SECURE SYSTEM WITH HIGHER RELIABILITY AND CONFIDENTIALITY USING DISTRIBUTED DEDUPLICATION OF MULTIMEDIA

[1]Aishwarya R,Anusha G Bedekar, Chandana R, [2]Dr. Malathy M
[1]Department of CSE,RRCE,Bangalore-74.
[2]Professor, Department of CSE,RRCE,Bangalore-74.
[1]ash.ramchandra@gmail.com,anushabedekar94@gmail.com,chandana.r13@gmail.com, [2]anandnmalathy@gmail.com

ABSTRACT-Data deduplication is for removing duplicate copies of data, and is used in cloud storage to reduce space and upload files which reduces bandwidth. There is only 1 copy for every file, image file, audio file or a video file kept in cloud although such a file is owned by an enormous range of users. Deduplication system improves storage utilization while reducing reliability. When data are outsourced by users to cloud, the challenge of privacy for data arises. To overcome the protection challenges, this paper commit to formalize the notion of distributed reliable deduplication system. we tend to propose new distributed deduplication systems with higher dependability within which the data parts are distributed across many different cloud servers. The data confidentiality and tag consistency are achieved by employing a deterministic secret sharing scheme in distributed storage systems, rather than victimization convergent encryption. Security checks verifies that our deduplication systems are secure in terms of the definitions.

Keywords—Deduplication, distributed storage system, reliability, secret sharing

.I. INTRODUCTION

Deduplication methods are mainly applied to standby data and reduce network and loading overhead by detecting and eliminating redundancy among data. Deduplication rejects redundant data by custody only one physical copy and revealing to any other copy , and by not maintaining multiple data copies with the same content. A number of deduplication systems have various deduplication such as client-side or server-side deduplications, file-level or block-level deduplications. With the wide-ranging usage of cloud storage, data deduplication techniques become more good-looking and critical for the management of very large amount of records in cloud storage services which motivates enterprises and administrations to farm out data storage as analyzed in many real-life case studies [1] According to the analysis report of IDC, the quantity of data within the world is predicted to succeed in forty trillion gigabytes in 2020. [2] Drop box, Google Drive and Mozy, are applying deduplication to avoid wasting the network bandwidth and therefore the storage price with client-side deduplication. Two kinds of deduplication are there in terms of the size: (i) file-level deduplication, that discovers redundancies between completely different files and removes these redundancies to scale back capacity demands, and (ii) block-level deduplication, that discovers and removes redundancies between information blocks. The file will be divided into smaller fixed or variable-size blocks. Fixed size blocks simplifies the computations of block boundaries, whereas using variable-size blocks (e.g., supported Rabin fingerprinting) [3] Provides better deduplication efficiency. It also reduces the reliability of the system. A very crucial issue in a deduplication storage system is data reliability because there's only 1 copy for every file hold on within the server shared by all the owners. If a shared File/chunk was lost, quite a large amount of data becomes inaccessible. this is often because of the inaccessibility of all the files that share this file/chunk. A essential downside is to ensure high data reliability in deduplication system. several of the previous deduplication systems have solely been considering only one server. Users and applications intend for higher reliability, particularly in archival storage systems wherever data are critical and will be preserved over lasting periods. Thus, the deduplication storage systems should offer reliability comparable to alternative systems.

I. II. RELATED WORKS

Gantz J & Raiusel D have projected, the "digital universe" — a measure of all the digital knowledge created, replicated, and consumed throughout one year. it's also a projection of the size of that universe to the tip of the last decade. The digital universe is made of pictures and videos on mobile phones uploaded to YouTube, digital cinemas immigrating the pixels of our high-definition televisions, banking knowledge swiped in an ATM, security footage at airports and major events just like the Olympic Games, subatomic collisions recorded the massive hadron collider at CERN, transponders recording road tolls, voice calls zipping through digital phone lines, and texting as a widespread implies that of communications. However, duplication among the file since cloud is used [2].

M. O. Rabin have projected, an information dispersal algorithmic program (IDA) is developed that breaks a file F of length L = (F (into n pieces F, 1 5 i 5 n, every of length (F, 1 = L/m, so each m pieces fulfil for reconstructing F. dispersal and reconstruction are computationally efficient. The total of the lengths (F, 1 is (n/m) . L. Since n/m are typically chosen to be close to I, the IDA is area efficient. IDA has varied applications to secure and reliable storage of data in computer networks and even on particular disks, to fault-tolerant and well-organized broadcast of data in webs, and for parallel processors to communicate with one

another. For the latter downside provably time-efficient and intensely fault-tolerant routing is achieved, using merely constant size buffers [9].

M. Bellare, S. Keelveedhi, and T. Ristenpart have projected, Cloud storage service providers like Dropbox, Mozy, et al. perform deduplication to avoid wasting space by solely storing one copy of every file uploaded ought to clients conventionally encrypt their files, however, savings are lost. Message-locked encryption (the most outstanding manifestation of that is convergent encryption) resolves this tension. However, it's inherently subject to brute-force attacks which will recover files falling into a notable set. we tend to propose an architecture that has secure deduplicated storage resisting brute-force attacks, and understand it in a system known as DupLESS. In DupLESS, clients encrypt beneath message-based keys via an oblivious PRF protocol. It allows clients to store encrypted data with an existing service and make them perform deduplication on their behalf, and nevertheless achieves robust confidentiality guarantees. We tend to show that encryption for deduplicated storage can achieve performance and space savings near that of using the storage service with plaintext data. it's an occasional performance thus doesn't match acceptable requirements fully [5].

M. Bellare has proposed new cryptographic encryption is formalized referred to as Message bolted encryption (MLE), where the key below which the encoding and cryptography is performed is itself derived from the message. MLE provides a replacement way to accomplish secure deduplication, a goal targeted by varied cloud storage providers. It provides definitions each for privacy and for a sort of integrity that is referred to as tag consistency. Supported this foundation each practical and theoretical contributions are created. Users would possibly wish their files to be encrypted and conventional encryption makes deduplication unfeasible [6].

J. Li, X. Chen, M. Li, J. Li, P. Lee, and W. Lou have proposed the basic idea of eliminating duplicate copies of storage data and limit the injury of purloined data if we've a tendency to decrease the value of that purloined data to the attacker. This paper attempts to address the matter of achieving efficient and reliable key management in secure deduplication. We have a tendency to initially introduce a baseline approach among which each {and every} user holds an independent master key for encrypting the convergent keys and outsourcing them. However, such a baseline key management scheme generates a large vary of keys with the increasing vary of users and wishes users to dedicatedly defend the master keys. To this end, we tend to propose Dekey, User Behavior profiling and Decoys technology. Dekey new constructions among those users haven‘t got to be compelled to manage any keys on their own but instead firmly distribute the convergent key shares across multiple servers for insider offender. As a symbol of idea, we tend to implement Dekey using the Ramp secret sharing scheme and demonstrate that Dekey incurs restricted overhead in realistic environments. User profiling and decoys, then, serve a pair of functions. Initial one is validating whether or not data access is permitted once abnormal data access is detected, and other is that confusing the attacker with phony data. We have a tendency to posit that the mixture of those security measures can offer unprecedented levels of security for the deduplication in insider and outsider offender. The varied styles of data for each user hold on among the cloud and so the demand of long-term continuous assurance of their data safety, the matter of supportive correctness of data storage among the cloud becomes even more durable. Cloud Computing is not merely a third party data warehouse. The data hold on among the cloud may even be frequently updated by the users, still as insertion, deletion, modification, appending, reordering, etc. One crucial challenge of today‘s cloud storage services is that the management of the ever-increasing volume of data. Every user ought to associate an encrypted convergent key with each block of its outsourced encrypted data copies, thus on later restore the data copies. Although completely different users may share identical data copies, they have to possess their own set of convergent keys thus no various users will access their files. Second, the baseline approach is unreliable, as a result of it wants each user to dedicatedly defend his own master key. If the master key is accidentally lost, then the user data can't be recovered; if it's compromised by attackers, then the user data are leaked [11].

M. W. Storer, K. Greenan, D. D. E. Long, and E. L. Miller have proposed - As the world moves to digital storage for depository functions, there is an increasing demand for systems which can provide secure data storage in an exceedingly cost-efficient manner. By distinctive common chunks of data every among and between files and storing them one time, deduplication can yield cost savings by increasing the utility of a given amount of storage. Sadly, deduplication exploits identical content, whereas encryption makes a trial to make all content appear random; identical content encrypted with a pair of completely different keys results in very different cipher text. Thus, combining the space efficiency of deduplication with the secrecy aspects of encryption is problematic. We've developed a solution that has each data security and space efficiency in single-server storage and distributed storage systems. Encryption keys are generated throughout a uniform manner from the chunk data; so, identical chunks will perpetually encrypt to an identical cipher text. Furthermore, the keys can't be deduced from the encrypted chunk data. Since the data each user should access and decrypt the chunks that compose a file is encrypted using a key known exclusively to the user, even a full compromise of the system cannot reveal that chunks are used by that users. It'd have security issues [20].

M. Li, C. Qin, P. P. C. Lee, and J. Li have proposed cloud-of-clouds storage exploits diversity of cloud storage vendors to provide fault tolerance and avoid vendor lock-ins. Its inherent diversity property in addition permits us to produce keyless data security via dispersal algorithms. However, the keyless security of existing dispersal algorithms depends on the embedded random data that breaks data deduplication of the distributed data. To simultaneously enable keyless security and deduplication, we've an inclination to propose a novel dispersal approach referred to as convergent dispersal that replaces original random data with deterministic cryptographic hash data that is derived from the initial data

but can't be inferred by attackers whereas not knowing the complete data. We've a tendency to develop a pair of convergent dispersal algorithms, specifically CRSSS and CAONT-RS. Our analysis shows that CRSSS and CAONT-RS provide complementary performance benefits for varied parameter settings [16].

II. III. PROPOSED SYSTEM

In this paper, we analyze how to develop a secure deduplication systems with higher reliability in cloud computing. We introduce the concept of distributed cloud storage servers into deduplication systems to provide well error acceptance. In addition to shield data secrecy, the concept of secret sharing method is utilized, which is also well-matched with the distributed storage systems. A file is first divided and encoded into fragments by using the technique of secret sharing. These parts will be distributed across multiple independent storage servers. Furthermore, to support deduplication, short cryptographic hashtag of the content will also be computed and sent to each storage server. Only the data owner who leading to uploads the data is essential to use and issue the fragments, while all succeeding users who own the similar file copy do not want to do the similar task. To recover data copies, users need access a least number of storage servers through certain safety checks similar to authentication and obtain the secret shares. In other words, the secret shares of data can be accessible only by the users who own the corresponding data copy after being authorized. Four new safe deduplication structures are used to care deduplication with high consistency for file-level and block-level deduplication, individually. [1]The secret splitting technique, instead of old-style encryption approaches, is exploited to keep data privacy. Data are split into parts by using secure secret sharing schemes and stored at different servers.

III. IV. ARCHITECTURE

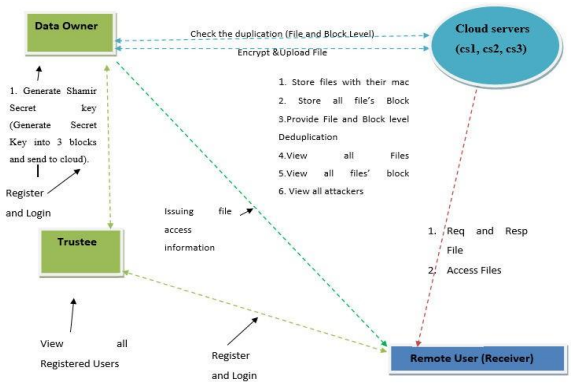

Fig 1:System Architecture

Data Owner

Here, the owner of the data uploads their data in the cloud server. [2]For the security purpose the data owner encrypts the file and then store in the cloud. The data owner can check the duplication of the file and multimedia file over Corresponding cloud server. The Data owner can have capable of manipulating the encrypted data file and the data owner can check the multiple cloud data as well as the duplication of the specific file. And also he can create remote user with respect to registered cloud servers. And also data owner has migrate to another cloud option, by this he can migrate files from one cloud server to another cloud server.

Trustee

In this module, the connector helps to check duplication of file existed or not in cloud server and you can check in multi cloud servers also. If it is existed, then also owner trying to upload the same file in same cloud server then connector automatically blocks his access permission. If it is not existed, then data owner can upload file in multi cloud servers at a time.

Cloud Server

[3]Data owners encrypt their data files and store them in the cloud for sharing with Remote User. [4]To access the shared data files, end users download encrypted data files of their interest from the cloud and then decrypt them.

Remote User

In this module, remote user logs in by entering his user name with the password. He will then request the cloud server for a secret key of the particular file. After getting secrete key he is trying to download file by entering file name and secrete key from cloud server.

Attacker Module

In remote user module, while downloading time if remote user entered any wrong file name or secrete key then cloud servers treats him as attacker and moves his access permission to block/attacker list.

V. IMPLEMENTATION

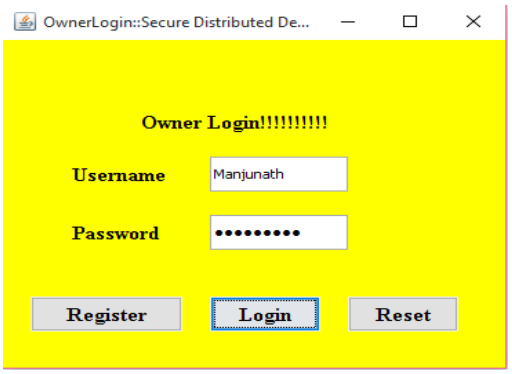

Fig 2: Owner login

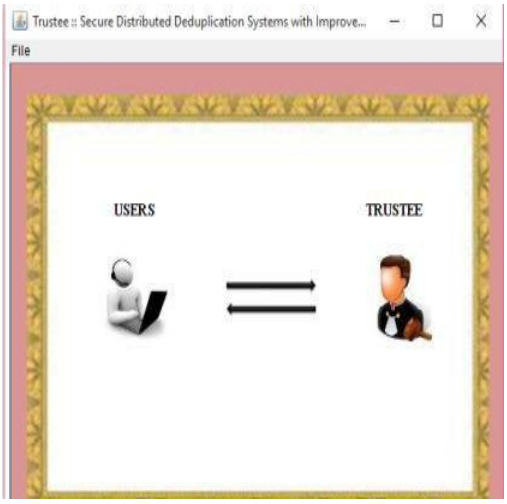

Fig 3: Trustee

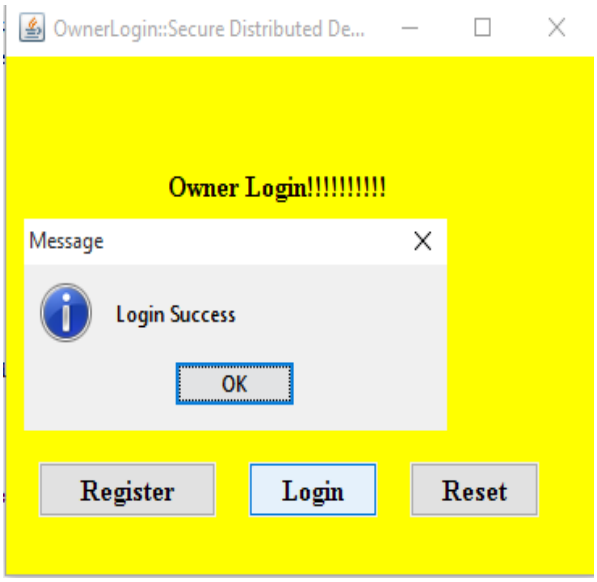

Fig 4: login successful

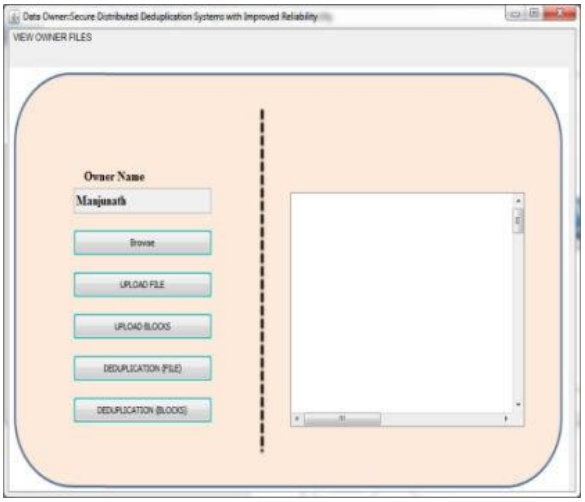

Fig 5:Data owner

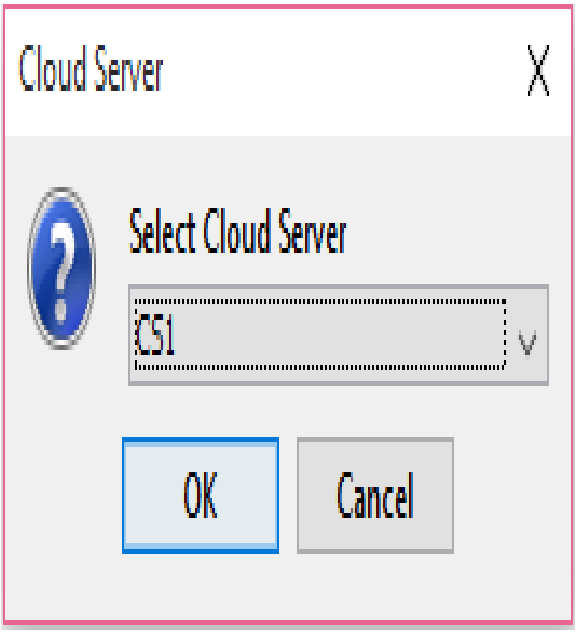

Fig 6: Select cloud server

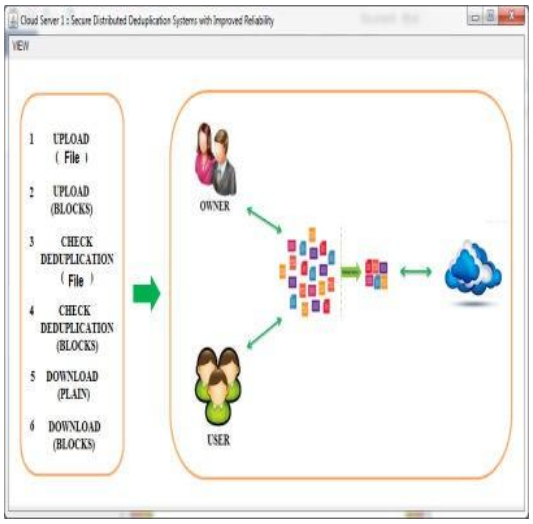

Fig 7:Cloud Server

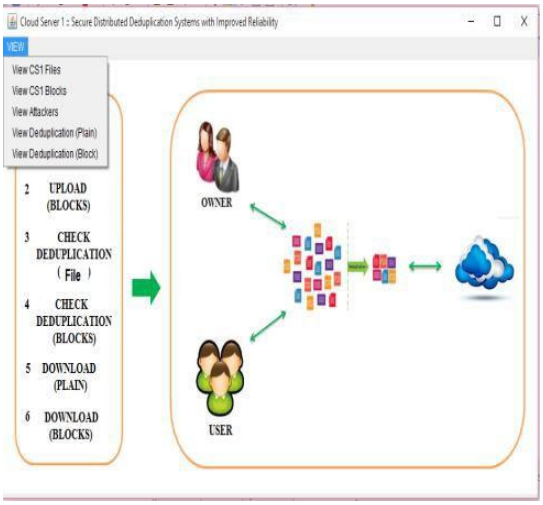

Fig 8: view files

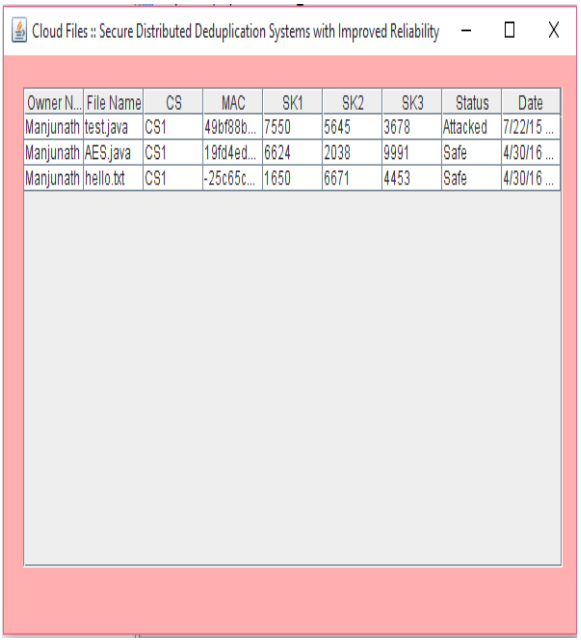

Fig 9: Files

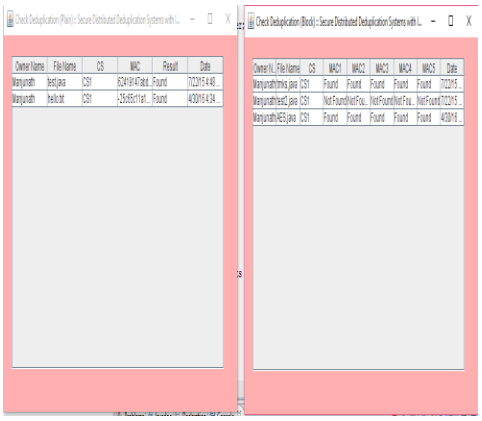

Fig 10: Check duplication

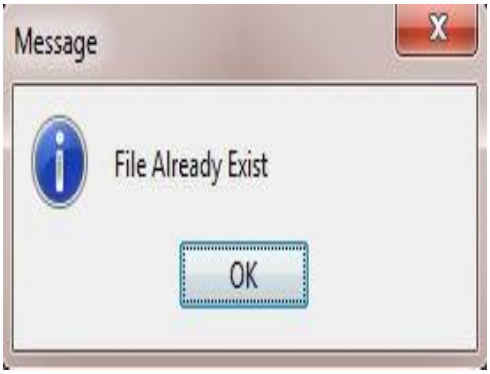

Fig 11: Pop up window

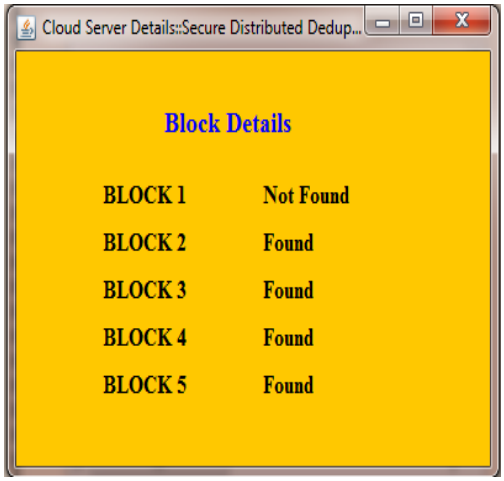

Fig 12: Block details

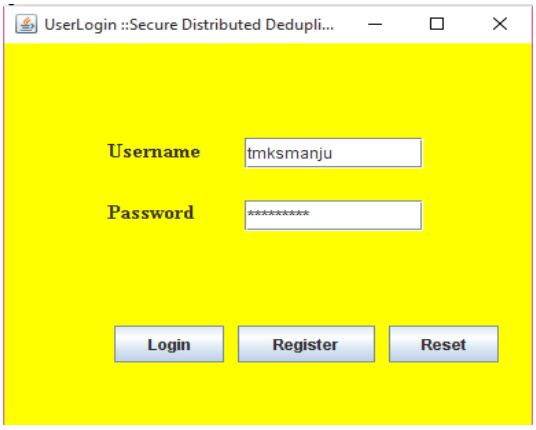

Fig 13: User login

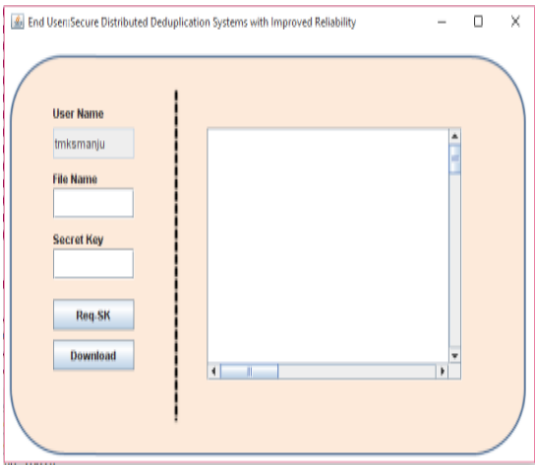

Fig 14: End user

Fig 15: secret key for download

VI. CONCLUSION

We proposed the distributed deduplication systems to improve the reliability of data while achieving the confidentiality of the users outsourced data without an encryption mechanism. Four constructions were proposed to support file-level and fine-grained block-level data deduplication. The security of tag consistency and integrity were achieved. We implemented our deduplication systems using the Ramp secret sharing scheme and demonstrated that it incurs small encoding/decoding overhead compared to the network transmission overhead in regular upload/download operations.

REFERENCES

[1] Amazon,–CaseStudies,‖ backup.
[2] J. Gantz and D. Reinsel, –The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east,‖ http://www.emc.com/collateral/analyst-reports/idcthe-
Digital-universe-in-2020.pdf, Dec 2012.
[3] M. O. Rabin, –Fingerprinting by random polynomials,‖ Center for Research in Computing Technology, Harvard University, Tech. Rep. Tech. Report TR-CSE-03-01, 1981.
[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*, 2002, pp. 617–624.

[5] M. Bellare, S. Keelveedhi, and T. Ristenpart, ―Dupless: Serveraided encryption for deduplicated storage,‖ in *USENIX Security Symposium*, 2013.

[6] M. Bellare,―Message-locked encryption and secure deduplication,‖ in EUROCRYPT, 2013, pp. 296–312.

[7] 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.

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

[9] M. O. Rabin, ―Efficient dispersal of information for security, load balancing, and fault tolerance,‖ *Journal of the ACM*, vol. 36, no. 2, pp. 335–348, Apr. 1989.

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

[11] 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*, 2014, pp. vol. 25(6), pp. 1615–1625.

[12] S. Halevi, D. Harnik, B. Pinkas, and A. Shulman-Peleg, ―Proofs of ownership in remote storage systems.‖ in *ACM Conference on Computer and Communications Security*, Y. Chen, G. Danezis, and V. Shmatikov, Eds. ACM, 2011, pp. 491–500.

[13]J. S. Plank, S. Simmerman, and C. D. Schuman, ―Jerasure: A library in C/C++ facilitating erasure coding for storage applications - Version 1.2,‖ University of Tennessee, Tech. Rep. CS-08-627, August 2008.

[14]J. S. Plank and L. Xu, ―Optimizing Cauchy Reed-solomon Codes for fault-tolerant network storage applications,‖ in *NCA-06: 5th IEEE International Symposium on Network Computing Applications*, Cambridge, MA, July 2006.

[15] C. Liu, Y. Gu, L. Sun, B. Yan, and D. Wang, ―R-admad: High reliability provision for large-scale de-duplication archival storage systems,‖ in *Proceedings of the 23rd international conference on Supercomputing*, pp. 370–379.

[16] M. Li, C. Qin, P. P. C. Lee, and J. Li, ―Convergent dispersal: Toward storage-efficient security in a cloud-of-clouds,‖ in *The 6th USENIX Workshop on Hot Topics in Storage and File Systems*, 2014.

[17] P. Anderson and L. Zhang, ―Fast and secure laptop backups with encrypted de-duplication,‖ in *Proc. of USENIX LISA*, 2010.

[18] Z. Wilcox-O'Hearn and B. Warner, ―Tahoe: the least-authority filesystem,‖ in *Proc. of ACM StorageSS*, 2008.

[19] 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.

[20] M. W. Storer, K. Greenan, D. D. E. Long, and E. L. Miller, ―Secure data deduplication,‖ in *Proc. of Storages*, 2008.