International Journal of Engineering &Technology, 7 (3.12) (2018) 933 -937

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper



Redundancy Prevention and Secure Audit of Encrypted BigData in HDFS Cloud using CloudGaurd^{+} System

Vinit Atul Shevade[1], D. A. Kulkarni[2]

[1](P.G) M.Tech Student, [2]Associate Professor,
[1,2]Department of Computer Science & Engineering, KLE's Gogte Institute of Technology, Udyambag, Belagavi.
**Corresponding Author Email: [1]vinit.sam23@gmail.com, [2]dak@git.edu*

Abstract

The recent emerging Cloud Storageand Computing Technology provides a better approach for administrations and service provisioning over the Internet byoffering various computing and storage resources. The key significant and favored cloud service's is information storage. Therefore, to maintain and conserve the privacy of data owners, data is often encrypted and stored onto the cloud as ciphered data. Nevertheless, a ciphered data introduces some issues for cloud data redundancy, which has been proved to be critical for Bigdata storage and computation in cloudenvironment. Conventional redundancy prevention methods do not work efficiently over ciphered data. Existing systems forciphered data redundancy prevention raises security issues and eventually effects the cloud processing. These existing systems are not capable of efficiently handling data access control and revocations. Therefore aiming to overcome these issues and achieving both redundancy prevention and integrity of cloud data, in this paper we propose a system, namely CloudGaurd^{+}, to prevent redundancy of the stored ciphered data in HDFS Cloud based on authentication provocation and cryptography techniques. CloudGaurd^{+} incorporates cloud information redundancy prevention with access control. We assess its efficiency and performance based on substantial analysis and computer programming and simulations. The investigation and reenactment comes about demonstrates the unrivaled productivity and adequacy of the CloudGaurd^{+} system for prospective deployment practically , particularly for Bigdata redundancy prevention in HDFS distributed cloud storage can be achieved.

Keywords: *Redundancy Prevention, Cloud Storage, File Access Control, BigData, Data-Tag.*

1. Introduction

Cloud Storage and Computing Technologyis an recent emerging technology since past decade, which provides a better approach for administrations and IT services, by providing heterogeneous computing and storage resources and assigning these resources to the cloud clients based on their consumption and requirements. A vast resource pool is provided by CC (Cloud Computing) technology, by connecting variety of network services and resources together. Scalability, flexibility, pay-as-per-use and fault-tolerance are some of the prominent properties possessed by Cloud technology. These Cloud properties and characteristics have made this technology a user-likely storage and computing service platform.

Information storage service is the most significant and favored cloud service usage by the users. CSP (Cloud Service Provider) is an Cloud entity which is responsible for managing the Cloud services according to the user requirements. The CSP (Cloud Service Provider) is also responsible for maintaining the privacy and confidentiality of the data and other personal information uploaded by the Cloud clients. However, as this user sensitive data are likely to be effected by the intrusions and attacks, the CSP's alone are not capable of handling them [10]. Therefore, the Cloud clients cannot completely trust the Cloud Service Provider.

The rapid development of analysis technologies such as data-mining and other, has led to the data privacy issue which has become a serious concern [9]. Moreover, data privacy leakages issues [11] which results in the high data security risks, especially due to the loss of control over their own private and confidential data [12], [13]. Hence, to ensure data privacy and security, storing only the encrypted data to the cloud is the most effective approach. This approximation eventually provides user privacy [8] in Cloud environment. In Cloud environment, as the data are shared among many users, the same or different clients may upload redundant data in ciphered form to Cloud Service Provider. This scenario increases the concern about the data security in Cloud. The expansion and development of existing as well as new services requires to establish efficient RMM (Resource Management Mechanism's) [14], [15], this is due to the obvious reason, the cloud storage space is huge, data redundancy greatly limits the storage and computing resources and at the same time complicates data managing system. Therefore, redundancy prevention mechanism becomes crucialfor Bigdata storage and computation in the Cloud Environment.

Redundancy prevention (RP) mechanism has proven to be cost efficient. For example, Redundancy prevention mechanism (De-duplication) reducing up to 85-97% storage requirement for backup applications [7] and 65% in standard FSs (File Systems) [6].

Thus this cost efficiency directly and indirectly, has a huge impact on cloud users and also significantlyeffects to the commercial aspects of cloud environment. But managing encrypted data storage over the Cloud with de-duplication in awell-organized manner is a practical concern. However, current industrial Redundancy prevention solutions are not capable for handling encrypted data on the cloud. Especially, brute-force attacks

Copyright © 2018 Authors. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

conversely affect the working of existing solutions[1],[5]. The existing systems cannot support data access and management control at the same time and therefore these systems cannot ensure good performance with Security, Reliability and Privacy[17].Due to some obvious reasons related to cloud environment, practically, it is difficult to achieve and manage Redundancy Prevention (RP).Therefore, in many situations Cloud Service Provider fails to cooperate with data owners and users on Redundancy Prevention of stored data on Cloud.

In this paper, focusing at achieving Integrity in stored data and Redundancy Prevention in HDFS cloud environment, we propose a highly secured system namely, CloudGaurd$^+$ System. A TPA i.e. Third Party Auditor, which is an Auditing Entity, is centrally implemented by our proposed CloudGaurd$^+$ System. MapReduce cloud, a programming model, is maintained by the TPA, which facilitates cloud clients to generate DTs (data-tags) before storing their personal data on to the Cloud. The Auditor is also responsible for auditing the integrity of client data on the cloud. Convergent Encryption (CE) ensures data ciphering (isolation) in Redundancy Prevention mechanism. The system also addresses the problem and efficiently introduces a assured convergent encryption for systematic encryption without taking into consideration the various issues and overheads of the key management. Especially, the benefaction of this paper can be summarized as below:

- The proposed system generates DTs (data-tags) before uploading the files and checking redundant storage, with secure challenge and Bigdata support. This approach efficiently helps to verify and handle data-ownership issues.
- By proposing a CloudGaurd$^+$ system to manage encrypted data storage with RP mechanism, we aim at saving overall storage space in the cloud and at the same time preserve the privacy of cloud clients.

The organization of the following sections in this paper is presented as: Section 2 gives a brief overview of related work and background in both the areas i.e. integrity auditing and Redundancy Prevention. Section 3 introduces some prerequisites and preliminary concepts. The detailed description of our proposed CloudGaurd$^+$ system is presented in the Section 4. The security and system performance is analyzed in the Section 5. The last section concludes in our work.

2. Related Work

The technologies Dropbox[18], Google Drive [19], Mozy [20] are some of the prominent CSPs (Cloud Service Providers) for data storage which perform de-duplication to provide efficient storage space management by storing only one copy of each file in the cloud space. However, if a client traditionally ciphers their data files, there is a complete loss of storage space. This is because in cryptography, different encryption keys generate different cipher data for the same or different original plain data. Existing systems and proprietary solutions (such as Dropbox [18] and Google Drive [19]) fails to prevent redundancy of encrypted data in cloud. Another efficient de-duplication system example "DeDu: De-duplication Storage System over Cloud Computing" [21], fails to handle cipher data. Since our proposed system deals with both the issues namely, data integrity audit and secure redundancy prevention mechanism, we prominently focus on reviewing the previous works in these areas.

Notwithstanding what number of customers requested to store particular file record, the server stores just a solitary copy of each document. This technique of storing only one copy of file on the server is stated as Redundancy Prevention (RP i.e. De-duplication) technique. This RP technique facilitates by efficiently saving the cloud storage space and at the same time manages the network bandwidth usage. But, side channel information leakage occurs due to trivial client side de-duplication.

One of the distinguished works supporting redundancy prevention considers making this prevention on encrypted data by mainly considering the secrecy of de-duplicated data. As a complement of public data de-duplication protocols, Ng et al. [20] introduced the private data de-duplication. Bellare et al's security definitions [22] were further revised by Abadi [39], which considered the dependencies of plaintext distributions on the public parameters. An efficient cryptographic preliminary for safeguarding the data isolation in de-duplication is the Convergent encryption [21]. "DupLESS Systems" designed by Keelveedhi et al. [4] proposed that, to encrypt file the keys based on file itself has to be derived from a key server. The derivation of such keys from the servers required a pseudorandom function protocol. Another scheme introduced by Li et al. [18] to fix the concerned issues related to significant number of keys usage in CE by utilizing the key-disperse paradigm

Halevi et al. [3] introduced the PoW (Proof-of-Ownership) protocol which lets a client efficiently prove its ownership, over the desired files, to a server. Halevi et al. [3] PoW, at some extent restricted the leakage of side-channel information.

One of the distinguished works aiming at integrity auditing is POR (Proof-of-Retrievability) [22], which assured full recovery. Proof-of-Retrievability also assures that the servers at the cloud possess the target files. An authenticated file system (a POR protocol) subjected to frequent changes is proposed Stefanov et al. [23]. Li et al. [25] proposed a new architecture, with two independent cloud servers, to reduce the load and overhead of computations at client side for cloud storage. Azraoui et al. [24] developed a new POR protocol that combines the insertion in data segments with privacy-preserving word search algorithm. Ateniese et al. [26], [27] introduced the definition of PDP (Provable-Data-Possession) which stated that without retrieving or downloading the whole data, it is possible to assure that the data servers on the cloud, stores the target files or not. However, the works prosed in Ateniese et al. [26], [38], Erway et al.[29], F. Seb´e [30], results in the computational overheads for tag generation and maintenance issues at the client. Therefore to fix these overheads and issues a proxy PDP was proposed by Wang et al. [32] in public cloud.

As mentioned above, all the related and background works reviews and examines either redundancy prevention or integrity audit at a single instance. Therefore, in this paper, our system CloudGaurd$^+$, attempts to solve these both the issues simultaneously by auditing and preventing redundancy of encrypted data.

3. Prerequisites

Here we discuss some prerequisites and preliminaries that will facilitate for better understanding and foundation of our proposed system. This section mainly details some of the encryption schemes and algorithms used in our CloudGaurd$^+$ System.

3.1 SHA512 Algorithm

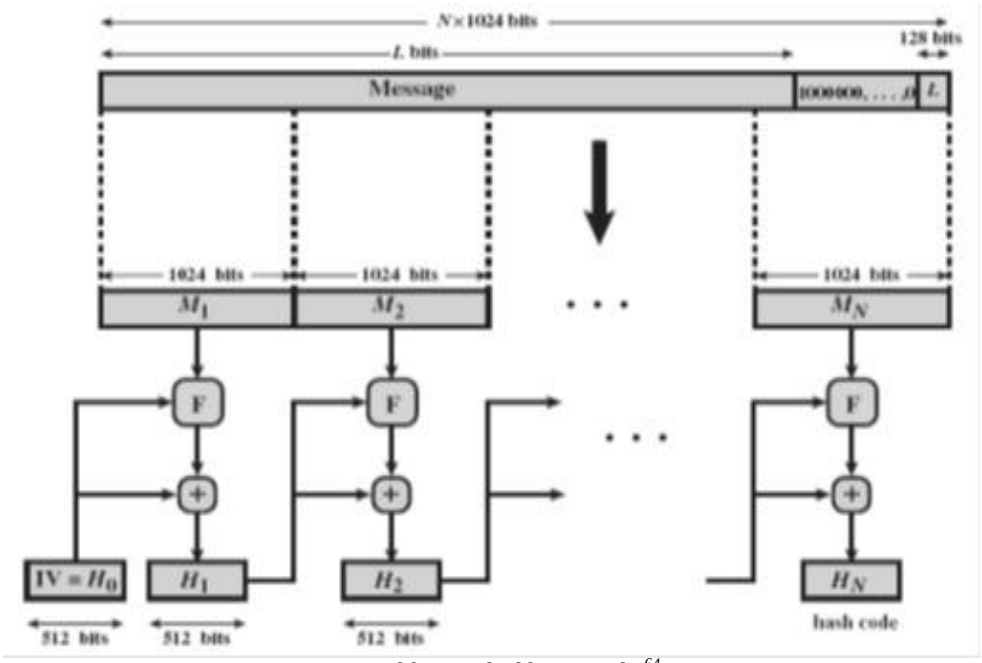
The Third Party Auditor and CSP (cloud server) performs the redundancy prevention (RP) test by sending Hash-Key value of the file. The importance and processing of a hash function for data integrity & validation [44] in redundancy prevention is as given in the following steps.

1. The Hash Key (HK) value is a function of the data in the bits format of the file, is computed by the client (or the system).
2. Both the original file and the HK value is then transmitted.
3. The same HK value calculation is performed the TPA (or CSP).
4. This newly calculated HK value with at the TPA (or CSP) is compared with the incoming HK value.
5. If TPA (or CSP) detects that the calculated HK value does not match's the incoming HK value, the TPA (or CSP) can conclude that either the HK value itself or the data file has been modified and eventually affected due to alteration.

To generate Convergent Hash Key value for each file there are number of algorithms available. In our work we have used (Secure Hash Algorithm-512) SHA-512 algorithm. If there is no redundancy, cloud client is allowed to upload his file. But, if redundancy occurs, the client has to perform a process to prove to

the CSP about its authentication & ownership by using the data-tags associated with the desired file. If this process is successful i.e. it's an authorized client, it is allowed to access this file stored in at the CSP, without the requirement of uploading the file again.

Formally the working of SHA-512 algorithm [46] is as shown in the Figure.1: (512-bit message digest: output from the Nth stage).



+ - word-by-word addition mod 2^{64}

Fig. 1: Message Digest Generation: SHA-512 Algorithm

3.2 Advanced Encryption Standard (AES) Algorithm

CE (Convergent Encryption) technique [36], [37], [38] allows a system to prevent redundant data files. Using Convergent Encryption, the proposed system can efficiently protect its stored data files without directly having possession of the cryptographic keys. Better privacy is provided by CE as compared to the traditional cloud storages.

In our work, we implement convergent encryption by using the symmetric encryption algorithm i.e. Advanced Encryption Standard (AES) algorithm [43].

AES is an effective replacement for DES. AES uses 128-bit data, and a key sizes varying as 128/192/256-bit [43]. AES mainly uses "substitution–permutation structure". AES process bytes data. Therefore, the plaintext block of 128bits is treated as 16 bytes in AES algorithm. These 16 bytes are arranged in 4 x 4matrix for processing. For 256-bit keys AES mainly uses 14 rounds, for which a different 128bit round key. All encrypted values are stored in database as key-value pair (key is token number another value is document part).

4. Proposed System: Cloudgaurd^{+}

In this section weelaborate our proposed CloudGaurd^{+} System, aiming at achieving Integrity Audit and Redundancy Prevention (RP) in HDFS cloud environment. Specifically, we begin with giving detailed description of the CloudGaurd^{+} system module and then illustrate the protocols used by our system to perform Data-Integrity and Redundancy Prevention (RP).

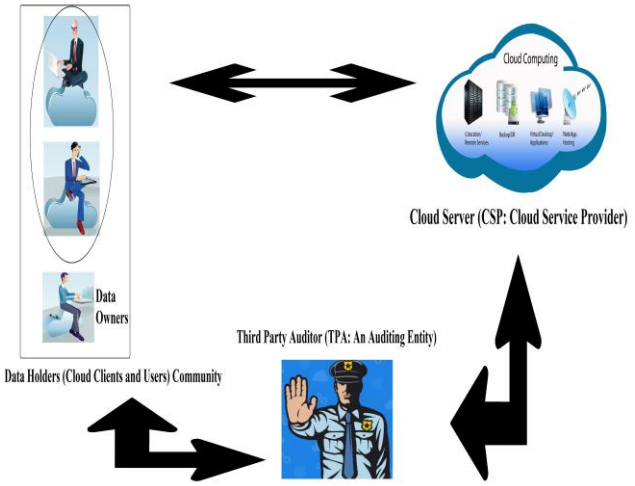


Fig. 2: Model Architecture: Cloudgaurd^{+} System

4.1 CloudGaurd^{+} SystemModel Architecture

In our proposed, CloudGaurd^{+} System there are mainly 3 entities that together facilitates for achieving redundancy prevention and Integrity Auditing. These entities are depicted in the Figure 2.

1. *Data Holders* such as, data owners and other cloud users are the cloud clients that want to store their data on the cloud storage in cloud environment.
2. *The CSP* (Cloud Service Provider) i.e. also known as Cloud Servers provides the storage and computational resources according to the specific requirements of the cloud clients. The CSP provides both Public and Private Storage. Cloud users, typically, buy or lease the cloud resources.
3. CloudGaurd^{+} System implements A *TPA i.e. Third Party Auditor*, which is an Auditing Entity, is centrally implemented by our proposed CloudGaurd^{+} System. MapReduce cloud, a programming model, is maintained by the TPA, which facilitates cloud clients to generate DTs (data-tags) before storing their personal data on to the Cloud. The Auditor is also responsible for auditing the integrity of client data on the cloud. Convergent

Encryption (SHA512 and AES algorithm) ensures data ciphering (isolation) in Redundancy Prevention (RP) mechanism.

To support redundancy prevention and integrity audit, our proposed CloudGaurd$^+$ system includes two distinguished protocols namely; File Uploading CG$^+$ protocol and File Retrieval CG$^+$ protocol. Both these protocols perform integrity auditing (using DT) before uploading and downloading the user files to and from the cloud.

File Uploading CG$^+$ protocol. This protocol is used to facilitate cloud client to upload its files on to the cloud thorough the TPA. Technically, this protocol functions as follows six steps:

- *Step1*Client before uploading the files to the cloud register's itself to the CloudGaurd$^+$ system by providing its authentication details.
- *Step 2*After successful registration, the client selects the file that is to be uploaded and generates a Hash Key value (using SHA-512) for that file. And then sends the File Name + HK of that file to the TPA.
- *Step 3* Eventually, as the TPA is connected with the Cloud Service Provider, it performs the redundancy prevention mechanism to verify the duplication of the respective file. If the redundancy occurs, the Auditor sends negative acknowledgement to the client. Then client, instead of uploading, can directly download the derided file by performing integrity audit with the CSP. But, if acknowledgement is positive (i.e. no redundancy) Step 4 occurs.
- *Step 4* The desired file is segmented into equal size blocks. The Hash Key value (using SHA-512) is generated. The segmented blocks are encrypted by using the convergent encryption (SHA512 + AES) and the combined cipher blocks and the respective hash value is send to the TPA.
- *Step 5* The TPA performs the redundancy prevention mechanism to check the duplication of the respective blocks. If the redundancy occurs, the TPA sends negative acknowledgement to the client. Then client then can directly download the derided file by performing indignity audit with the CSP. But, if acknowledgement is positive (i.e. no redundancy) Step 6 occurs.
- *Step 6* A new hash key is calculated for the desired file. The system also generates Data-Tag (DT) for the file to be uploaded. This DT is useful for integrity audit during retrieval of the file. The file and the new hash key undergoes cryptography (encryption) mechanism. This ciphered file and hash value with the are then uploaded to the CSP i.e. Cloud Storage.

File Retrieval CG$^+$ protocol. This protocol is used to facilitate cloud client to retrieve i.e. download its files from the cloud but without the involvement of the TPA. Technically, this protocol functions as follows foursteps:

- *Step 1* Client before retrieving the files from the cloud needs to login the CloudGaurd$^+$ system by providing its authentication credentials provided by the system.
- *Step 2* After successful login, to retrieve the desired file, the client selects the desired file from the list of available files on the cloud server. Then send a retrieval request for this file to the CSP.
- *Step 3* Upon receiving the request, the CSP audits for the integrity of the client for the respective file by verifying its tag value associated with the desired file. If the audit is successful, CSP sends the positive acknowledgement to the client.
- *Step 4* The acknowledgement (positive) from the CSP consists of a Secret Key, a ciphered Hash Key and the desired file in the cipher form. The Hash Key is decrypted by using the Secret Key. Then with the use of this key, file is also decrypted and retrieved in its original plane text format.

5. Security and Performance Analysis

Our proposed system provides a securely vibrant approach to prevent redundancy of the data stored in cloud by hiding the original text data from both TPA and CSP. The authentication of the proposed system is safeguarded by Data-Tag concept; Symmetric key convergent encryption and HK value Generation.

Hypothesis1. The TPA and CSP operates without overlapping, which assures that only authenticate clients and data holders can access file and the files can be de-duplicated in a secure and efficient manner.

Proof. In our proposed CloudGaurd$^+$ system, CSP is not capable of knowing the content of the file since it is always inan encrypted form. TPA also cannot access data files since the CSP has the right to block its access. In addition, the system applies proper management protocols to support efficient data storage management and data owner management to prevent redundancy of the data files at the same time.

The usage of Advanced Encryption Standard (AES) and SHA-512algorithm[44] in our proposed system ensures data confidentiality. Therefore, encrypted data file is impossibleto be obtained through the hash code. In data replication, encrypteddata filesare associated with unique secret data-tags (DTs). Thus if the same data file is under two or more user, the respective file can be retrieved successfully without any ambiguity for a particular user by using the DTs.

Hypothesis2. The Hash Key value that is the key to prevent redundancy of the data stored in cloud cannot be acquired by unauthorized cloud clients.

Proof. The Hash Key plus the encrypted data fileis used as the token for preventing redundancy cannot be obtained by unauthorized cloud clients, especially HK value due to Elliptic curve discrete logarithm problem [44]. Therefore, the token gets intercepted by some malicious attackers, HK value is still not interpretable.

To efficiently evaluate the performance and sustainability of our proposed CloudGaurd$^+$ System, we implemented and testedthe proposed scheme. Table 1elaborates our implementation andtesting platforms.

Table 1: Implementation and Testing Platform

Hardware Platform	System CPU: Pentium IV 2.4 GHz. Memory: 4GB SDRAM and 500GB HDD
Software Platform	OS: Microsoft Windows 10, Ubuntu v10.0 Programming Editor Platform: NetBeans 8.2 ProgrammingLanguage: Java SE 10 Backend (database) Platform: MySQL 5.7.22

MySQL database was used tostore data files and related information. We have focused on testing the performance ofthe redundancy preventionmechanism and algorithms used inour proposed scheme. The implemented system with the above platform specifications have resulted in efficient drastic increase in the performance and security, while achieving the redundancy prevention and secure integrity audit in the cloud storage environment.

6. Conclusion

Aiming at successful cloud storage service system, especially for Bigdata storage Managing encrypted data with redundancy prevention, we propose CloudGaurd$^+$ System. The CloudGaurd$^+$ System, a practical approach, manages the encrypted Bigdatain HDFS cloud with Redundancy Prevention based on ownership challenge and integrity audit. CloudGaurd$^+$ allows only an authenticate data clients to obtain the keys used data decryption. This facilities to securely access the encrypted data. Practical performance analysis and test showed that our proposed CloudGaurd$^+$ System is secure and efficient for Bigdata redundancy prevention under the described security model. Compared with the previous work, the computation overhead on the client side in CloudGaurd$^+$ System is significantly reduced. The practicability of our system is denoted by the computer simulation results.

References

[1] Z.O. Wilcox, "Convergent Encryption Reconsidered," 2011, http://www.mailarchive.com/ cryptography@metzdowd.com /msg08949.html.

[2] J. Pettitt, "Hash of Plaintext as Key?" http://cypherpunks.venona.com/date/ 1996/02 /msg02013.html.

[3] The Freenet Project. Freenet. https://freenetproject.org/.

[4] M. Bellare, S. Keelveedhi, and T. Ristenpart, "Message-Locked Encryption and Secure Deduplication," Proceedings of Cryptology–EUROCRYPT 2013, 2013, pp. 296–312, doi:10.1007/978-3- 642-38348-9_18.

[5] D. Perttula, B. Warner, and Z. Wilcox-O'Hearn, "Attacks on Convergent Encryption," http://bit.ly/yQxyvl.

[6] D.T. Meyer and W.J Bolosky, "A Study of Practical Deduplication," ACM Transactions on Storage, 7(4), pp. 1-20, 2012, doi:10.1145/2078861.2078864.

[7] Opendedup. http://opendedup.org/.

[8] C. Liu, C. Yang, X.Y. Zhang, and J.J. Chen, "External Integrity Verification for Outsourced Big Data in Cloud and IoT: A Big Picture," Future Generation Computer Systems, 2015, vol. 49, pp. 58-67.

[9] C.W. Tsai, C.F. Lai, H.C. Chao, and A.V. Vasilakos, "Big data analytics: a survey," Journal of Big Data, vol. 2, no. 1, pp. 1-32, 2015, doi:10.1186/s40537-015-0030-3.

[10] T.T. Wu, W.C. Dou, C.H. Hu, and J.J. Chen. "Service Mining for Trusted Service Composition in Cross-Cloud Environment," IEEE Systems Journal, 2014, vol. PP, no. 99, pp. 1-12, doi:10.1109/JSYST.2014.2361841.

[11] L.F. Wei, H.J. Zhu, Z.F. Cao, X.L. Dong, W.W. Jia, Y.L. Chen, and A.V. Vasilakos, "Security and privacy for storage and computation in cloud computing," Information Sciences, 2014, vol. 258, pp. 371-386, doi:10.1016/j.ins.2013.04.028.

[12] M. Ali, S.U. Khan, and A.V. Vasilakos, "Security in cloud computing: Opportunities and challenges," Information Sciences, 2015, vol. 305, pp. 357-383, doi:10.1016/j.ins.2015.01.025.

[13] M. Ali, R. Dhamotharan, E. Khan, S.U. Khan, A.V. Vasilakos, K.Q. Li, and A.Y. Zomaya, "SeDaSC: secure data sharing in clouds," IEEE Systems Journal, vol. PP, no. 99, pp. 1-10, 2015, doi: 10.1109/JSYST.2014.2379646.

[14] N.X. Xiong, A.V. Vasilakos, L.T. Yang, L.Y. Song, Y. Pan, R. Kannan, and Y.S. Li, "Comparative Analysis of Quality of Service and Memory Usage for Adaptive Failure Detectors in Healthcare Systems," IEEE Journal on Selected Areas in Communications, 2009, vol. 27, no. 4, pp. 495-509, doi:10.1109/JSAC.2009.090512.

[15] Y.Z. Zhou, Y.X. Zhang, H. Liu, N.X. Xiong, and A.V. Vasilakos, "A Bare-Metal and Asymmetric Partitioning Approach to Client Virtualization," IEEE Transactions on Services Computing, 2014, vol. 7, no. 1, pp.40-53, doi:10.1109/TSC.2012.32. 2332-7790 (c) 2016 IEEE.

[16] P. Meye, P. Raipin, F. Tronel, and E. Anceaume, "A Secure Two-Phase Data Deduplication Scheme," Proceedings of HPCC/CSS/ICESS, 2014, pp. 802-809, doi:10.1109/HPCC.2014.134.

[17] M. Kaczmarczyk, M. Barczynski, W. Kilian, and C. Dubnicki, "Reducing Impact of Data Fragmentation Caused by In-Line Deduplication," Proceedings of the 5th Annual International Systems and Storage Conference, 2012, pp. 15:1–15:12, doi:10.1145/2367589.2367600.

[18] Dropbox, "A File-Storage and Sharing Service," http://www.dropbox.com/.

[19] Google Drive, http://drive.google.com.

[20] Mozy, "Mozy: A File-storage and Sharing Service," http://mozy.com/.

[21] Z. Sun, J. Shen, and J.M. Yong, "DeDu: Building a Deduplication Storage System over Cloud Computing," Proceedings of IEEE International Conference on Computer Supported Cooperative Work in Design, 2011, pp. 348-355, doi:10.1109/ CSCWD.2011.5960097.

[22] H. Shacham and B. Waters, "Compact proofs of retrievability," in Proceedings of the 14th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryp-tology, ser. ASIACRYPT '08. Springer Berlin Heidelberg, 2008, pp. 90–107.

[23] E. Stefanov, M. van Dijk, A. Juels, and A. Oprea, "Iris: A scalable cloud file system with efficient integrity checks," inProceedings of the 28th Annual Computer Security Applications Conference, ser. ACSAC '12. New York, NY, USA: ACM, 2012, pp. 229–238.

[24] M. Azraoui, K. Elkhiyaoui, R. Molva, and M. ¨ Onen, "Stealthguard: Proofs of retrievability with hidden watchdogs," inComputer Security -ESORICS 2014, ser. Lecture Notes in Computer Science, M. Kutyłowski and J. Vaidya, Eds., vol. 8712. Springer International Publishing, 2014, pp. 239–256.

[25] J. Li, X. Tan, X. Chen, and D. Wong, "An efficient proof of retrievability with public auditing in cloud computing," in5th International Con-ference on Intelligent Networking and Collaborative Systems (INCoS), 2013, pp. 93–98.

[26] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," inProceed-ings of the 14th ACM Conference on Computer and Communications Security, ser. CCS '07. New York, NY, USA: ACM, 2007, pp. 598– 609.

[27] G. Ateniese, R. Burns, R. Curtmola, J. Herring, O. Khan, L. Kissner, Z. Peterson, and D. Song, "Remote data checking using provable data possession,"ACMTrans. Inf. Syst. Secur., vol. 14, no. 1, pp. 12:1–12:34, 2011.

[28] G. Ateniese, R. Di Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in Proceedings of the 4th International Conference on Security and Privacy in Communication Netowrks, ser. SecureComm '08. New York, NY, USA: ACM, 2008, pp. 9:1–9:10.

[29] C. Erway, A. K¨upc¸ ¨u, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," inProceedings of the 16th ACM Conference on Computer and Communications Security, ser. CCS '09. New York, NY, USA: ACM, 2009, pp. 213–222.

[30] F. Seb´e, J. Domingo-Ferrer, A. Martinez-Balleste, Y. Deswarte, and J.-J. Quisquater, "Efficient remote data possession checking in critical information infrastructures," IEEE Trans. on Knowl. and Data Eng., vol. 20, no. 8, pp. 1034–1038, 2008.

[31] Y. Zhu, H. Hu, G.-J. Ahn, and M. Yu, "Cooperative provable data possession for integrity verification in multicloudstorage,"IEEE Trans-actions on Parallel and Distributed Systems, vol. 23, no. 12, pp. 2231– 2244, 2012.

[32] H. Wang, "Proxy provable data possession in public clouds," IEEE Transactions on Services Computing, vol. 6, no. 4, pp. 551–559, 2013.

[33] S. Halevi, D. Harnik, B. Pinkas, and A. Shulman-Peleg, "Proofs of ownership in remote storage systems," inProceedings of the 18th ACM Conference on Computer and Communications Security. ACM, 2011, pp. 491–500.

[34] W. K. Ng, Y. Wen, and H. Zhu, "Private data deduplication protocols in cloud storage," in Proceedings of the 27th Annual ACM Symposium on Applied Computing, ser. SAC '12. New York, NY, USA: ACM, 2012, pp. 441–446.

[35] J. Douceur, A. Adya, W. Bolosky, P. Simon, and M. Theimer, "Reclaim-ing space from duplicate files in a serverless distributed file system," in 22nd International Conference on Distributed Computing Systems, 2002, pp. 617–624.

[36] M. Bellare, S. Keelveedhi, and T. Ristenpart, "Message-locked encryp-tion and secure deduplication," in Advances in Cryptology – EURO-CRYPT 2013, ser. Lecture Notes in Computer Science, T. Johansson and P. Nguyen, Eds. Springer Berlin Heidelberg, 2013, vol. 7881, pp. 296–312.

[37] M. Abadi, D. Boneh, I. Mironov, A. Raghunathan, and G. Segev, "Message-locked encryption for lock-dependent messages," inAdvances in Cryptology – CRYPTO 2013, ser. Lecture Notes in Computer Science, R. Canetti and J. Garay, Eds. Springer Berlin Heidelberg, 2013, vol. 8042, pp. 374–391.

[38] S. Keelveedhi, M. Bellare, and T. Ristenpart, "Dupless: Server-aided encryption for deduplicated storage," in Proceedings of the 22Nd USENIX Conference on Security, ser. SEC'13. Washington, D.C.: USENIX Association, 2013, pp. 179–194. [Online]. Avail-able:https://www.usenix.org/conference/usenixsecurity13/technical-essions/presentation/bellare.

[39] J. Li, X. Chen, M. Li, J. Li, P. Lee, and W. Lou, "Secure deduplication with efficient and reliable convergent key management," IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 6, pp. 1615–1625, June 2014.

[40] J. Yuan and S. Yu, "Secure and constant cost public cloud storage auditing with deduplication," in IEEE Conference on Communications and Network Security (CNS), 2013, pp. 145–153.

[41] Jingwei Li, Jin Li, DongqingXie, and Zhang Cai "Secure Auditing and Deduplicating Data in Cloud" IEEE TRANSACTIONS ON COMPUTERS, VOL. 65, NO. 8, AUGUST 2016.

[42] Zheng Yan, Senior Member, IEEE, Wenxiu Ding, Xixun Yu, Haiqi Zhu, and Robert H. Deng,Fellow, IEEE "Deduplication on Encrypted Big Data in Cloud" IEEE TRANSACTIONS ON JOURNAL NAME, MANUSCRIPT ID.

[43] William Stallings "Cryptography and Network Security: Principles and Practice, Sixth Edition" ISBN 10: 0-13-335469-5, ISBN 13: 978-0-13-335469-0.