completeness in closes tag consistency may be derived. To clarify additional if identical worth is keep in numerous cloud storage then de-duplication check by strategies. It cannot oppose the collision attack established by several servers. To our knowledge no connected work on secure de-duplication will justly address, the dependability and tag consistency drawback. The file level and block level de-duplication is employed for higher reliableness. The secret splitting technique is employed for protect knowledge. Our planned structure supports each ancient de-duplication strategies. Privacy, credibility and integrity may be achieved in our planned system. In answer to quite secret agreement attacks are thought-about. These are the attack on the information and the attack against servers. The information is secure once the opponent management restricted variety of storage servers.

a) Block Diagram/Architecture of Proposed System:-

When the user wants to upload and transfer the file from cloud storage at that point initial user request to the online server for uploading file. It suggests that solely approved user will transfer the file to internet server for that purpose it use the proof of ownership rule. User to prove their relation of associate owner to the factor possessed of knowledge copies to the storage server. When file is uploaded it splits into blocks i.e., by default size of block is 4KB.According to file size the block happens. After that de-duplication detection happens.

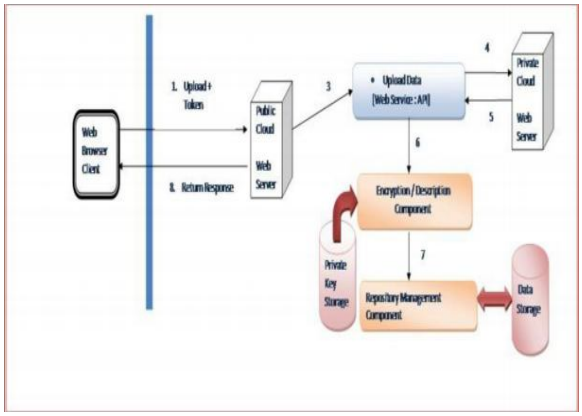


Fig: Workflow of upload/download

b) File Level De-duplication Systems:

To support efficient duplicate check, tags for every file are going to be computed and are sent to S-Cloud Service Providers. To transfer a file F, the user interacts with S-Cloud Service providers to perform the de-duplication. More exactly, the user first of all measures and sends the file tag ϕF = TagGen(F) to S-Cloud Service Providers for the file duplicate

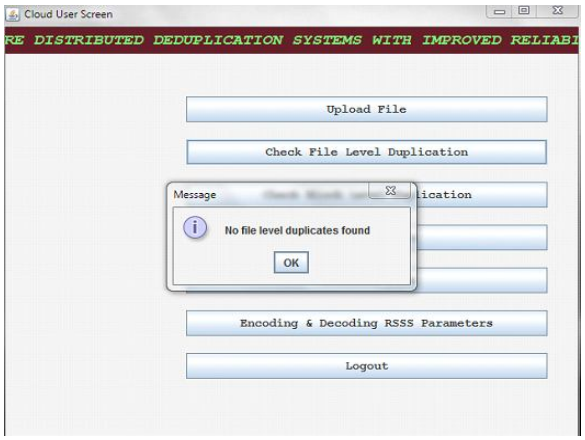
check. If a same is found the user measures and sends it to a server via a secure channel. Otherwise if no duplicate is found the method continues, i.e., secret sharing theme runs and therefore the user can transfer a file to CSP. To transfer a file the user will use the key shares and transfer it from the SCSP"s .This approach provides fault tolerance and permits the user to stay accessible although any restricted subsets of storage servers fail.

c) Block Level De-duplication Systems:

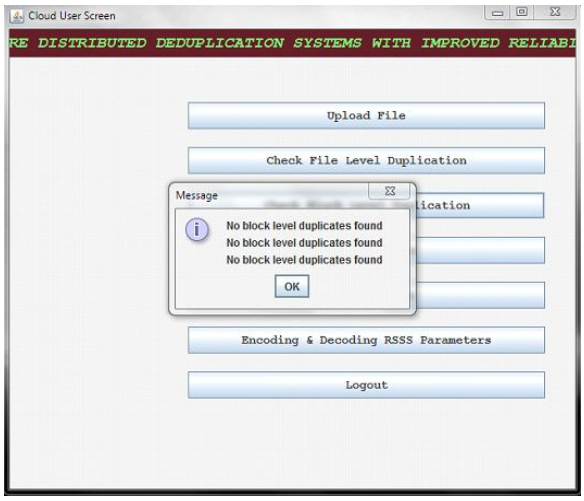
In this module we'll show to attain fine grained block-level distributed de-duplication systems. In a block level de-duplication system, the user conjointly must first off perform the file-level de-duplication before uploading his file .If no duplicate is found, the user divides this file into number of blocks and performs block-level de-duplication. The System setup is comparable to the file level de-duplication except the parameter changes. To download a block the user gets the key shares and transfers the blocks from CSP.

IV. EXPERIMENTAL RESULTS

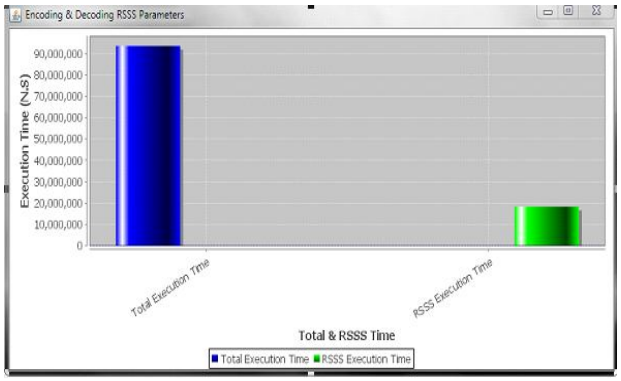
Our proposed system achieves very good results in experimentally. In our system we check the file level duplication and block level duplication also. In file level duplication it checks same file is existed or not in the cloud server. If same file is not existed it shows the acknowledgement like no duplicate is found.



The above screen shows that we are checking the file level duplication.



In similar way it checks the block level duplication also. In block level de-duplication it checks the each and every block in file.



Above screen shows that the runtime chart for Total and RSSS time

V. CONCLUSION

We projected the distributed de-duplication systems to improve the dependability of knowledge whereas achieving the confidentiality of the users' outsourced knowledge while not an encryption mechanism. Constructions were projected to support file-level knowledge de-duplication. The safety of tag consistency and integrity were achieved. De-duplication systems was enforced exploitation the Ramp secret sharing scheme and incontestable that it incurs tiny encoding/decoding

overhanging compared to the network transmission overhead in regular upload/download operations.

REFERENCES

[1] 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, 2002, pp. 617–624.

[2] Message-locked encryption and secure de-duplication,‖ in EUROCRYPT, 2013, pp. 296– 312.

[3] G. R. Blakley and C. Meadows, ―Security of ramp schemes,‖ in Advances in Cryptology: Proceedings of CRYPTO ’84, ser. Lecture Notes in Computer Science, G. R. Blakley and D. Chaum, Eds. Springer-Verlag Berlin/Heidelberg, 1985, vol. 196, pp. 242–268.

[4] A. D. Santis and B. Masucci, ―Multiple ramp schemes,‖ IEEE Transactions on Information Theory, vol. 45, no. 5, pp. 1720–1728, Jul. 1999.

[5] A. Shamir, ―How to share a secret,‖ Commun. ACM, vol. 22, no. 11, pp. 612–613, 1979.

[6] J. Li, X. Chen, M. Li, J. Li, P. Lee, and W. Lou, ―Secure de-duplication with efficient and reliable convergent key management,‖ in IEEE Transactions on Parallel and Distributed Systems, 2014, pp. vol. 25(6), pp. 1615–1625.

[7] W. K. Ng, Y. Wen, and H. Zhu, “Private data de-duplication protocols in cloud storage.” in Proceedings of the 27th Annual ACM Symposium on Applied Computing, S. Ossowski and P. Lecca, Eds. ACM, 2012, pp. 441– 446.

[8] M. Bellare, S. Keelveedhi, and T. Ristenpart, “Dupless: Serveraided encryption for deduplicated storage,” in USENIX Security Symposium, 2013

[9] J. Stanek, A. Sorniotti, E. Androulaki, and L. Kencl, “A secure data de-duplication scheme for cloud storage,” in Technical Report, 2013.

[10] W. K. Ng, Y. Wen, and H. Zhu, “Private data de-duplication protocols in cloud storage.” in Proceedings of the 27th Annual ACM Symposium on Applied Computing, S. Ossowski and P. Lecca, Eds. ACM, 2012, pp. 441–446.

[11] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, “Provable data possession at untrusted stores,” in Proceedings of the 14th ACM conference on Computer and communications security, ser. CCS ‟07. New York, NY, USA: ACM, 2007, pp. 598–609.

[12] A. Juels and B. S. Kaliski, Jr., “Pors: proofs of retrievability for large files,” in Proceedings of the 14th ACM conference on Computer and communications security, ser. CCS ‟07. New York, NY, USA: ACM, 2007, pp. 584–597.