International Journal of Innovative Research in Science and Engineering
Vol. No.2, Issue 03, March 2016
www.ijirse.com
ISSN: 2454-9665

SMART DISTRIBUTED DEDUPLICATION SYSTEM WITH SECURITY AND RELIABLITY USING ALGORITHM

Sandhya Nikalje[1], Ratna Kumari[2], Swati Ingole[3]

[1,2,3]DYPIEMR , Akurdi B.E. Computer (Pune University)

ABSTRACT

Data is a technique for eliminating duplicate copies of data, and has been widely used in cloud storage to reduce storage space and upload bandwidth. However, there is only one copy for each file stored in cloud even if such a file is owned by a huge number of users. As a result, system improves storage utilization while reducing reliability. Furthermore, the challenge of privacy for sensitive data also arises when they are outsourced by users to cloud. Aiming to address the above security challenges, this paper makes the first attempt to formalize the notion of distributed reliable system. We propose new distributed systems with higher reliability in which the data chunks are distributed across multiple cloud servers. Small to large enterprises have been adopting this new technology as it gives significant Return on Investment by Reducing the storage capacity required to store the data and Reducing network bandwidth required to transfer the data.The storage device cost has been reducing with the advance in disk technology. On the other end, the IO bandwidth is on the upswing with Fibre Channel (FC) based networks, Gigabit Ethernet and iSCSI. IO throughput of these storage devices has increased and cost per GB of storing data has reduced. Data Deduplication does reduce the cost per GB of data even further but can be taxing on the storage device resources due to its mathematically extensive operations..

The distributed file system gives availability by replicating every file onto multiple nodes. Since this replication consumes symbolic storage space,it is important to rescue cast-off space where possible. Measurement of over 500 desktop file systems depicts that more half of all consumed space is occupied by duplicate files. We present a technique to reclaim space from this incidental replication to make it available for controlled file replication. Our mechanism includes 1) convergent encryption, which permits duplicate files to consolidate into the space of a single file, even if the files are encrypted with individual users' keys, and 2) SALAD, a Self- Arranging, Lossy, Associative Database for clustering file content and location information in a demoralized, scalable, fault-tolerant manner. Large-scale simulation experiments show that the duplicate-file converging system is scalable, highly effective, and fault-tolerant. Cloud depository service providers such as Dropbox, Mozy, and others perform deduplication to save space by only storing one copy of each file uploaded. Should clients commonly encrypt their files, however, savings are lost. Message-locked encryption (the most noteworthy manifestation of which is convergent encryption) resolves this tension. However it is intrinsicly subject to brute-force attacks that can recapture files falling into a known set. We show that encryption for deduplicated depository can achieve high potential and space economy close to that of using the storage service with plaintext data.

ISSN: 2454-9665

Keywords: Basic Algorithm, Convergent Encryption, Deduplication, Manifestation, Replicate, Hybrid Cloud , Proof of Ownership, SALAD

I. INTRODUCTION

With the explosive growth of digital data, deduplication techniques are widely employed to backup data and minimize network and storage overhead by detecting and eliminating redundancy among data. 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 has received much attention from both academia and industry because it can greatly improves storage utilization and save storage space, especially for the applications with high deduplication ratio such as archival storage systems. A number of deduplication systems have been proposed based on various deduplication strategies such as client-side or server-side deduplications, file-level or block-level deduplications. Especially, with the advent of cloud storage, data deduplication techniques become more attractive and critical for the management of ever-increasing volumes of data in cloud storage services which motivates enterprises and organizations to outsource data storage. Irrespective of vendor implementation, Data Deduplication can be divided into four steps of data segmentation, creating fingerprints for these data segments, matching these finger prints for duplicates and storing unique segments on disk.Our objectives are performing file and block level deduplication with improved reliability.

II. MOTIVATION

There is a huge amount of duplicate or redundant data existing in current storage systems, especially backup systems. There are numerous techniques for reducing redundancy when data is stored or sent. Recently, data deduplication, the new emerging technology motivated by this scenario of data expansion, has received a broad attention from both industries and academics. Data deduplication, also called intelligent compression or single instance storage, eliminates redundancy caused by identical objects which can be detected efficiently by comparing hashes of the objects content. Storage space requirements can be reduced by a factor of 10 to 20 or even 90 when backup data is de-duplicated. Client based deduplication processing can reduce the amount of data being transferred over the network to thestorage system, but there are often requirements for extra CPU and disk I/O processing on the client side. Clients might be constrained to file level visibility of the data, which can reduce the granularity of the deduplicated component analysis. Also, clients might only have visibility to a limited pool of data, which can impact the duplicate frequency of the pool and reduce deduplication efficiency.

III. COMMONLY USED STEPS

There are more than a dozen major vendors for Deduplication applications. Irrespective of vendor implementation Data Deduplication can be categorized into four major steps:

1. Identifying the unit of comparison.
2. Creating smaller unique identifier of these units to be compared.
3. Match for duplicates.

International Journal of Innovative Research in Science and Engineering
Vol. No.2, Issue 03, March 2016
www.ijirse.com
ISSN: 2454-9665

4. Saving unique data blocks.

3.1 IDENTIFYING THE UNIT OF COMPARISON

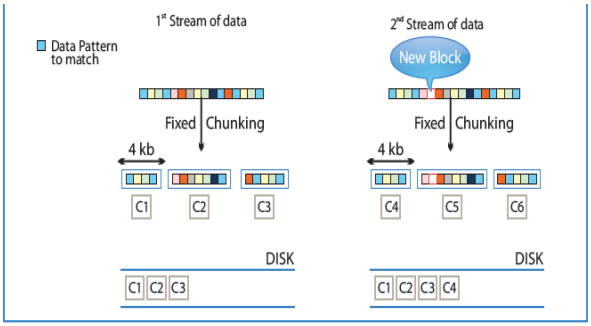


Figure 3: Variable Chunking

Deduplication needs to first identify the unit of data to work upon to find if the unit already exists on the disk. Deduplication can be either content-aware (knows about file boundaries and works on files) or can work at block levels, the latter being more popular and widely used methodology.

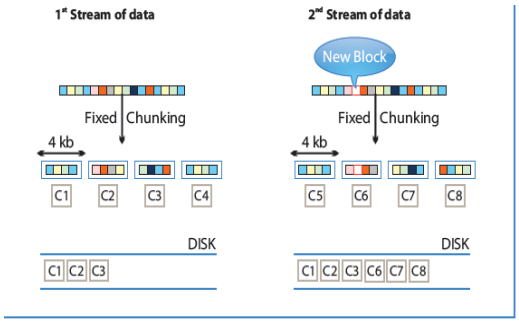


Figure 2: Fixed Chunking Drawback

3.2 CREATING SMALLER UNIQUE IDENTIFIER OF THESE UNITS TO BE COMPARED

When the unit of comparison is a file, matching is done using binary level comparison and/or hash addressing. However, data chunks need to be indexed for the purpose of matching if the data chunk already exists on disk. Since the chunks sizes are in Kilobytes, comparing each chunk with the chunk on disk is a near impossible task. Hence, the Data Deduplication application creates a unique identifier for each of the chunks which are exponential small in size as against the chunk size.

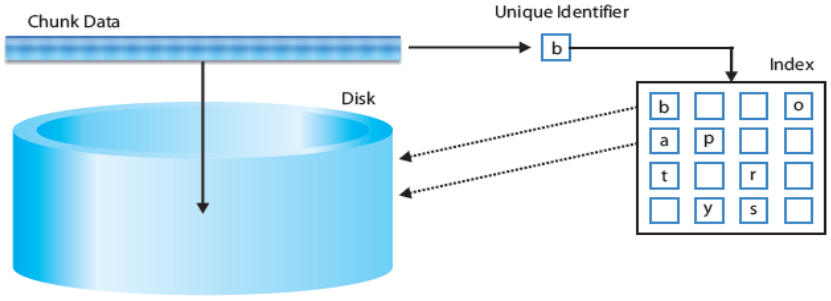


Figure 4: Creating Unique Identifier and Indexing

This can be achieved by hashing. Hashing creates a significantly smaller representation of a large data. Some of the popular hashing algorithms used are Secure Hash Algorithm (SHA1, SHA512, SHA256), Message Digest Algorithm (MD4, MD5) and so on.

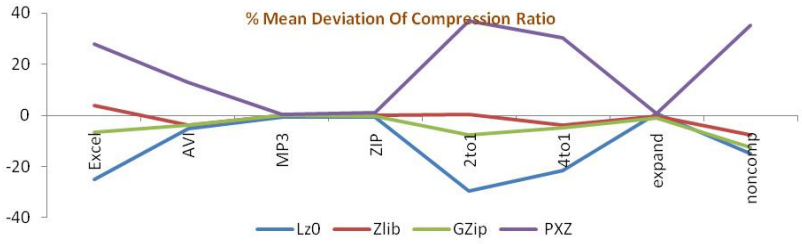
Hash algorithm	Size of Digest (bits)	Number of chunks to be generated for 50% or greater probability* of a hash collision
MD5	128	2^{64}
SHA1	160	2^{80}
SHA256	256	2^{128}
SHA512	512	2^{256}
Whirlpool	512	2^{256}

3.3 MATCH FOR DUPLICATES

Matching for duplicates becomes the crux of the Data Deduplication application and has the maximum impact on both dedupe ratio and throughput. As the data processed by Data Deduplication application increases, the size of the index table also increases and searching the index table becomes slower by time.

3.4 SAVING UNIQUE DATA BLOCKS

Saving of unmatched data on to the disk would also have a bearing on the dedupe ratio and throughput**.** There are a few open source and commercially available compression libraries which can be used. Note the following before using the compression library:



IV. EXISTING SYSTEM

There are many de-duplication schemes proposed by the research community. The reliability in de-duplication has also been addresse.. However, they only focused on traditional files without encryption, without considering the reliable de-duplication over ciphertext. Convergent encryption ensures data privacy in de-duplication. Bellare et al. formalized this primitive as message-locked encryption, and explored its application in spaceefficient secure outsourced storage. Another two-layered encryption scheme with stronger security while supporting de-duplication was proposed for unpopular data. In this way, they achieved better tradeoff between the efficiency and security of the outsourced data.

1. **ISSUES WITH EXISTING SYSTEM**

- Large number of users may upload same redundant data to the cloud.
- Enormous number of copies are created if the data is same.
- Block level deduplication can not performed.
- Data chunks are not distributed across multiple cloud servers.

International Journal of Innovative Research in Science and Engineering
Vol. No.2, Issue 03, March 2016
www.ijirse.com
ISSN: 2454-9665

V. PROPOSED SYSTEM

New secure de-duplication systems are proposed to provide efficient de-duplication with high reliability for file-level and block-level de-duplication, respectively. The secret splitting technique, instead of traditional encryption methods, is utilized to protect data confidentiality. Specifically, data are split into fragments by using secure secret sharing schemes and stored at different servers. Confidentiality, reliability and integrity can be achieved in our proposed system. Two kinds of collusion attacks are considered in our solutions. These are the collusion attack on the data and the collusion attack against servers. In particular, the data remains secure even if the adversary controls a limited number of storage servers. Systems are using the Ramp secret sharing scheme that enables high reliability and confidentiality.

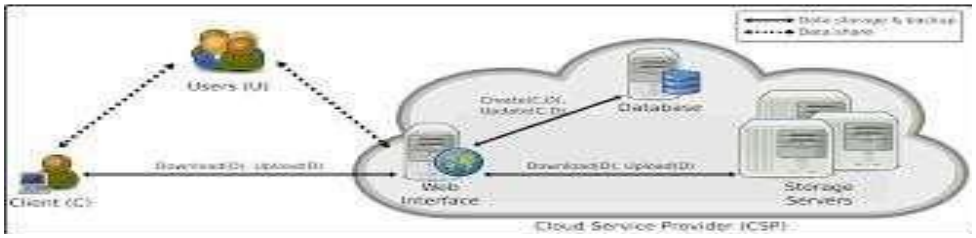


Figure : Proposed System

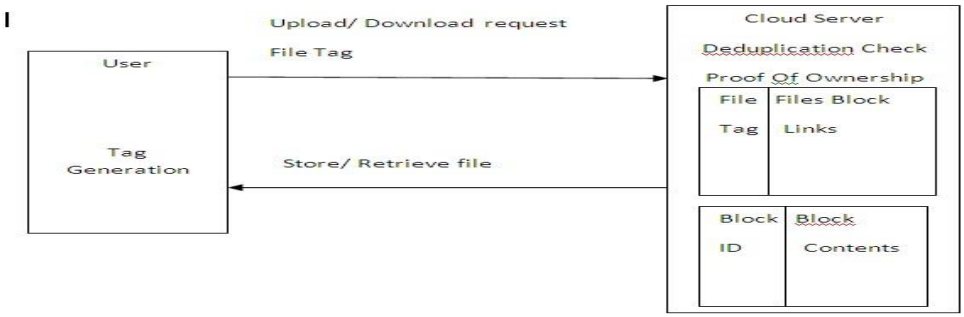


Figure : System Architecture

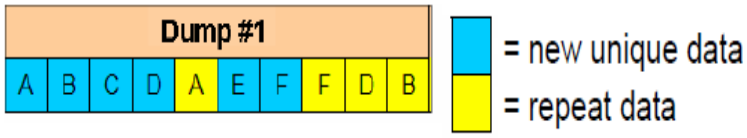
VI. METHODOLOGIES USED

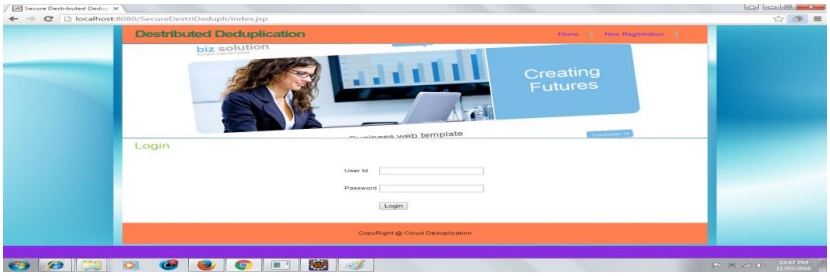
The general procedural methods which are being carried out are as follows :

- **MODULE 1 : Data de-duplication**

Data deduplication involves finding and removing duplication within data without compromising its fidelity or integrity. The goal is to store more data in less space by segmenting files into small variable-sized chunks (32–128 KB), identifying duplicate chunks, and maintaining a single copy of each chunk. Redundant copies of the chunk are replaced by a reference to the single copy. The chunks are compressed and then organized into special container files in the System Volume Information folder. After deduplication, files are no longer stored as independent streams of data, and they are replaced with stubs that point to data blocks that are stored within a common chunk store.

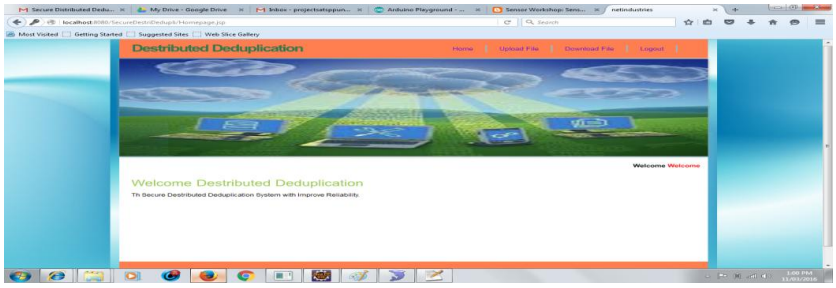
International Journal of Innovative Research in Science and Engineering
Vol. No.2, Issue 03, March 2016
www.ijirse.com
ISSN: 2454-9665





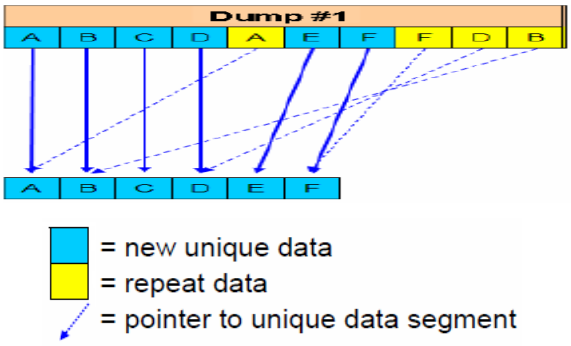
- **MODULE 2 : Uploading a file**

This may include the process of uploading the data to the server by the user. This maybe the unique content for the upload for the first time.



- **MODULE 3 : Distributed data de-duplication**

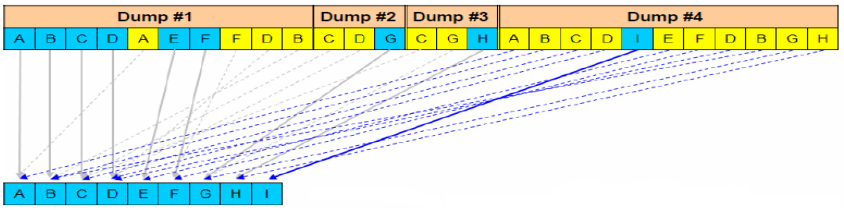
The distributed systems proposed aim is to reliably store data in the cloud while achieving confidentiality and integrity. Its main goal is to enable and distributed storage of the data across multiple storage servers. Instead of encrypting the data to keep the confidentiality of the data, our new constructions utilize the secret splitting technique to split data into shards. These shards will then be distributed across multiple storage servers. Building Blocks Secret Sharing Scheme. There are two algorithms in a secret sharing scheme, which are Share and Recover.

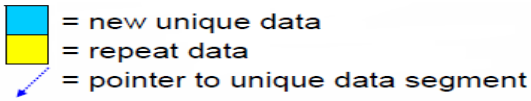


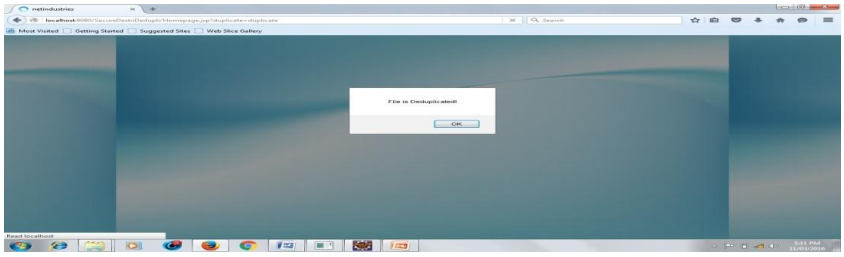
- **MODULE 4 : File level de-duplication**

International Journal of Innovative Research in Science and Engineering
Vol. No.2, Issue 03, March 2016
www.ijirse.com
ISSN: 2454-9665

To support efficient duplicate check, tags for each file will be computed and are sent to S-CSPs. To prevent a collusion attack launched by the S-CSPs, the tags stored at different storage servers are computationally independent and different.







- **MODULE 5 : Block level de-duplication**

We show how to achieve the fine-grained block-level distributed de-duplication. In a block-level de-duplication system, the user also needs to firstly perform the file-level de-duplication before uploading his file. If no duplicate is found, the user divides this file into blocks and performs block-level de-duplication.

Along this we are able to showcase the role of each actors and the actions taking place in it.

ISSN: 2454-9665

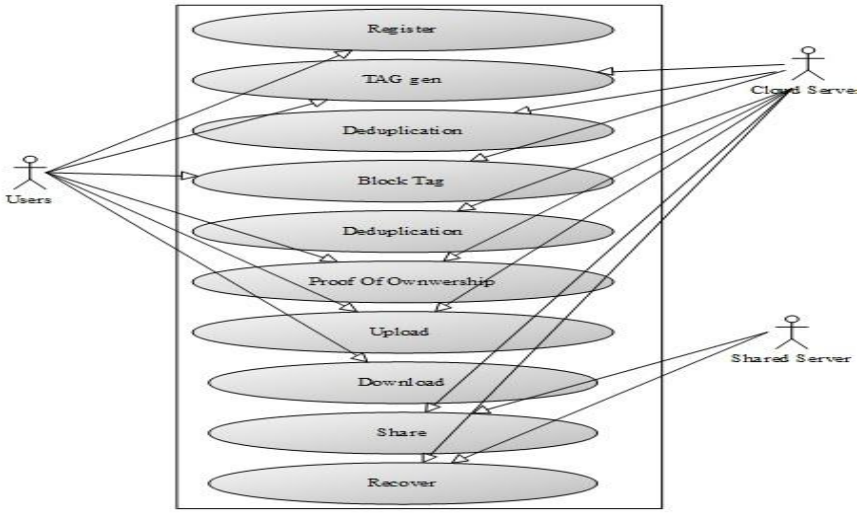


Figure : Use Case Diagram

- **MODULE 6 : Downloading a file**

This is the end process where user can view or download their file after the process of deduplication.

VII. POSSIBLE OUTCOMES

We have proposed and implemented the paper and hence we have come up with some sort of possible outcomes and the results for which we are sure and optimistic that the steps taken towards the implementation are correct and unbiased.

VIII. RESULTS

Storage administrators are struggling to handle spiraling volumes of documents, audio, video, images and large email attachments. Adding storage is not always the best solution, and many companies are turning to data reduction technologies such as data deduplication. This article explains the basic principles of data deduplication, and looks at implementation issues.

IX. ADVANTAGES

- Reduced Storage Allocation
- Efficient Volume Replication
- Effective increase in the banwidth
- Fast recovery
- Cost efficient

X. APPLICATIONS

- In back up systems.
- Cloud Service Providers
- Inline Processing

ISSN: 2454-9665

- Target based deduplication
- Deduplication with tapes and disks.
- Deduping the primary storages.
- Records retention and E-mailing

XI. FUTURE SCOPE

ClouDedup may be extended with more security features such as proofs of retrievability, data integrity checking and search over encrypted data. In this paper we mainly focused on the definition of the two most important operations in cloud storage, that are storage and retrieval. We plan to define other typical operations such as edit and delete. After implementing a prototype of the system, we aim to provide a full performance analysis. Furthermore, we will work on finding possible optimizations in terms of bandwidth, storage space and computation.

XII. CONCLUSION

We proposed the distributed de-duplication 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 de-duplication. The security of tag consistency and integrity were achieved.We implemented our de-duplication 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.

ACKNOWLEDGEMENT

We would like to express our sincere thanks to our guide, Prof. Nareshkumar Mustary, Assistant Professor at the Department of Computer, DYPIEMR , Akurdi and our beloved Mam Prof. P. P. Shevatekar HOD of Computer Dept, for the support and motivation while carrying out the work presented here and also in writing this paper. We thanks our principal Dr. A. V. Patil for providing the necessary environment and facilities to conduct our project work.

REFERNCES

[1] Amazon, "Case Studies," https://aws.amazon.com/solutions/casestudies/# backup.

[2] J. Gantz and D. Reinsel, "The digital universe in 2020: Big data, bigger digi tal 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.

ISSN: 2454-9665

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

[6] "Message-locked encryption and secure de-duplication," 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," Commun. ACM, vol. 22, no. 11, pp. 612–613, 1979.

[11] 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.