Middle-East Journal of Scientific Research 24 (7): 2252-2259, 2016
ISSN 1990-9233
© IDOSI Publications, 2016
DOI: 10.5829/idosi.mejsr.2016.24.07.23711

Integrity Checking of Multiple Data in Cloud Storage

[1]P. Alka Devi and [2]C. Emilin Shyni

[1]Department of Computer Science and Engineering, KCG Collage of Technology, Chennai, India
[2]Department of Information Technology, KCG Collage of Technology, Chennai, India

Abstract: Cloud service providers are used for outsourcing the data by the customers. Customers use the storage of the cloud service providers. Data owner selects the data and split it into multiple parts. Then generates the Message Authentication Code (MAC) for that split data. For MAC, SHA-1 algorithm is used. It uses 160 bit key. Then encrypt the data by using the blowfish algorithm. Blowfish algorithm uses 448 bits of keys. It is more secure then advanced encryption standard algorithm. Data owner upload the data into cloud service provider with multiple copies. If the data owner needs to modify the data, he gets it from the cloud service provider (CSP) and decrypts the data. The data owner can then modify the data and generate the updated MAC for the updated data. The cloud service provider must update the modification in the other copies too. For integrity checking, the client sends the request to the service provider and receives the data. The decryption process is done at the client side and the MAC for the decrypted data is generated. The client then compares the newly generated MAC with the updated MAC. If both are equal then the data possession in service provider is confirmed. Otherwise the data possession is not confirmed.

Key words: Cloud Storage · Data Availability · Data Integrity · Data Outsourcing · Data Security

INTRODUCTION

The potential infinite storage space offered by cloud service providers(CSPs) are used by the users for storing their personal as well as confidential data and they pay for the service that they get from these Cloud Service Provider. The costing is based on the user's or clients' usage. But, these users tend to use a lot of space. The service vendors use various techniques in order to minimize redundant data and maximize the space savings. This data often needs to be stored at multiple locations for a long time due to operational purposes and regulatory compliance. The local management of such huge amount of data is problematic and costly due to the requirements of high storage capacity and qualified personnel [1].

The data once outsourced to a remote CSP may not be trustworthy as the data owners lose direct control over their sensitive data. This causes serious issues related to data integrity and data confidentiality in cloud computing. The confidentiality issue can be handled by encrypting the data before outsourcing. Provable Data Possession (PDP) is the technique for validating data integrity over remote servers, where the data owner generated a metadata for the data file to be used for later verification process [2].

One of the main design principles of outsourcing data [3] is to provide dynamic behavior of data for various applications. That is, the remotely stored data can not only be accessed by the authorized users, but also updated by the data owner. The outsourced database (ODB) model is an example of Client-Server paradigm. In ODB, the Database Service Provider (or "Server") has the infrastructure to host outsourced databases and provides efficient mechanisms for its clients to create, store, update and query the database. A client, here, is not necessarily a single entity, such as a user. Instead, he can be an administrative entity, such as an organization or a set of authorized users.

Related Work: There are many systems which provide storage, but traditional encryption techniques are not secure enough for critical data. Zhengwei Ren, Lina Wang and Qian Wang [3] proposed an enhanced dynamic proof of retrievability scheme supporting public auditability and

Corresponding Author: P. Alka Devi, Department of Computer Science and Engineering, KCG Collage of Technology, Chennai, India.

communication-efficient recovery from data corruptions. To this end, data is split up into small data blocks and each data block is encode individually using network coding before outsourcing. The main drawback is that the original file cannot be retrieved.

Ateniese [4-6] introduced "Provable Data Possession (PDP)" model and proposed an integrity verification scheme for static data using RSA based homomorphic authenticator. Juels[5] introduced PoR is stronger than PDP since it supports data retrievability of outsourced data. Cash proposed a solution providing PoR for dynamic storage in a single server setting via oblivious RAM.

Ayad F.Barsom and M.Anwar Hasan [7] proposed a map-based provable multicopy dynamic data possession (MB-PMDDP) scheme. It provides an evidence to the customers that the CSP is not cheating by storing fewer copies. It supports outsourcing of dynamic data, i.e., it supports block-level operations such as block modification, insertion, deletion and append. It allows authorized users to seamlessly access the file copies stored by the CSP. In addition, the security against colluding servers is shown and how to identify corrupted copies by slightly modifying the proposed scheme is discussed. This work has studied the problem of creating multiple copies of dynamic data file and verifying those copies stored on untrusted cloud servers.

A new PDP scheme (referred to as MB-PMDDP) is proposed, which supports outsourcing of multi-copy dynamic data, where the data owner is capable of not only archiving and accessing the data copies stored by the CSP, but also updating and scaling these copies on the remote servers.The proposed scheme is the first to address multiple copies of dynamic data.A slight modification can be done on the proposed scheme to support the feature of identifying the indices of corrupted copies.

Pasquale Puzio, Refik Molva, Melek Onen and Sergio Loureiro [8] proposed cross-user deduplication scheme. Deduplication is to store duplicate data (either files or blocks) only once. If a user wants to upload a file (block) which is already stored, the cloud provider will add the user to the owner list of that file (block).Convergent encryption can be used which meets both data confidentiality and integrity,but it has some weaknesses like dictionary attacks.This scheme does not define typical operations like edit and delete.

Zhuo Hao, Sheng Zhong and Nenghai Yu [9] proposed a new remote data integrity checking protocol for cloud storage.This is based on Sebe *et al*'s. [10] protocol which is remote data integrity checking protocol that supports data dynamics. This paper uses Sebe *et al*'s protocol for public verifiability. The proposed protocol does not leak any private information to third-party verifiers. This protocol has a very good efficiency in the aspects of communication, computation and storage costs. This system is suitable for providing integrity protection of customers' important data. Inspite of all these advantages, this paper has some major drawbacks. This paper does not support data level dynamics. In the current construction, data level dynamics can be supported by using block level dynamics. Whenever a piece of data is modified, the corresponding blocks and tags are updated. However, this can bring unnecessary computation and communication costs.

Benchmarking

Data Integrity Schemes

Provable Data Possession: G. Ateniese, R. Curtmola, R. Burns, J. Herring, L. Kissner, Z. Peterson and D. Song introduced a model for provable data possession (PDP) [11]. In this, a Provable Data Possession (PDP) model which is provably secure for remote data checking is constructed. They introduced the concept of RSA-based homomorphic verifiability tags (HVTs) which are the building blocks for PDP scheme. A PDP protocol checks whether an outsourced storage site retains a file, which consists of a collection of data blocks. The client pre-processes the file, generating a piece of metadata that is stored locally, transmits the file along with metadata to the server and may delete its local copy. The server stores the file and responds to challenges issued by the client. As part of pre-processing, the client may alter the file to be stored at the server. The client may expand the file or include additional metadata to be stored at the server. Before deleting its local copy of the file, the client may execute a data possession challenge to make sure the server has successfully stored the file. Clients may encrypt a file prior to out-sourcing the storage. At a later time, the client issues a challenge to the server to establish that the server has retained the file. The server computes a proof of possession, which it sends back to the client. Using its local metadata, the client verifies the response. This scheme:

- Allows public verifiability
- Lack of privacy preservation
- No dynamic support
- Unbound no. of queries

Proof of Retrievability: Proofs of Retrievability (POR) uses only a single cryptographic key. Also the archive needs to access only a small portion of the file. POR protocol encrypts the file and embeds a set of randomly-valued check blocks called sentinels. The use of encryption here renders the sentinels indistinguishable from other file blocks. Juels and Kaliski [12] describe a "proof of retrievability" model, where spot-checking and error-correcting codes are used to ensure both "possession" and "retrievability" of data files on archive service systems. Specifically, some special blocks called "sentinels" are randomly embedded into the data file F for detection purpose and F is further encrypted to protect the positions of these special blocks. The number of queries a client can perform is also a fixed priori and the introduction of precomputed "sentinels" prevents the development of realizing dynamic data updates. In addition, public auditability is not supported in their scheme. Shacham and Waters design an improved PoR scheme with full proofs of security in the security model defined in [13]. They use publicly verifiable homomorphic authenticators built from BLS signatures [14], based on which the proofs can be aggregated into a small authenticator value and public retrievability is achieved. Still, the authors only consider static Data files. This scheme:

- Prevents dynamic auditing
- Support only a limited number of queries
- Recoverability
- Privacy preserving

In this scheme, the data are divided into a number of blocks. This technique uses the auditing protocol when solving the problem of integrity. Here any client who wants to check the integrity of the outsourced data then there is no need to retrieve full content. Here user stores only a key, which is used to encode a file F which gives the encrypted file F'. This procedure leaves the set of sentinel values at the end of the file

F'. Server only stores F'. Server doesn't know that where the sentinel value are stored because they indistinguishable from regular and it is randomly stored in the file F'.

Message Authentication Code (MAC) Method: Assume the outsourced data file F [15] consists of a finite ordered set of blocks m1; m2;. . . ; mn. One straight forward way to ensure the data integrity is to pre-compute MACs for the entire data file. Specifically, before data outsourcing, the data owner pre-computes MACs of F with a set of secret keys and stores them locally. During the auditing process, the data owner each time reveals a secret key to the cloud server and asks for a fresh keyed MAC for verification. This approach provides deterministic data integrity assurance straightforwardly as the verification covers all the data blocks. However, the number of verifications allowed to be performed in this solution is limited by the number of secret keys. Once the keys are exhausted, the data owner has to retrieve the entire file of F from the server in order to compute new MACs, which is usually impractical due to the huge communication overhead. Moreover, public auditability is not supported as the private keys are required for verification.

Comparison of Existing Methods:

Table 1: Comparison of various methods

Integrity Check Method	Advantage	Disadvantage
PDP	Data need not be downloaded for verification	Does not support data recovery
PoR	Supports data recovery	Cannot be used in original form, requires preprocessing for encoding
Digest	Low computation overhead, Suitable for small size	No public auditability
Encryption algorithm	Supports confidentiality, low computation and storage overhead at client side	No support for dynamic data
RSA Based	Supports public auditability	Computational overhead due to exponential operation

Underlying Algorithm

Blowfish Algorithm: This algorithm [16] was introduced in 1993 as an alternative to other algorithms like AES,DES etc. It is fast, compact, simple and secure. It is suitable for applications where the key does not change often.

Blowfish symmetric block cipher algorithm encrypts 64-bits of block data at a time. It follows the feistel network and the algorithm is divided into two parts.

- Key Expansion
- Data Encryption

Key-Expansion: It converts a key of at most 448 bits into several subkey arrays totaling 4168 bytes. Blowfish uses large number of subkeys. The p-array consists of 18, 32-bit subkeys: P1,P2,…………,P18. Four 32-bit S-Boxes consist of 256 entries each:

Step 1: S1,0, S1,1,………. S1,255
Step 2: S2,0, S2,1,……….. S2,255
Step 3: S3,0, S3,1,……….. S3,255
Step 4: S4,0, S4,1,..............S4,255
Step 5: xR is XORed with P**[16]**
Step 6: xL is XORed with P**[17]**.
Step 7: Finally combine xL and xR.

Decryption:

Step 1: Divide the 64 bit input data into two32-bit halves (left and right): xL and xR
Step 2: fori=17 to1xL is XORed with P[i].Find F(xL)F(xL) is XORed with xR.InterchangexL and xR.
Step 3: InterchangexL and xR.
Step 4: xR is XORed with P[1].
Step 5: xL is XORed with P[0].
Step 6: Finally combine xL and xR.

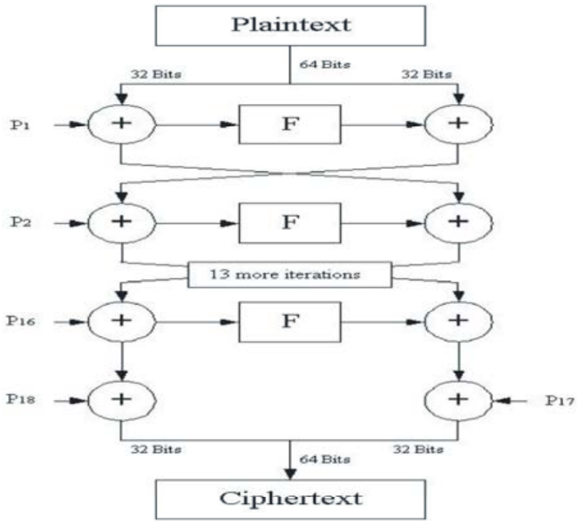

Fig. 1: Fiestel structure of blowfish

Data Encryption: It is having a function to iterate 16 times of network. Each round consists of key-dependent permutation and a key and data-dependent substitution. All operations are XORs and additions on 32-bit words. The only additional operations are four indexed array data lookup tables for each round.

Pseudo Code for Blowfish Algorithm

Encryption:

Step 1: Divide the 64 bit input data into two32-bit halves (left and right): xL and xR
Step 2: for i=0 to16xL is XORed with P[i].Find F(xL)F(xL) is XORed with xR.InterchangexL and xR.
Step 3: InterchangexL and xR

Blowfish Decryption: Decryption is exactly the same as encryption, except that P1, P2. .... P18 are used in the reverse order.

Comparison of Algorithms:

Table 2: Comparison of existing algorithms

FACTORS	AES	3DES	DES
Key Length	128,192 or 256 bits	168,112 bits	56 bits
Cipher type	Symmetric block cipher	Symmetric block cipher	Symmetric block cipher
Block Size	128,192 or 256 bits	64 bits	64 bits
Developed	2000	1978	1977
Cryptanalys is Resistance	Strong against differential, Truncated differential, linear,inter polation and square attacks	Vulnerable to differential,b rute force attacker could analyze plain text using differential cryptanalysis	Vulnerable to differential and Linear cryptanalysis, weak Substitution tables
Security	Secure	Only one weakness which is exit in DES	Proven inadequate

Proposed System: The system architecture shown in Fig. 4, explains the data uploading, downloading and integrity checking by the data owner of the data which is stored in the cloud service provider.. The proposed scheme provides an adequate guarantee that the CSP stores all copies that are agreed upon in the service contract. Moreover, the schemes support outscoring of dynamic data, i.e. it supports block-level operations such as block modification, insertion, deletion and append. The authorized users, who have the right to access the owner's file, can seamlessly access the copies received from the CSP.

Suppose that a CSP offers to store n copies of an owner's file on n different servers to prevent simultaneous failure of all copies. Thus, the data owner needs a strong evidence to ensure that the CSP is actually storing no less than n copies, all these copies are complete and correct and the owner is not paying for a service that he does not get. A solution to this problem is to use any of the previous PDP schemes to separately challenge and verify the integrity of each copy on each server. This is certainly not a workable solution; cloud servers can conspire to convince the data owner that they store n copies of the file while indeed they only store one copy. Whenever a request for a PDP scheme execution is made to one of the n severs, it is forwarded to the server which is actually storing the single copy. The CSP can use another trick to prove data availability by generating the file copies upon a verifier's challenge; however, there is no evidence that the actual copies are stored all the time. The CSP can use another trick to prove data availability by generating the file copies upon a verifier's challenge; however, there is no evidence that the actual copies are stored all the time. The main core of this cheating is that the n copies are identical making it trivial for the servers to deceive the owner. Therefore, one step towards the solution is to leave the control of the file copying operation in the owner's hand to create unique distinguishable/differentiable copies.

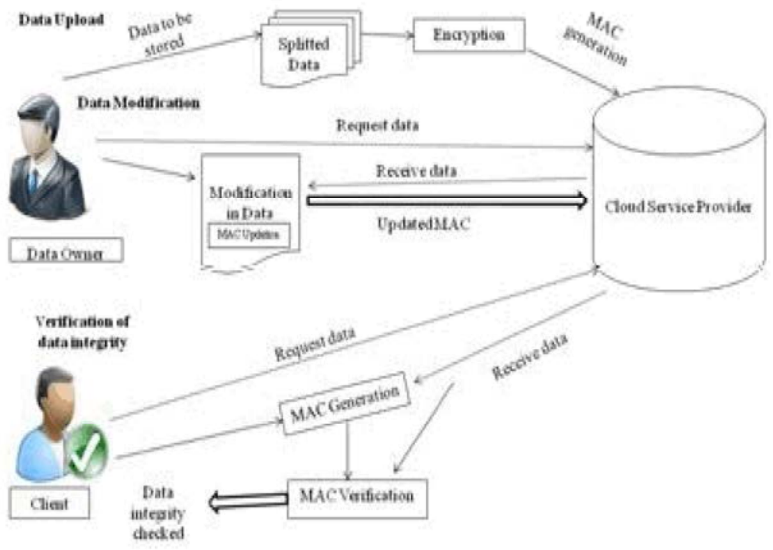

Fig. 2: System Architecture

Validating such copies of dynamic data requires the knowledge of the block versions to ensure that the data blocks in all copies are consistent with the most recent modifications issued by the owner. Moreover, the verifier should be aware of the block indices to guarantee that the CSP has inserted or added the new blocks at the requested positions in all copies. It provides an evidence to the customers that the CSP is not cheating by storing fewer copies. It supports outscoring of dynamic data, i.e. it supports block-level operations, such as block modification, insertion, deletion and append. And it also authorized users to seamlessly access the file copies stored by the CSP.

Implementation: The Fig. 4 shows the system architecture. In the data upload phase, the data owner will select the data to be uploaded into the cloud service provider. Before outsourcing the data, the data owner will split the data into several blocks. The number of splits are decided by the data owner himself.

The message authentication code (MAC) is generated for each block. The MAC generation is done with the help of SHA1 algorithm. In order to make the data more secure, the MAC is encrypted with the help of blowfish algorithm. The encrypted MAC is the outsourced into the cloud service provider (CSP).

The data download phase comprises of data modification. The data once given to the cloud service provider (CSP) can be modified by the data owner. He can add, delete, append or modify the data once outsourced. The data owner will request the data to be modified to the CSP. The CSP will send back the MAC of the requested data. Since the data owner contains the key to MAC, the data owner will get the original data. Then, the data owner will make the modifications on the data. Then he will generate the modified MAC and then send this updated MAC back to the CSP. Now the CSP contains the updated MAC.

In the verification phase, the client can verify the integrity of the data stored in the Cloud service provider(CSP).This means that the clients will make sure that the CSP contains the updated data. All the modifications are updated in the CSP. The data owner and the clients will share the secret key to decrypt the encrypted MAC.

The client will request the cloud service provider (CSP) for the data to be checked for integrity. The CSP will send back the requested MAC. In the meantime, the client locally computes the MAC. On receiving the MAC from the CSP, the client decrypts the MAC using blowfish algorithm, Then he compares this MAC with the generated MAC. It both are same, then the integrity of the data is maintained in the CSP. Otherwise, the CSP is not trustworthy.

Result and Analysis: In this section we evaluate the execution time of the various algorithms inorder to show that blowfish algorithm has the lowest execution time. For reference, we use the same parameters as [5].

Data owner registers the details with the cloud service provider and then select the data. The data are splitted. A data owner that can be an organization originally possessing sensitive data to be stored in the cloud. A CSP who manages cloud servers (CSs) and provides paid storage space on its infrastructure to store the owner's files. Authorized users — a set of owner's clients who have the right to access the remote data. The storage model used in this work can be adopted by many practical applications. MAC is generated for the splitted data. Then the data are encrypted. And uploaded into the cloud service provider. The data owner has a file F consisting of m blocks and the CSP offers to store n copies {F1,F2,...,Fn} of the owner's file on different servers — to prevent simultaneous failure of all copies — in exchange of pre-specified fees metered in GB/month.

The number of copies depends on the nature of data; more copies are needed for critical data that cannot easily be reproduced and to achieve a higher level of scalability. This critical data should be replicated on multiple servers across multiple data centers. On the other hand, non-critical, reproducible data are stored at reduced levels of redundancy. The CSP pricing model is related to the number of data copies. For data confidentiality, the owner encrypts his data before outsourcing to CSP. After outsourcing all n copies of the file, the owner may interact with the CSP to perform block-level operations on all copies. These operations includes modify, insert, append and delete specific blocks of the outsourced data copies. The users send the request to the cloud service provider. Cloud service providers send the related data to the user. An authorized user of the outsourced data sends a data-access request to the CSP and receives a file copy in an encrypted form that can be decrypted using a secret key shared with the owner [17] According to the load balancing mechanism used by the CSP to organize the work of the servers, the data-access request is directed to the server with the lowest congestion and thus the user is not aware of which copy has been received. We assume that the interaction between the owner and the authorized users to authenticate their identities and share the secret key has already been completed and it is not considered in this work. The user get the key from the data owner. And get the encrypted data from the cloud service provider. Then it decrypts the data. The authorized users, who have the right to access the owner's file, can seamlessly access the copies received from the CSP. a new PDP scheme, which supports outsourcing of multi-copy dynamic data, where the data owner is capable of not only archiving and accessing the data copies stored by the CSP, but also updating and scaling these copies on the remote servers.

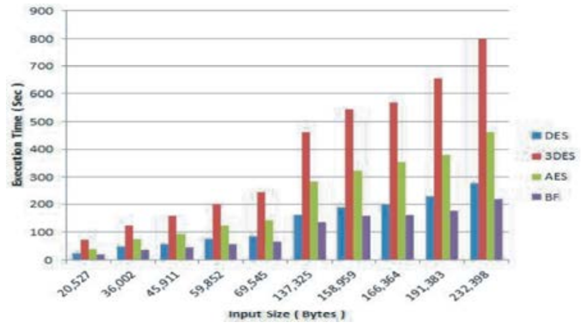

Fig. 3: Execution time of various algorithms

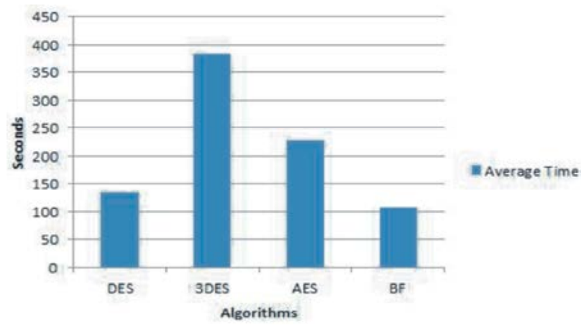

Fig. 4: Average time of various algorithms

Conclusion and Future Work: It has been seen in this paper the fact that data owners no longer physically possess their sensitive data raises new challenges to the tasks of data confidentiality and integrity in cloud computing systems. Unauthorized access and misuse of customers' confidential data are serious concerns regarding data outsourcing. The owner sends the file to be stored on a remote server which may be untrusted and deletes the local copy of the file. As a proof that the server is still possessing the data file in its original form, it needs to correctly compute a response to a challenge vector sent from a verifier — who can be the original data owner or a trusted entity that shares some information with the owner.

The data converted into MAC before it is sent for storing in the cloud.SHA1 is used for the generation. This method helps the data owner to be sure that the data being uploaded in the cloud is secure and that the data is not lost during the transmission.

With the help of Blowfish algorithm, the data is encrypted before being sent for storing. Inorder to reduce data duplication in the cloud service provider, the data owner uses another method called data de-duplication. In this method, the data owner first checks whether a particular block of data which is split is already present in the CSP. If so, the he simply does not copy that block, instead he adds a tag to that block. This helps in reducing the storage space to a huge extent.

For checking the data integrity of the data stored in cloud the clients requests the data from the CSP. It locally calculates a MAC and then compares it with the MAC given by the csp. If both match, then integrity is maintained. Otherwise the CSP does not have the updated copy of the data.

To calculate the rent that must be paid by the data owner, the constant speed model of power consumption can be used which gives maximum price optimization to the clients as well as the servers.

RFERENCES

1. Qian Wang, Kui Ren, Member, Wenjing Lou and Jin Li, 2011. Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing, in IEEE Transactions on Parallel and Distributed Systems, 22(5).
2. Gayathri, L., R. Ranjitha, S. Thiruchadai Pandeeswari and P.T. Kanmani, 2015. Preserving Data Privacy in Third Party Cloud Audit, in International Journal of Computer Science and Network, 4(6).
3. Reenu Sara George and S. Sabitha, 2013. Survey on Data Integrity in Cloud Computing, in International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), 2(1).
4. Lina Wang and Qian Wang, XXXX. Member IEEE, Rongwei Yu,Ruyi Deng, Zhengwei Ren, Dynamic Proofs of Retrievability for Coded Cloud Storage Systems, IEEE Transactions, Available: http:// www.ece.iit.edu/.
5. Ateniese, G., *et al*., 2007. Provable data possession at untrusted stores, in Proc. 14th ACM Conf. Comput. Commun. Secur. (CCS), NewYork, NY, USA, pp: 598-609.
6. Ateniese, G., L.V. Mancini, R.D. Pietro and G. Tsudik, 2008. Scalable and efficient provable data possession, in Proc. 4th Int. Conf. Secur. Privacy Commun. Netw. (SecureComm), New York, NY, USA, 2008, Art. ID 9.
7. Ateniese, G., S. Kamara and J. Katz, 2009. Proofs of storage from homo-morphic identification protocols, in Proc. 15th Int. Conf. Theory Appl. Cryptol. Inf. Secur. (ASIACRYPT), Berlin, Germany, pp: 319-333.
8. Ayad F. Barsom and M. Anwar Hasan, 2015. Provable Multicopy Dynamic Data Possession in Cloud Computing Systems, IEEE transactions on Information Forensics and Security, 10(3).
9. Pasquale Puzioio, Refik Molva, Melek Onen, Sergio Loureiro, 2013. ClouDedup:Secure Deduplication with Encrypted Data for Cloud Storage, IEEE transactions, 2013.Available: http://www.eurecom.fr/
10. Zhu, Y., H. Wang, Z. Hu, G.J. Ahn, H. Hu and S.S. Yau, 2010. Efficient provable data possession for hybrid clouds, in Proc. 17th ACM Conf. Comput. Commun. Secur. (CCS), pp: 756-758.
11. Sebé, F., J. Domingo-Ferrer, A. Martinez-Balleste, Y. Deswarte and J.J. Quisquater, 2008. Efficient remote data possession checking in critical information infrastructures, IEEE Trans. Knowl. Data Eng., 20(8): 1034-1038.

12. Juels, A. and B.S. Kaliski, Jr., 2007. Pors: Proofs of retrievability for large files, in Proc. 14th ACM Conf. Computer Communication Security (CCS), pp: 584-597.
13. Hao, Z. and N. Yu, 2010. A multiple-replica remote data possession checking protocol with public verifiability, in Proc. 2nd Int. Symp. Data, Privacy, E-Commerce, pp: 84-89.
14. Lina Wang and Qian Wang, XXXX. Member IEEE, Rongwei Yu, Ruyi Deng, Zhengwei Ren, Dynamic Proofs of Retrievability for Coded Cloud Storage Systems, IEEE Transactions, Available: http://www.ece.iit.edu/.
15. Li, J., W. Lou, K. Ren, Q. Wang and C. Wang, 2009. Enabling public verifiability and data dynamics for storage security in cloud computing, in Proc. 14th Eur. Symp. Res. Comput. Secur. (ESORICS), Berlin, Germany, pp: 355-370.
16. Mahesh S. Giri, Bhupesh Gaur and Deepak Tomar, 2015. A Survey on Data Integrity Techniques in Cloud Computing, in International Journal of Computer Applications, (0975-8887): 122(2).
17. Ayad F. Barsoum and M. Anwar Hasan, XXXX. On Data Replication and Storage Security over Cloud Computing: Are we getting what we are paying for?, in CiteseerX.