Third Party Auditor: A Study of Effective Approaches for Verifying Data Integrity

Sanket Sandesh Shahane, Raj B Kulkarni

Abstract— Technology makes life easy beyond one's imagination, provided without human intervention. Many storage service providers are facing the problem of data security that is exacting authentic scheme which guarantees data security and solves the problems, such as data leakage, corruption, loss etc. Nevertheless, the need persists to monitor different activities that provide data storage services such as Cloud. The proposed work aims at providing a system called Third Party Auditor (TPA) that detects and reports suspicious activities. TPA operates with the technique called TAG GENERATION. It provides user the logical proof to monitor the Cloud's activities without receiving any actual data from storage server, while doing integrity check, tags are compared frequently. The process of auditing is done with three different approaches such as simple, batch and random auditing with multi-user and multi-cloud support. Simple auditing verifies a single file; however, batch entertains various files in a group. Random auditing selects distinct blocks of different files. Thus, proposed system can unexceptionably provide an option for the users seeking for better data storage and integrity issues.

Index Terms— Cloud Auditing, Cloud Services, Cloud Service Provider, Data Auditor, Data Owner and Third Party Auditor.

1 INTRODUCTION

According to a study, IT outsourcing has grown at a huge rate [1]. The companies seek to trim down costs, which includes like data storage, security and its maintenance. This reduced cost helps the companies to target on their core competencies like upgrading the software and hardware, training fresh manpower and recruiting skilled ones. Thus to save the costs, the technology allows an increasing number of enterprises to outsource their various IT functions or business processes like storing, modifying, deleting, retrieving, etc., to Cloud that may give various kinds of services at a far lower price. Data storing and processing service outsourcing is a major component as most of IT functions such as Application development, Integrated services, Network Engineering and support, System engineering and support, etc., evolve around data processing [2]. Hence, security and integrity of data are crucial for outsourced data processing services, because such service providers might not be trustworthy or may not be strongly administrated. In such unscrupulous times, the system should verify intactness of data and assure security from any intervention. The system of Audit should be efficient and at the same time light enough, so do not add any overhead on both storage service provider and the users. The Audit system must work automatically and free from the control of the Storage service provider and as well as the data owners. Alberto Trombetta, Wei Jiang, et al. [3] proposed the addition of sentinels into the data. Sentinels are check blocks which were added to the original data. These are added such that for the intruder, this data seems to be original but when it is decrypted provides no information. Thus, it was helpful to secure the data even it is interpreted by hackers. But, if the intruder gets the algorithm of the sentinel application, then the data can be breached easily. Hence, it appeared ineffective in due course of time. Thus a need for specific system rose to have such process which can monitor the cloud without adding any sentinels. Hence, demand of more improvised Auditor emerged which can provide effective integrity assurance for database services. Due to the absence of such Auditor, have led to many inevitable situations, for example, recently a big robbery of the data had occurred in the South Korea of three major banks. It had shown major drawbacks in the security and storage mechanism of IT infrastructure. The data which were robbed contained information like monthly card usage, card numbers, salaries etc, which had led to a lot of tension among the account holders. The card holders were running towards the bank for cancelling the card, so that no money from their account could be transferred. This episode provided a lesson to keep the storage servers watertight, protected and detect prohibited action of culprits. In IT industry, large improvements in authentication system, firewalls and data access has to be made to avoid such incident in the future. The presented work checks the integrity of data, so that no part of data could be modified, deleted and inserted without user permission. This paper provides users, the freedom of examining the integrity which observes indifferently toward both Cloud service providers (CSP) as well Data Owners. Monitoring of data needs requires profound study of database storage as well as methods of data transfer and access which is a very complex job. The methods must be reliable and must be transparent so that questions will not arise on the way of working of Auditor. The current methods and approaches are limited to particular problem, but cannot solve all problems and also incur heavy costs of storage. The paper provides the user a method that makes user capability to verify the honesty of cloud. One can find the illegal acts of storage servers, with automatically generated report from TPA. Thus, users get an opportunity to force storage service providers to improve the policy of working. The conventional storage service provider do not have issues like integrity and security of data because internal data

- *Sanket Sandesh Shahane is currently pursuing masters degree program in computer Science and engineering in Solapur University, India, PH- +91 9423331007. E-mail: sksanket1@gmail.com*
- *Dr.Raj B. Kulkarni is Associate Professor in department of computer science engineering in Solapur University, India, PH-+91 9822002072. E-mail: raj_joy@yahoo.com*

processing was always trustworthy because data was not provided to untrusted storage under any circumstances. Consequently, the number of cloud users emerged and the trusted storage servers were compelled to store up at untrusted servers, as some time and huge investment required for scaling storage servers. Data Storage has a huge volume of data which makes system inadequate and complex to monitor the integrity and security of storage. The goal of this paper is to provide a straightforward, however elegant and economical protocol, so no overhead occurs while monitoring the integrity of outsourced database services by providing integrity assurance which is a new and challenging task. Current approaches for this problem require either change to be made in blocks of data stored in cloud, or a significant subset of the data blocks to be stored locally at the client site to check with actual data stored in the cloud. These present approaches are expensive, tough to put into practice and hazardous, especially when storage is not reliable and data is vast. Here, assuming that the data and communication are encrypted. The database system at the service provider supports request processing over encrypted data, and the problem of data privacy has been taken into consideration [4, 5]. In addition to data privacy, an important security concern in the database outsourcing paradigm is integrity [6, 7, 8, 9]. When a Data Owner (DO) receives a result of the service provider, user wants to be assured that the result is both correct and complete, where correct means that the result must originate in the owner's data and was not tampered and complete means that the result includes all records satisfying the query. Particularly, in mobile computing a severe challenge is triggered by a rising trend– in which more and more clients are accessing database services from such devices as PDAs and cell phones, which have limited storage capacity and processing power. There is also a possibility that data can be interpreted during uploading for that purpose dynamic audit operations are provided. Occasionally users did not observe their uploaded data for years that may be completely or partially deleted for saving space and maintenance cost, the user came to know only when it tries to retrieve it. The goal of the paper is to provide such an automatic mechanism which is impartial to both and follow the simple and efficient methods that are fast and error-free which provide notifications to users when Data Storage System behaves illicitly. To make the Audit system more effective the various obstacles and opportunities like Data lock-in, Data transfer bottlenecks, etc., in the Cloud are considered [10].

The rest of the paper structured as follows: In, Section II explains research background and related work. Section III addresses audit techniques and system architecture with actual implementation. In, Section IV the experimental results were discussed along with limitations and future work in Section V and finally conclude in Section VI.

2 Literature Review

This section explains the various systems which were proposed by other researchers with pros and cons. The Y. Zhu, H. Wang, Z. Hu, et al. [11], proposed a scheme which checks the integrity of data by traditional cryptography method. In this, the data is stored at the cloud and second copy is maintained at user side to check the integrity of data. But this results in waste of space and increased expense of transmission between user and data storage. Thus, it is effective but incredibly costly to retain data at both sides. Alina Oprea, Michael K. Reiter, Ke Yang [12] provided a solution to find integrity of data by using block identity number and random block number which are insufficient to check the integrity of data. Sanket Sandesh Shahane and Raj B. Kulkarni [13] explained the cloud characteristics, services provided by cloud, properties of data integrity, necessity with benefits of cloud and necessity with applications of Auditing. The paper explained about the three processes like Tag generation, Sampling auditing and Dynamic Auditing which supports third party auditing. H.C Hsiao, Y.H Lin, et al. [14] projected a study of user-friendly hash that describes some schemes which are quickest and most accurate. In this study, Chinese, Korean and Japanese characters are compared with each other for using them as hash values. It had described the strength and weakness of each scheme. Wang Qian, Cong Wang, Kui Ren, Wenjing Lou and Jin Li [15] explained that Data Storage System would attempt to hide the errors like data lost during relocation, power failure, etc. from the clients for the betterment and maintaining reputation of their own. The service provider might neglect to keep or deliberately delete hardly ever or never accessed data files which belong to an ordinary client for saving maintenance cost and storage space. For blocks, the usage of tag authentication Merkle Hash tree construction makes system more complex and slows the process. Merkle scheme has limited number of possible signatures. Thus the question arises, Can Merkle theme can solve the problems generated through modern applications? Thus there exists a doubt to handle integrity of the bulk data. G. Ateniese, R.C. Burns, R. Curtmola, et al. [16] the proposed purpose behind Provable Data Possession (PDP) was to check the servers, validate the integrity of data which were stored at doubtful servers and find the illegal actions performed like data modification and deletion. It did not consider the dynamic data auditing and various reasons behind data corruption like dishonest TPA, non-authenticated users, etc. It applied public auditing, in which the complete integrity of data verification, done without providing entire data to auditor with help of sampling strategies and RSA-based homomorphic authenticator. It also checked the way the data stored at the server so that loop holes can be closed before disaster. Disappointingly, it did not support the commercial data storage industry needs as their mechanism was appropriate only for auditing the integrity of personal data with various limitations like single user and lone storage servers. Juels and Kaliski [17] proposed the system which makes the system capable to verify the correctness and integrity of data on suspicious server. The sentinels were added to the original file and in such a way that sentinels were invisible consequently plays an important role in checking the data integrity. The sentinels made up of fake blocks of data added to the file that increases security of data but also increases the actual size of the data. The verifier with the values of the sentinels asks the suspicious server to send the associated values of sentinels associated with blocks at requested positions. If the values of both sides at server as well as client matches then indicates intact integrity of data but if the values changes then data corruption had taken place. During verification stage, some random blocks are selected for testing and the corrupted blocks provide the probability of data loss. The intensity of deception depends on number of data blocks corrupted, but there exist a serious flaw in the system which makes it ineffective. Suppose, a data corruption event had already occurred at the server and the verifier sends the sentinels to server, the sentinels at server side remained intact accidentally or purposefully, but actually, data had spoiled, in such a condition, it becomes impossible for the verifier to validate the real integrity of data. Thus, sentinels add flaw to the effectiveness of the system. Hence, Proof of Retrievability (POR) cannot be used for public databases, such as libraries, repositories or collection. Thus its use restricted to confidential data only. In addition, during verification verifier sends the position of sentinels to the storage server, the storage server gets revealed to hidden sentinels. These known locations of the sentinels must change before the misuse done by the server. Hence, reuse of sentinels becomes risky and useless for the verification which is based entirely on symmetric-key cryptography proves it less secured as compared to asymmetric-key cryptography. A. Oprea, M. K. Reiter, and K. Yang [18] proposed method of verifying the doubtful server by the help of block ciphers. The copy of data stored at client used for verifying with the data stored at server. This leads to correct result but increase the cost of data storage, which leads to wastage of space, power and memory. The storage of data at both server and client makes data storage at server useless as it violates fundamental intention behind cloud. Hence, impractical due to above provided reasons. Giuseppe Ateniese, Roberto Di Pietro, Luigi V. Mancini and Gene Tsudik [19] proposed technique based on the symmetric key cryptography, however less secure as compared to the asymmetric key cryptography even it requires less encryption, thus unsuitable for third party verification. It supports various services like data modification, append and deletion but not considered multi cloud and multi user scenario. Attila A. Yavuz and Peng Ning [20] in BAF: An Efficient Publicly verifiable Secure Audit Logging Scheme for Distributed Databases, processed without any online Trusted Third Party (TTP) support could produce publicly verifiable forward secure and aggregate signatures with near-zero storage, communication expenses and computational for the loggers. These scheme works with symmetric cryptography to offer forward security in a computationally capable way. Thus, in virtual computing clouds and protected logging on suspicious platforms it may lift some controversy. Hovav Shacham and Brent Waters [21] presented two solutions to check the integrity of the storage. The first one described with pseudorandom functions and the second for publicly verifiable proofs in bilinear groups. Consequently, both the solution combines and forms the proof which depends on homomorphic properties. The disadvantage with the pseudorandom functions requires coordination between locations of block and block number. The bilinear groups also bring drawbacks in terms of efficiency due to complex working and high memory demand. Decio Luiz Gazzoni Filho, Paulo Sergio Licciardi Messeder Barreto [22] proposed the protocol with supported content delivery with security and keeps away the intruders from the system. It made auditing easy and bends according to the condition demand, but low efficiency hits the performance badly. Thus, a further exploration especially in elliptic curve cryptography makes it necessary that boost the techniques chiefly. C. Chris Erway, Alptekin Kupcu¸ Charalampos Papamanthou, Roberto Tamassia [23] presented cost-effective and efficient constructions (DPDP) that extends the Provable Data Possession (PDP) model. It used a new version based on rank information to organize the dictionaries in which rank refers to machine learning techniques for learning the model in a ranking task. It made use of 4-tuple values connected to each skip list node, therefore form high communication cost scheme which adds overhead to the data transportation in particular for multicloud environment. Yan Zhu, Gail-Joon Ahn, Hongxin Hu, Stephen S. Yau, et al. [24] presented a technique for dynamic audit services for suspicious and outsourced storages, it explained an efficient method for periodic sampling audit to improve the performance of (Third Party Auditor)TPA's and storage service providers. The audit service constructed on the methods based on fragment structure, random sampling, and index-hash table providing the provable updates to outsourced data and timely abnormality detection. The sampling audit requires a few blocks of data for auditing which added a little, constant amount of overhead and reduces computation and communication costs. However, there exists a drawback with sampling, if corruption occurred at some blocks at storage server and blocks selected for sampling from data owner were different then the Auditor provide the result with intact integrity. Umesh Maheshwari, Radek Vingralek and William Shapiro [25] presented the problems related to latest generation approach data storage than old ones and new ways to secure like chalk and cheese way. It proposed the architecture and functioning of a database system with reliability, however, adds little load on trusted storage for storing the hash and log values. The untrusted programs were unable to read the database or modify it undetectably due to encrypted database and validated against a collision-resistant hash kept in honest storage. The model protects data and metadata evenly through Trusted Data Base (TDB) that combines encryption and hashing together. The data placed at storage server with the help of the checkpoints which made for data backup, if attacker attacks the system the data could be rolled back up to checkpoint, however, if checkpoints got lost or erased due to some internal or external reason then backup becomes a problem rather than solution.

3 METHODOLOGY

This section explains the actual working of the system which includes the important function like tag generation and techniques of auditing like simple auditing, dynamic auditing, batch auditing with support of multi-cloud as well as multiuser. The method of Sample auditing had been eliminated because if assumed some blocks of file, clean, did not indicate that complete file as undamaged and intact. Hence, the method of Random block auditing has been added that checks the blocks of different files to save time and certify the blocks only, but not the related file.

a. Basic Considerations:
Before moving to the actual implementation, first consider the various fundamental aspects of the system.

What should be an optimum block size?
The size of blocks plays a vital role in time consumption and security of data during uploading and tag generation. Different size of blocks were tested and applied for uploading, the time required as shown in the table 1.1.

File Size	Size of Block	Time Required for Uploading
5.10 Mb	20 Kb/block	55 seconds/file
5.10 Mb	50Kb/block	26 seconds/file
5.10 Mb	100Kb/block	13 seconds/file

Table 1.1

Further, the size of the block could be still increased which requires less time than required by 100Kb block, but makes impossible to recognize type of information have been interpreted from corrupted block. For small size files, generates less blocks that produces less verification tag, further, reduces the probability to detect the corruption of data. The size of block if decreased at much lower size result in increased time of hashing and uploading which provides the opportunity for the hackers to intercept data. Thus, on considering the mean of merits and drawbacks, 100 Kb size of block helps to achieve the intended purpose which adds security and reduces time for hashing and uploading.

b. Which hashing technique to use?
An algorithm that maps data of arbitrary length to data of a fixed length is called as Hash functions. It supports accelerate table lookup and performs important tasks like finding items in database, detecting duplicated or similar records in a large file and protects data. Due to its reliability and efficiency, it was recently used in internet payment system. The Hash tables store the values generated by Hash functions, this function supports the security and efficiency of data storage. Further, Cryptographic hash functions were studied because their ideal property suits the demands of data storage. It has diverse advantages like verifying the integrity of files, verifying the password, data identifier and pseudorandom functions. Through Cryptographic hash function it becomes easy to generate hash values for any message, hardly possible to generate a message that has given hash, futile to modify a message without changing the hash and impossible to find two different messages with same hash. However, hash has many applications; it must be able to withstand all different types of cryptanalytic attack and properties as follows.
1) In pre-image resistance, hash h should be difficult to find any message m such that h=hash(m), functions that lack this property are vulnerable to preimage attacks.
2) In second pre-image resistance input m1 should be such that difficult to find another input m2 such that m1 ≠ m2 and hash (m1) =hash (m2), functions that lack this property were vulnerable to second preimage attacks.
3) In collision resistance different messages m1 and m2 such that hash (m1) = hash (m2). Such a pair is called cryptographic hash collision. This property is sometimes referred to as strong collision resistance. It requires a hash value at least twice as long as that required for preimage-resistance; otherwise collisions may be found by a birthday attack.

There are different types of algorithms for hash like GOST, MD2, MD4, MD5, PANAMA, SHA-0, SHA-1, SHA-3, TIGER, WHIRLPOOL, etc. Further, a hash should have above mentioned properties but at the same time efficient that do not add overhead and threats which helped to select SHA-1 for hashing. The functioning of hash is as shown in the figure 1.1.

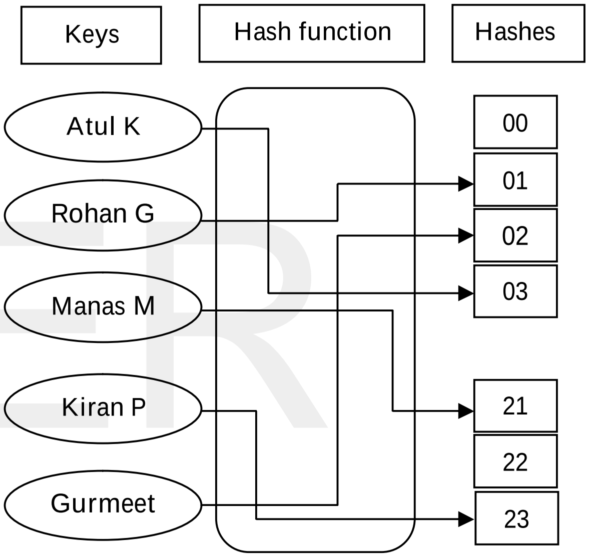

Fig.1.1

Previously during testing MD5 was used, but a successful collision attack compelled to use SHA-1. SHA stands for 'Secure Hash Algorithm'. It produces a 160-bit hash value and produces a message digest based on principles similar to the design of MD4 and MD5 algorithms. The collisions were found in both MD5 and SHA-0, so inappropriate to be used in the cryptographic security. A theoretical attack with a complexity of 2^{61} operations could be possible in SHA-1, but no such actual collisions have yet reported. There was no actual security problem with the function, as SHA-1 is practically unfeasible to break. Excellent data consistency check, imparts to best feature of SHA-1, rather than security, therefore most useful for checking integrity of data.

c. How Auditing Works?

The process of Data Auditing for checking data integrity is as follows:

1) Tag generation:

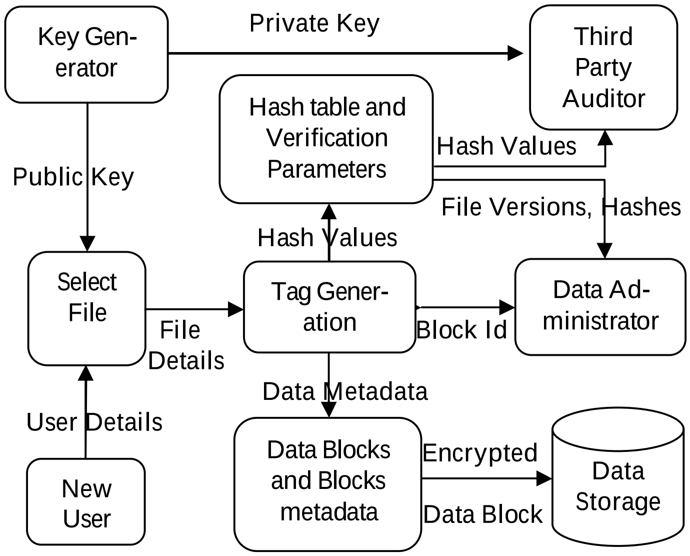

Fig. 1.2

Tag generation lays the foundation of Third Party Auditor by performing the most critical procedures. Initially, user has to make an account with its personal information like contact number, email and birth date with a username and password. During the registration of the user a public key and private key are generated using key generator. Further, when the user uploads a file to storage server the file is divided into the blocks of constant size and the hash value of each block is calculated using hashing function as discussed above. The hash table and verification parameters are provided to Auditor and Data administrator. The hash table consists of hash value, file version identification number, hash identification number and data storage number. The verification parameters consist of blocks identification number, owner identification number and hash value. The tags generated play a vital role in generating the values which support the Auditor during the process of Integrity checking. (See Fig. 1.2)

2) Simple Auditing: It is the simple and direct process of Audit check in which directly a file is selected. Before performing audit check the system must check the existence of file in database. If file exists, the system retrieves the block identification number and data owner identification number from the cloud. Hash values are provided by Data Administrator and with the help of Zero Knowledge Protocol, both values from Data Owner and Cloud are compared. The report consists of summary of integrity of data, it tells the user about the number of lost or corrupted blocks with block number. (See Fig. 1.3)

3) Batch Auditing: In batch auditing, the files are selected in batch or in group, so that no need to select a file individually. Consider, if user have uploaded a hundred files and want to check integrity of files, then he has to select every file and perform integrity check. Thus, it may become a hectic task. Simply, select a bunch of file and perform audit check. Hence, eliminates efforts and saves time to execute integrity check and the report of Auditor is provided to user individually so that it becomes comfortable for user to observe. (See Fig. 1.4)

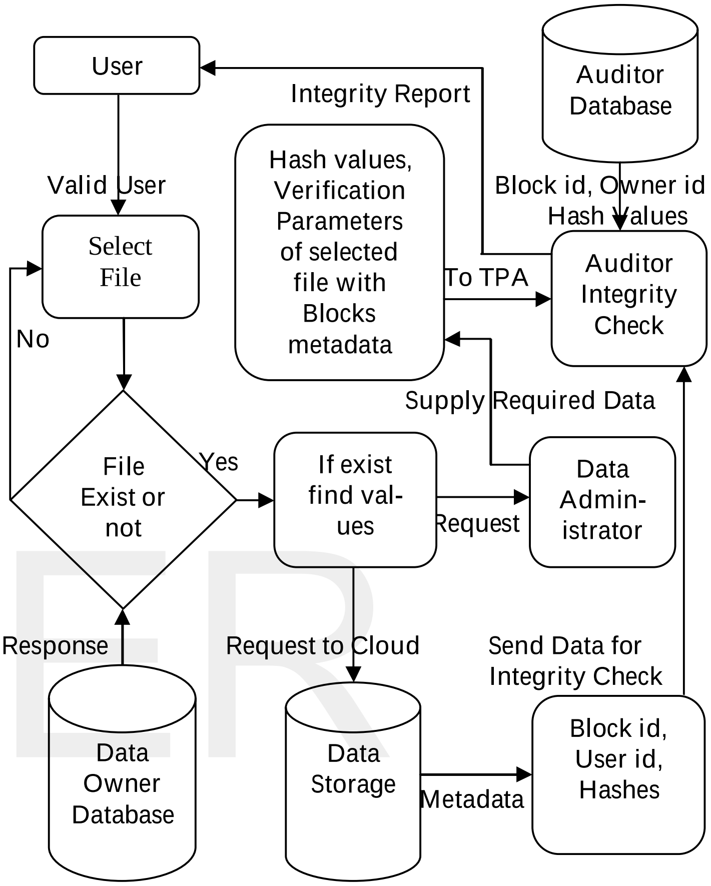

Fig. 1.3

4) Dynamic Auditing: In dynamic auditing, the integrity of the files is checked during uploading, which means the uploading process and integrity check process are done simultaneously. The aim of dynamic auditing is to avoid the interpreter from making changes in data before storing at Cloud by authenticating the user. The various attacks like forge attacks, replace attacks and replay attacks could be made futile by the help of feature called Dynamic Auditing. In this, hash of data is calculated; further signature is produced using private key of user and hash value produced before. From the auditor side, public key of the user is provided to Integrity check. The Integrity check process gets the signature timestamp during uploading; the received values are verified by the auditor. The timestamp is added to blocks which ensures data has been received in prescribed time at auditor, after expiry of time data blocks arrived from data owner are rejected, so that eliminates the probability from any interceptor from modifying data. If the signature is successfully verified then data is uploaded or if signature verification fails then data blocks are discarded, consequently, the data owner gets the information of integrity failure. (See Fig. 1.5)

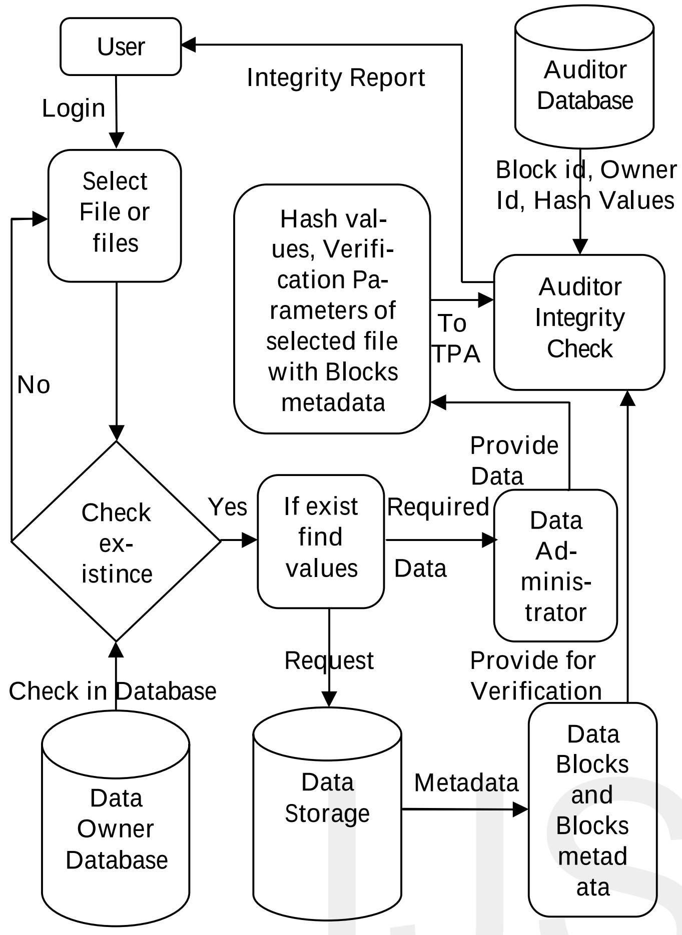

Fig. 1.4

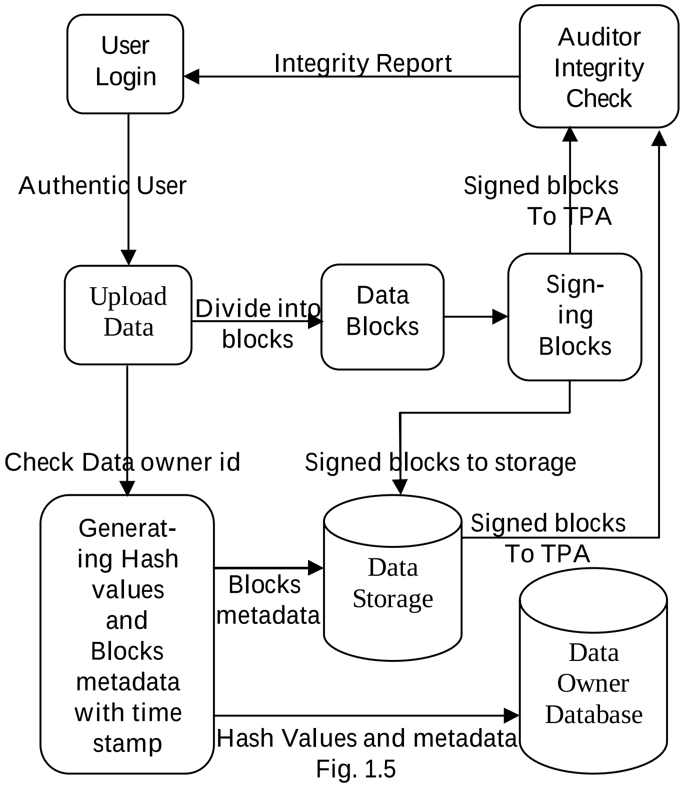

Fig. 1.5

5) Multi-User and Multi-Cloud:
This is the special feature which makes the system more useful for many users and cloud storages. In multi-user, each user is provided with unique data owner identification number. This number is used for each and every function like encryption, signature generation and verification of data owner. Due to this, a number of users can use the system and can check the integrity of data. The data owner keeps record of data storage by providing an identification number to the Cloud storage, which is provided to the Auditor at the time of integrity check. The block metadata is also provided to the Cloud storage with user identification number so to have knowledge of blocks and its owner. Thus, Auditor generates and sends report to user and provides genuine information about data integrity. (See Fig. 1.6)

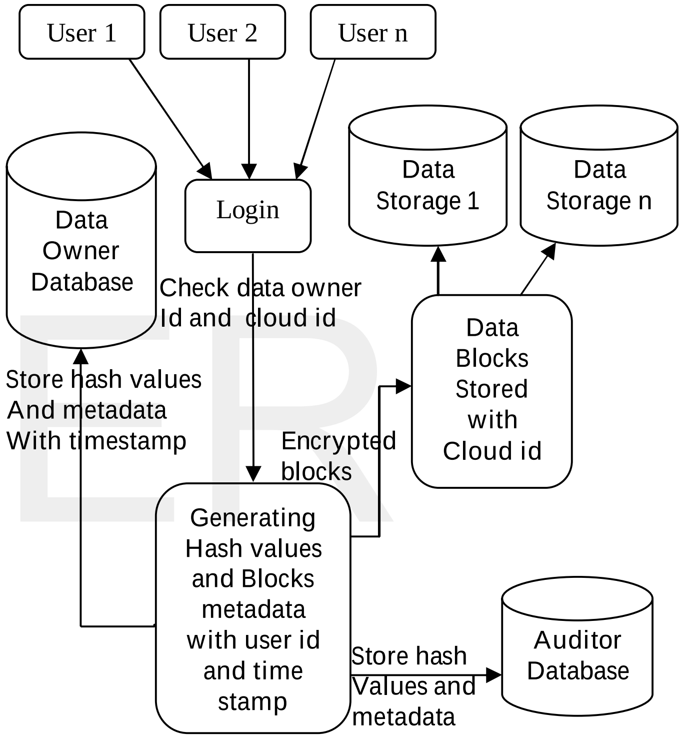
User 1
User 2
User n
Data Owner Database
Login
Data Storage 1
Data Storage n
Check data owner Id and cloud id
Data Blocks Stored with Cloud id
Store hash values And metadata With timestamp
Encrypted blocks
Generating Hash values and Blocks metadata with user id and time stamp
Store hash Values and metadata
Auditor Database

Fig. 1.6

6) Random Block Auditing: In this, whenever the Auditor is idle, then the Auditor selects blocks randomly from any files of any users and performs auditing process. However, processing the blocks do not indicate particular files, if blocks are corrupted then provides information to concern users. It could test blocks which were never checked by the user. As it selects random blocks, it can check many files in lesser time. For e.g. 100 blocks of 50 different files can be checked instead of 100 blocks of single file. Thus, effective in working and increases overall performance of the system. (See Fig. 1.7)

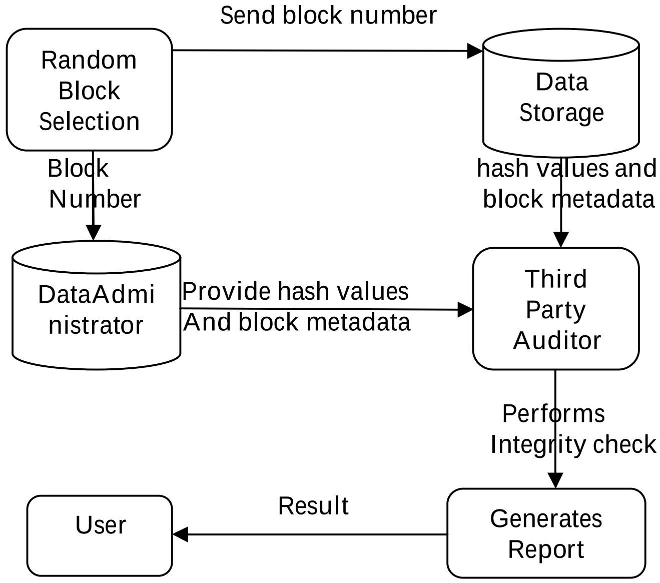

Fig. 1.7

4 Result and Implementation

Time is the important factor for each one of us. The methods are useless if consumed more time which is beyond tolerance. Thus, the above method designed in purview to shrink time, assured security and impartial toward all. The time required for the single user uploading with tag generation is shown in the figure 1.8.

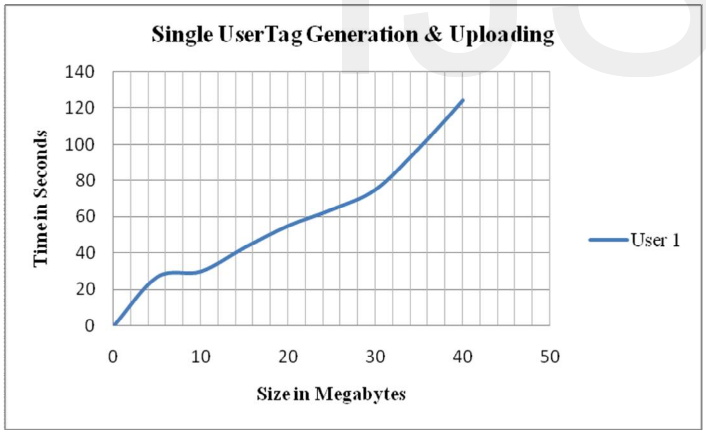

Fig. 1.8

Rohit R. Joshi et. al.[26] proposed the metrics for identifying actual author, similarly for verifying data integrity the factors such as the computation overhead for the user, the storage overhead for the server, the computation overhead for the server and the computation cost for the Auditor are considered. Additionally, the multi-user and multi-cloud environment increases the functionality of Auditor. Here, three users are assumed for the experiment. But, as the number of users increases the uploading process lags behind. Thus, time required for tag genaration with uploading in multi-user mode is explained in the fig 1.9.

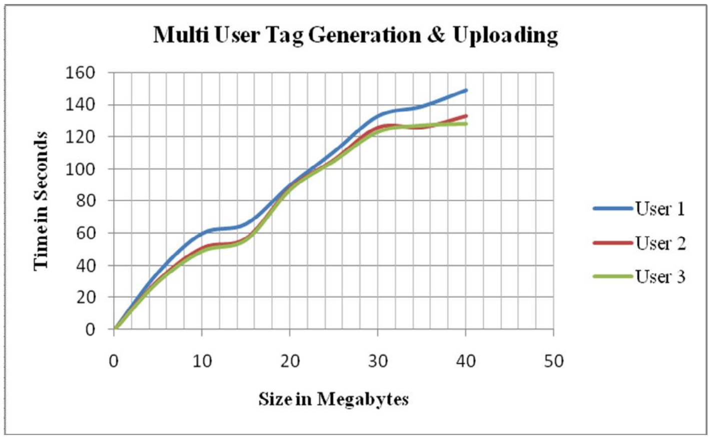

Fig. 1.9

The integrity check does not add overhead at both sides at data owner and storage server, as Auditor does not fectch any blocks. The fig 1.10 shows the time required for verifying the integrity for a single user.

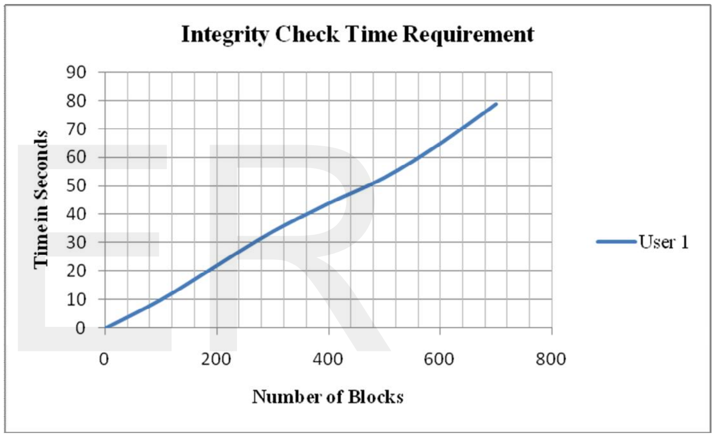

Fig. 1.10

During multi-user integrity check, the numbers of user are directly proportional to the number of blocks. Thus, Auditor has to check integrity of different blocks simultaneously, which impose some pressure on the performance as shown in fig 1.11.

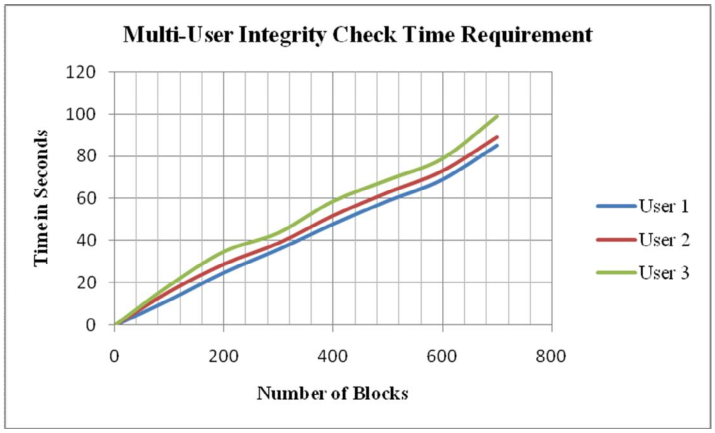

Fig. 1.11

5 Discussion And Future Work

Our result suggests that the Third Party Auditor is efficient. Auditor makes the user capable to verify the integrity of data. Hence, helpful to detect data corruption and notify the concerned user. In previous work, Auditing was carried with overhead, which increases the cost of communication. Thus, a successful attempt is made to remove the overhead by using hash values. Although, the approach of checking the integrity of data have been proposed for data storage servers which are applied on the database but not on Cloud. The communication cost between client and server is assumed to be constant. Hence, actual Cloud could be used for storage. Three numbers of users are considered during the experiment for both uploading and integrity check. Nevertheless in real time there could be thousand of the users online on Cloud who perform different operations simultaneously. The Auditor detects the data corruption, leakage or loss, but one subject that remains to be explored is how to determine the culprit who is actually responsible for the act. Sometimes, data corruption may occur due to the technical problems like power failure, hardware or software breakdown, etc., but Auditor did not recognize the exact reason behind the corruption and report directly to the user. For future researchers, the Auditor should detect the reason behind break in data integrity and implement on the Cloud. We have implemented the Auditing service on Java Platform. The experiments were conducted on a 64-bit, 2.20GHz Intel based core i3, with 2GB RAM and 2MB cache, running on Windows 8 with the Sun Java JDK 1.7.

6 Conclusion

Although, Data corruption, now have become severe problem for all users but this may affect the overall reputation and reliabilty of storage servers. In this paper, we presented a construction of dynamic audit services for untrusted and outsourced storages. We also presented an efficient method for random audit to enhance the performance of TPAs and storage service providers. Our experiments showed that our solution has no overhead, which diminish computation costs.

References

[1] Kakabadse, Andrew and Nada Kakabadse, “Trends in Outsourcing: Contrasting USA and Europe”, *European Management Journal*, 189-198, 2002.

[2] M Ozaki, Y. Adachi, Y. Iwahori, and N. Ishii, “Application of fuzzy theory to writer recognition of Chinese characters”, *International Journal of Modelling and Simulation*, 18(2), 1998, 112-116.

[3] Alberto Trombetta, Wei Jiang, Elisa Bertino and Lorenzo Bossi, “Privacy-Preserving Updates to Anonymous and Confidential Databases”, *In IEEE Transactions on Dependable and Secure Computing,* 8(4), 578-587, July-August 2011.

[4] Robling Denning, Dorothy Elizabeth, “Cryptography and data security”, *Addison-Wesley Longman Publishing Co.,* Inc., 1982.

[5] Douglas R. Stinson, “Cryptography: theory and practice”, *CRC press*, 2005.

[6] Einar Mykletun, Maithili Narasimha, and Gene Tsudik, “Authentication and integrity in outsourced databases”, *ACM Transactions on Storage (TOS)*, no. 2: 107-138, 2006.

[7] HweeHwa Pang, Arpit Jain, Krithi Ramamritham, and Kian-Lee Tan, “Verifying completeness of relational query results in data publishing." *In Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pp. 407-418. ACM, 2005.

[8] Radu Sion. "Query execution assurance for outsourced databases", *In Proceedings of the 31st international conference on Very large data bases,* pp. 601-612. VLDB Endowment, 2005.

[9] Feifei Li, Marios Hadjieleftheriou, George Kollios, and Leonid Reyzin, "Dynamic authenticated index structures for outsourced databases." *In Proceedings of the 2006 ACM SIGMOD international conference on Management of data,* pp. 121-132. ACM, 2006.

[10] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee et al. "A view of cloud computing", *Communications of the ACM 53,* no. 4, 50-58,2010.

[11] Zhu, Yan, Huaixi Wang, Zexing Hu, Gail-Joon Ahn, Hongxin Hu, and Stephen S. Yaun, "Efficient provable data possession for hybrid clouds." *Proceedings of the 17th ACM conference on Computer and communications security,* ACM, 2010.

[12] Alina Opera, Michael K. Reiter, and Ke Yang. “Space-Efficient Block Storage Integrity”, In *NDSS*, 2005.

[13] Sanket Sandesh Shahane and Raj B. Kulkarni, “Cloud Auditing: An approach for Betterment of Data Integrity”, In *Interantional Journal of Soft Computing and Engineering,* pp.107-113, 2014.

[14] H.C. Hsiao, Y.H. Lin, A. Studer, C. Studer, K.H.Wang, H.Kikuchi, A. Perrig, H.-M. Sun, and B.Y. Yang, “A Study of User-Friendly Hash Comparison Schemes”, *Proc. Ann. Computer Security Applications Conf. (ACSAC),* pp. 105-114, 2009.

[15] Wang Qian, Cong Wang, Kui Ren, Wenjing Lou and Jin Li, "Enabling public auditability and data dynamics for storage security in cloud computing", *Parallel and Distributed Systems, IEEE Transactions on, 22(5)*, 847-859, 2011.

[16] G. Ateniese, R.C. Burns, R. Curtmola, J. Herring, L. Kissner, Z.N.J. Peterson, and D.X. Song, “Provable Data Possession at Untrusted Stores”, *Proc. 14th ACM Conf. Computer and Comm. Security, pp. 598-609,* 2007.

[17] A. Juels and B. S. Kaliski, “PORs: Proofs of Retrievability for Large Files,” In *Proceedings of ACM CCS'07,* pp. 584–597,2007.

[18] A. Oprea, M. K. Reiter, and K. Yang. “Space-efficient block storage integrity”, In *Proc. of NDSS '05*, 2005.

[19] Giuseppe Ateniese, Roberto Di Pietro, Luigi V. Mancini, and Gene Tsudik. "Scalable and efficient provable data possession", In *Proceedings of the 4th international conference on Security and privacy in communication networks,* p. 9. ACM, 2008.

[20] Attila Altay Yavuz and Peng Ning. "BAF: An efficient publicly verifiable secure audit logging scheme for distributed systems." In *Computer Security Applications Conference, ACSAC'09*. Annual, IEEE pp. 219-228, 2009.

[21] Hovav Shacham and Brent Waters. "Compact proofs of retrievability." In *Advances in Cryptology-ASIACRYPT*, *Springer Berlin Heidelberg,* pp. 90-107, 2008.

[22] Gazzoni Filho, Décio Luiz, and Paulo Sérgio Licciardi Messeder Barreto. "Demonstrating data possession and uncheatable data transfer." *IACR Cryptology ePrint Archive*, 2006.

[23] Chris Erway, Alptekin Küpçü, Charalampos Papamanthou, and Roberto Tamassia. "Dynamic provable data possession." In *Proceedings of the 16th ACM conference on Computer and communications security*, *ACM*, pp. 213-222., 2009.

[24] Yan Zhu, Gail Joon Ahn, Hongxin Hu, Stephen S. Yau, Ho G. An and Chang-Jun Hu, "Dynamic audit services for outsourced storages in clouds." *Services Computing*, *IEEE Transactions on 6.2*: 227-238,2013.

[25] Umesh Maheshwari, Radek Vingralek, and William Shapiro. "How to build a trusted database system on untrusted storage." In *Proceedings of the 4th conference on Symposium on Operating System Design & Implementation-Volume 4,* pp. 10-10. USENIX Association, 2000.

[26] Joshi, Rohit R., Rajesh V. Argiddi, and Sulabha S. Apte, "Author Identification: An Approach based on Code Feature Metrics using Decision Trees", In *International Journal of Computer Applications*, 66.4, 2013.