Volume 6, Issue 5, May 2016 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Cryptographic Tuning to Minimize Storage Requirement on Cloud Using De-Duplication Mechanism

Naziya Tabassum*
M. Tech Student, Dept. of CSE
Tulsiramji Gaikwad-Patil College of Engineering
and Technology Nagpur, India

Prof. Roshani B. Talmale
Professor, Dept. of CSE
Tulsiramji Gaikwad-Patil College of Engineering
and Technology Nagpur, India

Abstract— ***Cloud computing provide several attractive benefits for business, enterprises and end users. The benefits of cloud computing includes self-services provision that means the end user can use the resources for almost any type of workload on-demand. As its flexibility of unlimited data storage and its access worldwide, it comes with a price. Cloud services provider provides both highly availability and parallel computing resources at low cost. These resources can be dynamically configured to allow optimum resources utilization. As cloud computing grows the amount of data increases day by day and shred by user with specified privileges. So it is most importance to first minimize the storage cost and eliminate the repetitive data. Data de-duplication is one of the data compression techniques that can be used to eliminate the repetitive data and can be used in cloud computing architecture. For privacy and security point of view convergent technique has been used to encrypt data before outsourcing. But previous systems have the limitation of convergent encryption and in proposed system applied the techniques of cryptographic tuning to make the encryption more secure and flexible. Data de-duplication prohibits the storage of repetitive blocks and implements the pointer concept which basically puts the pointer to the existing blocks. Access control is provided into the application which allows the data owner the freedom of selecting users to have access to the published file. The integrity of data outsourced to the cloud is managed by the hash calculation of any content following the proof-of-ownership module. Proposed system calculate hash of the content on source and destination side and request the hash for the cloud side to predict the tampering of data. The expected analysis shows the improvement in execution time and development cost.***

Keywords—De-Duplication, Authorized duplicate check, public Cloud, Convergent Encryption, Cryptographic tuning.

I. INTRODUCTION

Cloud computing has been the most widely spread and recent technology that is being used in corporate model. As its flexibility of unlimited data storage and its access worldwide, it comes with a price. The cloud computing runs over “Pay As You Go” model and bills the storage cost to the company. Cloud services provider provides both highly availability and parallel computing resources at low cost. These resources can be dynamically configured to allow optimum resources utilization. As cloud computing grows the amount of data increases day by day and shred by user with specified privileges.

So it is most importance to first minimize the storage cost and eliminate the repetitive data. During the cost minimization, you have to also care about privacy about data and you cannot ignore the fact. As the amount of data increased the management of the increased data is difficult. To make data manageable in cloud computing a well-known technique de-duplication and has attracted more attention recently.

One of the biggest challenges to the data storage community is how to effectively store data without taking the exact same data and storing again and again in different locations on the same servers, hard drives, tape, libraries etc. there have been many attempts to address these redundancies some more successful than others. There has been an attitude in the data storage community that as we saw significant price reductions the cost of many data storage options that data storage savings was an exercise whose time had passed. With the regulatory environment becoming more stringent, the volume of saved data again began to explode and more and more options began to be considered to address data storage concerns.

II. LITERATURE REVIEW

Secure De-Duplication: In cloud computing security of data de-duplication has attracted much attention from research. Jin Li in [1] present data compression techniques for eliminating 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 confidentiality of sensitive data while supporting de-duplication, the convergent encryption technique has been proposed to encrypt the data before outsourcing. To better protect data security, this paper makes the first attempt to formally address the problem of authorized data de-duplication. Different from traditional de-duplication systems, the differential privileges of users are further considered in duplicate check besides the data itself.

A.Rahumed in [12] presents a secure cloud backup system that provides a secure layer for today's cloud storage service. It helps in eliminating of redundant data which is stored in backup. Nesrin in [4] discuss the trend of leveraging cloud-based services for large scale content storage, processing, and distribution. Security and privacy are among top concerns for the public cloud environments. Towards these security challenges, this paper propose and implement, on OpenStack Swift, a new client-side de-duplication scheme for securely storing and sharing outsourced data via the public cloud.

Yuan in [10] proposed a de-duplication system which is used to reduce the storage size of the tags for integrity check. This works to enhance the security of de-duplication and provide confidentiality. Bellar in [3] it transforms predictable message to unpredictable message to show how to protect data from unauthorized access. J. Li in [2] propose the technique to eliminate the redundant data instead of taking number of copies of same file. It also provide different encryption scheme that provide different security to popular and unpopular data. Another two-layered encryption scheme for high security with support of de-duplication is support for un popular data.

Convergent Encryption: In convergent encryption R. D in [8] it offers keyless data security via dispersal algorithm. The algorithm uses the embedded random algorithm which breaks the data duplication in dispersal data. And ensure privacy in de-duplication.

Chun ho [12] introduce Scaling up the backup storage for an ever-increasing volume of virtual machine (VM) images is a critical issue in virtualization environments. While de-duplication is known to effectively eliminate duplicates for VM image storage, it also introduces fragmentation that will degrade read performance. It propose RevDedup, a de-duplication system that optimizes reads to latest VM image backups using an idea called reverse de-duplication. In contrast with conventional de-duplication that removes duplicates from new data, RevDedup removes duplicates from old data, thereby shifting fragmentation to old data while keeping the layout of new data as sequen-tial as possible. It evaluate the RevDedup prototype using microbenchmark and real-world workloads. For a 12-week span of real-world VM images from 160 users, RevDedup achieves high de-duplication efficiency with around 97% of saving, and high backup and read throughput on the order of 1GB/s. RevDedup also incurs small metadata overhead in backup/read operations. Bellare in [5] proposed new message locked encryption scheme which is useful in space-efficient secure outsourced storage.

Mihire in [3] formalize a new cryptographic primitive that they call Message-Locked Encryption (MLE), where the key under which encryption and decryption are performed is itself derived from the message. MLE provides a way to achieve secure de-duplication (space-efficient secure outsourced storage), a goal currently targeted by numerous cloud storage providers.

M. Bellare in [5] propose an architecture that provides secure de-duplicated storage resisting brute-force attacks, and realize it in a system called DupLESS. In DupLESS, clients encrypt under message-based keys obtained from a key-server via an oblivious PRF protocol. It enables clients to store encrypted data with an existing service, have the service perform de-duplication on their behalf, and yet achieves strong confidentiality guarantees. It show that encryption for de-duplicated storage can achieve performance and space savings close to that of using the storage service with plaintext data.Xu.in [4] give a secured convergent encryption for encryption without considering issues of key-management and block-level de-duplication.

Proof of Ownership: Some of the authors work on protocols proof of ownership for de-duplication system. K. Zhang in [14] they proposed two techniques proof of retrieve ability and proof of data possession to assure data integrity for cloud storage. And by using proof of ownership improves storage efficiency by securely removing unnecessary data which is stored on the cloud server. Proof of retrieve ability and proof of data possession are used in order to achieve both data integrity and storage efficiency which is the objective of proof of ownership.

Halevi in [11] also use the concept of proof of ownership for de-duplication system, such that the client can prove to the cloud storage server that the client owns the file without uploading the file itself. One interesting concept in above techniques is that no one consider data privacy. Henry C in [9] present FadeVersion, a secure cloud backup system that serves as a security layer on top of today's cloud storage services. FadeVersion follows the standard version-controlled backup design, which eliminates the storage of redundant data across different versions of backups. On top of this, FadeVersion applies cryptographic protection to data backups. Specifically, it enables fine-grained assured deletion, that is, cloud clients can assuredly delete particular backup versions or files on the cloud and make them permanently inaccessible to anyone, while other versions that share the common data of the deleted versions or files will remain unaffected. It implement a proof-of-concept prototype of FadeVersion and conduct empirical evaluation atop Amazon S3. It show that FadeVersion only adds minimal performance overhead over a traditional cloud backup service that does not support assured deletion. Then Ng in [15] extends proof of ownership for encrypted files for security but do not address how to minimize the key management.

Jiawei in [10] support efficient and secure data integrity auditing with storage de-duplication for cloud storage. In this paper it solve this open problem with a novel scheme based on techniques including polynomial-based authentication tags and homomorphic linear authenticators. Our design allows de-duplication of both files and their corresponding authentication tags. Data integrity auditing and storage de-duplication are achieved simultaneously. Our proposed scheme is also characterized by constant real time communication and computational cost on the user side. Public auditing and batch auditing are both supported.

Cloud: In some of the papers authors discuss about the twin clouds architecture. Bugiel in [7] provide and environment in which it consist of twin clouds for secure outsourcing of data. Zhang in [14] proposed the hybrid cloud technique to support more privacy then other proposed systems. The security model of our proposed model is similar to this related work, in related work they use the concept of private cloud but in our proposed system we use only public cloud.

III. PROPOSED METHODOLOGY

Existing system support the concept of convergent encryption on hybrid cloud. The basic form of convergent encryption is taking your original file and calculating a hash from it. Then using this hash as the key, you encrypt the rest of the file. Finally using your password, you encrypt this hash key and store it somewhere else. This means the only way to get the password of the file is by owning the original file. If bill and ted both have the same file, they will both calculate the same hash, and the same encrypted file. Therefore since the encrypted version is the same, you only need one copy. To decrypt the file, you first decrypt the hash using your password, then the decrypt the file using the hash.

Convergent encryption is insecure, since its short encryption key is generated from the input file in a deterministic way and could be leaked. Roughly speaking, convergent encryption is as insecure as “hash-as- a-proof” method (i. e using hash value hash (F) as a proof of ownership of file F), in the presence of leakage. There- fore, all existing works on applying convergent encryption method to implement de-duplication of encrypted data are insecure in the bounded leakage setting of Proof of ownership.

In proposed system we make use of public cloud. A public cloud is based on the standard computing model. In this service providers’ makes use of resources, like storage and applications, these applications and storage are available to the general public over the internet. Public cloud service are offered free or offered of a pay per usage model. Public cloud services are easy and inexpensive setup. There is no wastage of resources because you pay for what you use.

The proposed system proposed Cryptographic tuning which provides many benefits over the previous system, previous system makes use of convergent encryption which is insecure, since its short encryption key is generated from the input file in a deterministic way and could be leaked, roughly speaking convergent encryption is as insecure as “hash-as- a-proof” method (i.e. using hash value hash as a proof of ownership of file in the presence of leakage).

Therefore, all existing works on applying convergent encryption method to implement de-duplication of encrypted data are insecure in the bounded leakage setting of proof of ownership. In previous de-duplication system not support different authorization de-duplication check. This is important in many applications. In this authorized de-duplication check user issued a set of privileges during the system initialization. To solve the problem of de-duplication with different privileges in cloud system the proposed system consider cloud system, data owners outsource their data by utilizing public cloud and the data operations are managed in the same cloud. The de-duplication system support different de-duplication check in proposed under public cloud.

The proposed system is presented for carrying out secured authorized de-duplication process. It has many significant features. To increase the amount of information that can be stored on cloud by saving bandwidth and to eliminate duplicate copies of redundant data to preserve confidentiality of sensitive data while supporting de-duplication.

The proposed system use the concept of cryptographic tuning in this the system calculates two hash value say H1 and H2. For calculating H1, system use HMAC-SHA512 algorithm used with a specially-selected HMAC key not used for any other purpose. This hash is called the 'key'. The data is encrypted with the key (using any symmetric encryption function such as AES-CBC). The encrypted data is then hashed. For H2, system complex the second algorithm instead of applying any standard algorithm this is our proposed system. We are taking the base of MD5 algorithm and altering it. The MD5 message –digest algorithm is a widely used cryptographic hash function producing a 128-bit hash value, typically expressed in text format as a 32 digit hexadecimal number.

MD5 has been utilized in a wide variety of cryptographic applications, and is also commonly used to verify data integrity. As we are saying altering it means we are design the new algorithm by taking basic concept of MD5 and making changes in the logic behind it. We are taking alternate variables of the generated data and taking it as a hash value for the encryption.

Cryptographic tuning

In cryptographic tuning it make use of two hash algorithms or first standard algorithm is used SHA after that for second hash value customized MD5 is used.

- Cryptographic tuning:
- Upload file
- Get hash1(file) gives K1
- Use this K1 to encrypt that file using convergent encryption
- Again calculate hash of encrypted file hash2(file’) gives h2
- Use custom hashing function in this phase. We have done reverse of md5.

IV. RESULT ANALYSIS

Result analysis

Below shows the different comparisons through which we can conclude that how proposed system is better than existing system. Table show how much faster the proposed system is.

Table 7.1: Execution time comparison as per result

Approach	Execution Time (ms)	File Size(Kb)
Base Paper	78	156
Our Approach	48	156

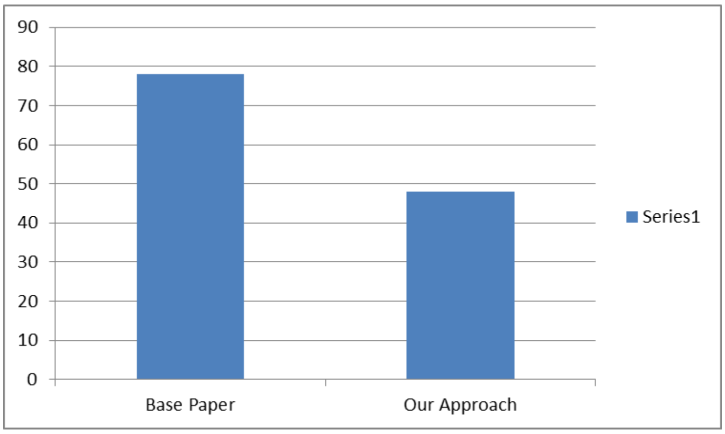

Fig 7.1 Execution time comparison

Table show the throughput time of base paper and proposed paper which shows the better through put and less execution time.

Table 7.2: Throughput time comparison as per result

Approach	Throughput	Execution Time (ms)	Execution Time (s)	File Size(Kb)
Base Paper	0.015	100	0.1	200
Our Approach	0.027	57	0.057	200

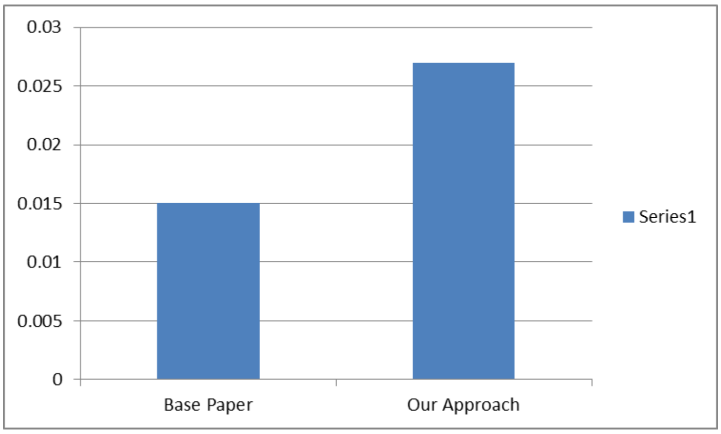

Fig 7.2: Throughput time comparison

Here the table and graph show the result analysis of the paper.

Table 7.3: Result analysis

File Size	156	200
Execution Time	78	89

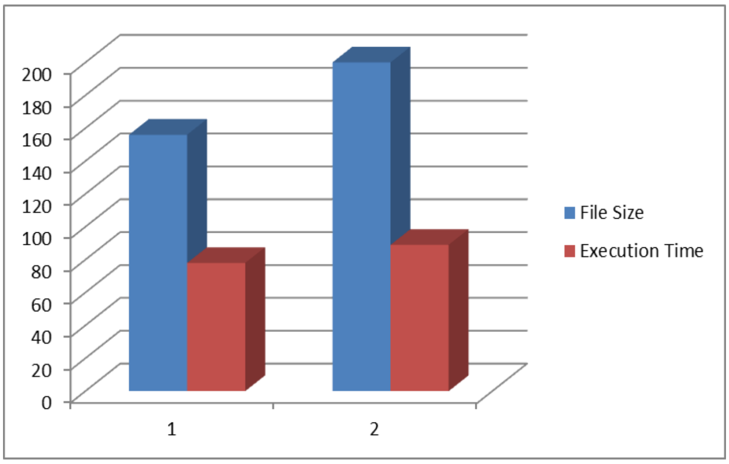

Fig 7.3: Result analysis of project

In below table the detailed shows the complexity of file its size and total time require to work on a particular file.

Table 7.4: Time complexity

File Name	Upload File	Read (Encrypt)	Decrypt	Download	File Size
brides19full.csv	247 ms	48 ms	26 ms	75 ms	156 KB
ON LOAD POP.txt	81 ms	29 ms	22 ms	67 ms	200 KB
data.txt	83 ms	19 ms	17 ms	26 ms	100 KB

V. CONCLUSION AND FUTURE WORK

Conclusion:

The notion of authorized data de-duplication was proposed to protect the data security by including differential privileges of users in the duplicate check. We also presented new de-duplication check to supporting authorized duplicate-check tokens of files that are generated by the cloud server with private keys. The proposed system is secure from insider and outsider attacks. The proposed system implement new prototype model in which we used convergent encryption with modification version to deal with brute force attack using Cryptographic tuning to make better authorized de-duplication technique.

Future work:

Block level encryption: Here block level encryption means the file is divided into blocks and then encryption is perform on it and here the comparison done in the manner it compare the file on the cloud block wise the block which is different it replace only that block not the full file.

Can be used on hybrid cloud: This algorithm and method which are using in this project can be used on hybrid cloud. Hybrid cloud comes in domain separation where we take two clouds public cloud and private cloud. As this project using this on public cloud the same things can be done on hybrid cloud as well in which the keys are put on private cloud and the operational data can be placed on the public cloud, which will provide more security then the present system.

ACKNOWLEDGMENT

With profound feeling of immense gratitude and affection, I would like to thank my guide Prof. R. B. Talmale, Department of Computer Science & Engineering for her continuous support, motivation, enthusiasm and guidance. Her encouragement, supervision with constructive criticism and confidence enabled me to complete this paper.

REFERENCES

[1] Li, Yan Kit Li, Xiaofeng Chen, Patrick P. C. Lee, Wenjing Lou. A Hybrid Cloud Approach for Secure Authorized De-duplication IEEE transaction VOL:PP NO:99 2014.

[2] J. Li, X. Chen, M. Li, J. Li, P. Lee, andW. Lou. Secure de-duplication with efficient and reliable convergent key management. In IEEE Transactions on Parallel and Distributed Systems, 2013.

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

[4] J. Xu, E.-C. Chang, and J. Zhou. Weak leakage-resilient client-side de-duplication of encrypted data in cloud storage. In ASIACCS, pages 195–206, 2013.

[5] M. Bellare, S. Keelveedhi, and T. Ristenpart. Dupless: Serveraided encryption for de-duplicated storage. In USENIX Security Symposium, 2013.

[6] P. Anderson and L. Zhang. Fast and secure laptop backups with encrypted de-duplication. In Proc. of USENIX LISA, 2010.

[7] S. Bugiel, S. Nurnberger, A. Sadeghi, and T. Schneider. Twin clouds: An architecture for secure cloud computing. In Workshop on Cryptography and Security in Clouds (WCSC 2011), 2011.

[8] R. D. Pietro and A. Sorniotti. Boosting efficiency and security in proof of ownership for de-duplication. In H. Y. Youm and Y. Won, editors, ACM Symposium on Information, Computer and Communications Security, pages 81–82. ACM, 2012.

[9] M. Bellare, C. Namprempre, and G. Neven. Security proofs for identity-based identification and signature schemes. J. Cryptology, 22(1):1–61, 2009.

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

[11] 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 and Communications Security, pages 491–500. ACM, 2011.

[12] C. Ng and P. Lee. Revdedup: A reverse de-duplication storage system optimized for reads to latest backups. In Proc. of APSYS, Apr 2013.

[13] 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.

[14] K. Zhang, X. Zhou, Y. Chen, X.Wang, and Y. Ruan. Sedic: privacyawaredata intensive computing on hybrid clouds. In Proceedings of the 18th ACM conference on Computer and communications security, CCS'11, pages 515–526, New York, NY, USA, 2011.

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