International Journal of Innovative Research in Computer Science & Technology (IJIRCST)
ISSN: 2347-5552, Volume-3, Issue-3, May-2015

Privacy Preserving Auditing and Data Recovery for Secure Cloud Storage

Dimple A. Bedmutha, P. M. Yawalkar

***Abstract*— Cloud Storage permits users to remotely store their data and also provides users with on-demand self service from a shared pool of configurable and computable resources and that can be rapidly provisioned and realized with minimal management efforts or service provider interaction. Despite of its advantage, outsourcing storage prompts a number of interesting challenges. One of the important factors that need to be taken into consideration is to assure the user about the correctness of his outsourced data. Also, cloud user should be able to use the cloud storage without worrying for the need to verify its correctness. Thus, enabling public verifiability for cloud storage system is of critical importance so that cloud user can resort to an external audit party i.e. third party auditor (TPA) to check the correctness of outsourced data. For TPA to be secure and effective, the auditing process should not introduce no new vulnerabilities that violate users' data privacy and no additional online burden to cloud user. In this paper, a secure storage system that supports user privacy preserving and auditing is implemented using RC4 algorithm to increase efficiency. To speed up auditing by TPA, batch auditing scheme is introduced. Proposed scheme helps to achieve secure and efficient dynamic operations on blocks of data as well as recovery of corrupted data or lost data can be achieved by replicating data on backup servers.**

***Index Terms*— Privacy preserving, public verifiability, cloud computing, cloud storage, TPA, RC4, batch auditing.**

I. INTRODUCTION

Cloud computing as a service over the internet is the provision of dynamically scalable and virtualized resources. When data is stored remotely to the cloud in a flexible way when required on demand brings in ample of features [1]:

- Agility which improves with ability of user to re-provision technological infrastructure resources.
- Improvements for systems over utilization and efficiency are often only 10-20% utilized.
- Using web services as the system interface performance can be monitored and consistent as well as loosely coupled architectures is being constructed.
- Because of the increased security-focused resources, on centralization of data, etc., security could be improved; but there are still chances about leakage of sensitive data. However, when the data is distributed over a larger number of devices, complexities of security is drastically increased. Moreover, access of user to security audit logs may seem to be difficult or may be impossible. Users' desire regarding the avoidance of loss of control over information security and to retain control over the infrastructure motivated installations of private cloud.
- Cloud computing applications need not to be installed on individual user's desktop and are accessible from any corner of the world, so maintenance of cloud applications is easier.

In order to solve security issues it has become necessary to verify the data integrity at untrusted servers. For example, cloud service providers (CSP) may discard the data that has been rarely accessed, or might even hide data loss incidents in order to maintain a reputation [2]-[4], [7]. Even though outsourcing data to cloud is economically affordable for long-term large scale data storage, but it fails to provide immediate guarantee on availability and data integrity. Direct adoption of traditional cryptography is not sufficient to assure the data security protection. User auditability ensures data integrity of remotely stored data under different system and security models [2], [5]-[7]. User auditability allows third party (external auditors) to ensure the integrity of data on behalf of user. Users rely on TPA for storage security of their data and do not want this auditing process to bring in new vulnerabilities of unauthorized information leakage that leads to violation of data security [3]. Without a properly designed auditing protocol, encryption alone cannot prevent data from being the hands of external parties during the auditing.

In order, to achieve a privacy-preserving external party auditing protocol to be independent of data encryption techniques, the following two requirements has to be fulfilled: 1) The audit of cloud data storage should be efficiently performed by TPA should efficiently audit cloud data storage without having the need to demand for local copy of data, and should not bring in any on-line burden to the user; 2) The third party auditor should not introduce any new vulnerabilities that will violate privacy preserving guarantee.

II. RELATED WORK

Juels et al. [5] proposed a Proof of retrievability protocol in which a server ensures a client that a outsourced file F is correctly present on server and the client can retrieve all of

Manuscript received May 23, 2015.
Dimple A. Bedmutha, Computer Department, MET BKC, Savitribai Phule Pune University, Nasik , India, (dimple.bedmutha@gmail.com).
P. M. Yawalkar, Computer Department, MET BKC, Savitribai Phule Pune University, Nasik , India, (prashant25yawalkar@gmail.com).

F with highest possible probability. Here, two techniques such as spot-checking and Error-correcting code help to ensure possession and retrievability of data files on remote backup service systems. The Limitation with this scheme is that the number of queries a client can make is limited and fixed a priori. This approach is suitable only with encrypted data. However, the quantity of audit challenges a user will perform could be a permanent priori, and public auditability isn't supported in their main scheme. Shacham et al. [6] designed the protocols based which uses homomorphic authenticators for file blocks, which makes efficient aggregation of block integrity values in order to reduce bandwidth in PoR protocol and also to encode the file this scheme can make use of more efficient erasure code which is later transformed into an error-correcting code. Advantage of this scheme is that the unlimited number of very audits can be done, but still the solution remains static. G. Ateniese et al. [8] constructed PDP technique which is highly efficient and secure is based on symmetric key cryptography and does not require bulky encryption. The Limitation of this technique is firstly that the number of updates to be made and challenges to be made by client is restricted and fixed a prior. Secondly, block insertions cannot be made anywhere and only append-type insertions are allowed. To get dynamic solution, if even change of few bits is made to the contents of F it must propagate via error-correcting code, which introduces certain complexities such as computation and communication was highlighted by Bowers in 2009. K. D. Bowers et al. [9] introduced High-Availability and Integrity Layer which is a distributed cryptographic system that allows multiple servers to ensure a client that a outsourced data file is correct and retrievable.

C. Erway et al. [10] designed a Dynamic Provable Data Possession method which assures a client that a cloud server maintains data file F in an informal sense. But it does not provide guarantee to client for retrieval of the files. Also the scheme failed to support dynamic operations on data. Curtmola et al. [11] tries to ensure data possession of multiple replicas over distributed storage environment. The PDP scheme in [12] is extended to cover multiple replicas without having need for being encoding individual replica separately, providing guarantee and that multiple copies of the distributed data are actually maintained.

C. Wang et al. [12] designed a scheme which is effective and flexible with explicit dynamic data support which ensure the user about correctness of data in the cloud. In file distribution preparation it relies on technique of erasure correcting code to help achieve redundancies and guarantee the data dependability. The limitations of the scheme is that the user's can perform limited numbers of challenges against the server and user has an overhead to store locally the pre-computed tokens.

A. Message Authentication Code Based Scheme – I

The portions of work presented in paper [14] describes that data can be authenticated using two ways: one of the ways is to just upload the data blocks with their corresponding message authentication codes (MACs) to the cloud server along with corresponding secret key sk to the TPA. Blocks and their MACs can be retrieved randomly by TPA to verify the integrity of stored data via sk. Drawbacks of this solution: 1) its communication and computation complexity; 2) It is essential for TPA to have the understanding of the data blocks for auditing purpose which violates privacy preserving guarantee.

B. Message Authentication Code Based Scheme – II

To overcome flaws of this scheme, other way is : For the whole data file F, cloud user select s message authentication code keys $\{sk\tau\}1\leq\tau\leq s$, randomly and pre-computes s (deterministic) MACs, $\{\text{MAC } sk\tau(F)\}1\leq\tau\leq s$ and then forwards these verification metadata (keys and the MACs) to TPA. For each audit, the TPA can send the secret key sk to the cloud server and for comparison TPA requests for a fresh keyed MAC. This scheme is an improvement over the above scheme I, were TPA cannot see the data and hence it preserves privacy [14]. However, it suffers from following drawbacks: 1) Job of the TPA is to maintain and update state between audits; 2) the number of times the particular file can be audited is limited by secret keys that must be fixed initially. Once all possible secret keys are used up, the user then has to download all the data from cloud to again compute and publish new MACs to TPA.

C. Homomorphic Linear Authenticators Based Scheme

G. Ateniese et al. [2],[6],[7] proposed a Provable Data Pos-session (PDP) model which helps client who had outsourced their data that on untrusted server to make audit that the server has maintained their original data without retrieving it. For auditing, RSA based homomorphic linear authenticators (HLA) scheme is being utilized and suggests random samples a few blocks of the file. For the public verifiability the scheme demands for the linear combination of sampled blocks to be sent to third party auditor. However, the direct usage of these techniques is not suitable because the linear combinations of blocks may potentially reveal the user data information, and thus it fails to provide the guarantee of privacy being preserved. In case, if quantity of the linear combinations of the same data blocks is being collected during audit process, then by solving a system of linear equations TPA can derive the user's data content.

D. Privacy Preserving Public Auditing Scheme

C. Wang et al. [14] proposed that privacy-preserving and auditing can be achieved by uniquely integrating RSA based homomorphic authenticator with random masking technique [2], [7]. Here, the linear combination of the sampled data blocks is being masked with randomness which is generated by a pseudo random function (PRF). By using random masking method, the TPA cannot be able get all the necessary information through which a correct group of linear equations can be build up and hence it cannot derive the user's original data blocks [2], [5]. Here, HLA proposed in [6], is utilized which is based on the scheme of short signature as proposed in [15].

E. Frequently Used Algorithms

There are four algorithms in public auditing scheme: Key Generation, Signature Generation, Genproof and Verifyproof. User uses key generation algorithm to set up the scheme. RSA algorithm is used for generating keys. Verification metadata is generated by the signature generation algorithm, where signature or identity of user is generated. Genproof algorithm is run on the cloud server to check the data storage correctness in the cloud, and for auditing the proof TPA uses to audit the proof of data storage integrity. Finally, VerifyProof is run by the TPA to verify the proof of storage correctness from the cloud server.

Public auditing scheme works in two phases namely: Setup and Verify.

Setup Phase: Client executes a Key Generation which in turn initializes the secret and the public parameters of the system and then user pre-processes the data file F to create the verification metadata via Signature Generation. User stores the data file F on cloud server and delete its local copy, and forwards the verification metadata to TPA for verification to check for the correctness of stored data.

Verify Phase: The TPA sends an audit message to the cloud server in order to get an assurity that cloud server has properly retained the data file F at the time of the verification. By executing GenProof, cloud server will derive a response message for stored data file F. TPA uses verification metadata and verifies the response by executing VerifyProof algorithm [14].

III. IMPLEMENTATION DETAILS

For cloud data storage security the public auditing scheme provides complete solution for checking integrity of outsourced data. In order to enable user-privacy preserving auditing for cloud data storage, mentioned protocol design should achieve the following performance and security guarantee:

- Public verifiability: Without retrieving the original copy of stored data, TPA can verify the correctness of cloud data.
- Data integrity: must ensure that cloud service provider should not be able send the audit from third party server without indeed users' data is stored correctly.
- Guarantees' Privacy: helps to ensures that from the information which is gathered during the auditing process the TPA should not be able to derive users' data content.
- Batch auditing: enables TPA to securely and efficiently to cope with multiple auditing delegations from probably large number of different users at a time.
- Lightweight: TPA should perform verification with minimum computation and communication overhead.
- File Replication: will eliminate single point of failure and assures user about availability of their stored data, by maintaining redundancy in data distribution.
- Support Dynamics: will assure correctness of stored data, even when users used to make any modification, deletion or append their data files in the cloud.
- Recovery: will ensure recovery of corrupted or lost outsourced data.

A. Proposed Architecture

In cloud storage, the users have no control over their outsourced data. This also gives perception of the problem related to storage and also to ensure about integrity of the data stored in cloud. Cloud data storage service has different entities as shown in Fig. 1: Cloud user (U), who has bulk of data that is required to be stored on cloud server(CS); the cloud server is being managed by the cloud service provider (CSP) which provides computation resources and storage space; the third party auditor (TPA), is trusted to assess the cloud data storage service reliability on behalf of cloud users' request and user stores his data via CSP into a multiple cloud servers (backup servers), which are executing in a cooperated and simultaneous distributed manner. Each backup server has QoS factor. Through CSP, the user interacts with the backup servers to access or retrieve his data in case of loss of data stored on cloud server. It is necessary to guarantee users about their stored data correctness and maintenance on cloud server. Users rely on CS for data storage in cloud and maintenance and also they can interact dynamically with CS in order to access and update their store data. CS might keep or delete rarely accessed data files of cloud user or hide data leakage to maintain its reputation. For the purpose of ensuring the storage integrity, users rely on TPA and hope to keep data private from TPA.

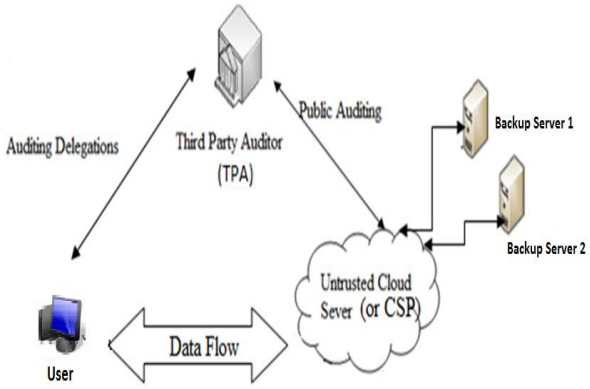

Fig 1: System Architecture

B. Privacy Preserving Auditing Module

To achieve privacy-preserving and public auditing, we propose to use the RC4 based homomorphic authenticator with random mask technique. Here, in the server's response the linear combination of sampled data blocks is being masked by randomness generated through pseudo random function (PRF) [14]. By using this, data integrity validation of the block-authenticator pairs will remain unaffected by the randomness generated via a PRF. The proposed scheme consists of Setup phase and Audit phase.

C. Batch Auditing Module

For TPA, auditing task of individual files can be tedious and very inefficient. So, this module allows TPA to

concurrently handle multiple auditing delegations from user which increases the efficiency. Due to batch auditing the computation cost on the TPA side is greatly decreased [14].

D. Data Dynamics Module

The scheme explicitly and efficiently handles dynamic data operations of outsourced data in cloud. The case, where a user wants to perform various dynamic block-level operations like updation, deletion and append operation to modify the data file while expecting the maintenance of stored data correctness assurance should be considered [7], [8]. If client wants to performs update operations such as modifying file, inserting data, or deleting data from file for any particular data block, then it is required for client to recalculate the verification token for updated data and client must also update the value of this newly calculated token to all the replicas of data in cloud storage as well as on backup servers.

E. File Replication Module

Here, the user file present on the main cloud server are replicated to backup servers and is stored on this multiple backup servers in form of backup. The proposed scheme has two backup servers apart from main cloud server. Initially, when user uploads first file, that file is being replicated to both backup servers. Suppose users latter updates the same uploaded file, then in such case the updated copy is reflected at main cloud server and backup server1 and copy before updation is present on backup server2.

F. Recovery Module

If an audit report from TPA implies that the data is not correctly present on main cloud server then user can recover the particular corrupted or lost file from the backup server by requesting main cloud server. So, cloud server makes a central call to backup servers and returns the user appropriate replicated file from either backup server1 or backup server2 and thus in such a manner it is feasible for user to restore the original file. At the same time the restored file's copy is being uploaded to main cloud server and new metadata for the same file is created and is again send to TPA. Likewise, consistency can be achieved.

G. Use of RC4 Algorithm

RC4 is a stream cipher and is officially termed as "Rivest Cipher 4". For real time processing stream ciphers prove to be more efficient over block ciphers. It is a variable key size stream cipher with byte oriented operations.

Metadata Generation:

We have divided the file in 'n' blocks where each block size is 32 bit. We have selected *k* random blocks from the file to generate metadata. For metadata generation, we used RC4 algorithm with bilinear mapping. Bilinear mapping technique is used to generate key of encryption and this key is passed to rc4 algorithm to encrypt *k* blocks.

Key Generation:

Steps:

1. Choose random "large" prime integers p and q of roughly the same size, but not too close together.
2. Calculate the product n=p*q (ordinary integer multiplication).
3. Choose a random encryption exponent *e* less than that has no factors in common with either p-1 Or q-1
4. Define d as d = e moduloInverse (p-1) (q-1)
5. Generate a key we have K = n^p^q^d

Encryption:

The RC4 algorithm produces a key stream that is XORed with the plain text. At the core of the algorithm is an 16×16 SBox (Substitution Box) which is initialised by the key. Every time a byte is produced the SBox is permuted.

The algorithm needs to share 3 variables between the two functions (a, b and SBox[]);

The procedure to initial SBox is as follows:

1. Each element in the SBox is filled with its element number (i.e. SBox[0]=0, SBox[1]=1, …..SBox[255] = 255)
2. Another array of equal size to the SBox(K) is filled with the key. Repeating the key as needed until the whole of the array has been filled.
3. The index b is set to 0 4:
   for (a = 0 to 255)
   b = (b + SBox[a] + K[b]) Mod 256
   swap SBox[a] with SBox[b]
   done

At the end of process, the key array(K) is not needed. To the next value in key stream the following steps are:

1. a = (a + 1) Mod 256
2. b = (b + SBox[a]) Mod 256
3. Swap SBox[a] with SBox[b]
4. t = (SBox[a] + SBox[b]) Mod 256
5. return SBox[t]

Metadata verification:

Verification is done by TPA. So, for this task TPA Select random number blocks *k* for a file of user U_i and generates challenge message for sampled *k* blocks and sends it to CSP. The CSP in response generates metadata M'i for blocks specified in challenge message and sends to TPA. Now, TPA compares both the metadata, one which was received by user Mi and other by CSP M'i as,

$$\sum_{i=0}^{k} Mi \stackrel{?}{=} \sum_{i=0}^{k} M'i$$

If metadata of each blocks matched then file is verified. A notification is generated to user for file verification result. Similarly, for the batch auditing, all the n files of user U_i are verified at a same time as follows:

$$\prod_{j=1}^{n} \sum_{i=0}^{k} Mi \stackrel{?}{=} \prod_{j=1}^{n} \sum_{i=0}^{k} M'i$$

IV. EXPERIMENTAL SETUP

The experiment is conducted using java on a windows system with an at least 5 Intel Pentium processor running at 1.86 GHz, 512MB of RAM. On one processor there will be

client, on second TPA and third one act as cloud server and fourth as backup server1 and fifth as backup server2 which ensures availability of data in worst case in which data is no longer correctly present on cloud server.

V. EXPERIMENTAL RESULTS

The system works as follows: Initially client runs Key Generation and the key gets generated. Using that key, the metadata is generated via Signature Generator. User sends metadata to TPA and original data to CSP. When user wants to audit the data stored on CSP, in that case TPA sends the challenge message to CSP specifying the block numbers against which metadata is required to be generated in response from CSP. Later, on receiving response in terms of metadata, the TPA compares both of the metadata and generates the audit report, which is send to user on his email address.

A . Metadata Creation:

For the system to should show the efficient performance in its execution, the privacy preserving should be achieved so, that TPA should not demand the copy of whole data and will not reveal any knowledge from the data or put more burden on the end user. We had implemented the privacy preserving auditing using RC4 algorithm and a prior work had utilized RSA algorithm [14]; Table 5.1 shows the metadata creation timing using both RSA and RC4.

From Fig. 2, we show the performance of cryptographic algorithms in terms of encryption time which gives comparison between the encryption time of RSA and RC4 algorithm over different file sizes and thus it is highlighted that RC4 takes more time to encrypt file. As Computational time has been increased, so to overcome the drawback of our mentioned proposed system, we proposed new technique as MD5 known as Message-Digest 5. MD5 is a cryptographic hash function which produces 128–bit hash value, expressed in text format as a 32 digit hexadecimal number. MD5 processes message input into fixed length output of 128 bits. Input message is being chunked into 512 bit blocks.

Steps of MD5 algorithm:

1. Append the padding bits to make length divisible by 512
   - Initially, input message is padded with one bit "1".
   - Later bit "0" is padded to make length of the message up to 64 bits less than a multiple of 512.
2. Append length
   - To represent the length of the original input message in bytes at the end of the padded message 64 bits are appended.
3. Initialize the MD buffer
   - Suppose A,B,C,D are the four buffer words, initialized to certain fixed constants.
4. Process input into 16 words blocks
   - For every input block, 4 rounds of operations are there and in each round there are 16 operations based on a non-linear function *F*, modular addition and left rotation.
   - Possible operations are: OR, AND, XOR, NOT
   - There are four possible function and a different one is used in each round:
     $F(B,C,D) = (B \text{ AND } C) \text{ OR } (\text{NOT}(B) \text{ AND } D)$
     $G(B,C,D) = (B \text{ AND } D) \text{ OR } (C \text{ OR NOT}(D))$
     $H(B,C,D) = B \text{ XOR } C \text{ XOR } D$
     $I(B,C,D) = C \text{ XOR } (B \text{ OR NOT}(D))$
5. Generate Output
   - The contents in buffer words A, B, C, D are returned in sequence with low-order byte first.

Table 2 shows the metadata creation timing using both RSA and MD5. From Fig. 3, we show the performance of cryptographic algorithms in terms of encryption time which gives comparison between the encryption time of RSA and MD5 algorithm over different file sizes and thus it is highlighted that MD5 takes less time to encrypt file; hence usage of MD5 is more efficient as compared to RSA and RC4.

TPA stores metadata in the form of triplet as,
<file_id, token_id,token_data>
eg. [<file1.txt, 7, <generatedtoken_1>>,
<file1.txt, 15, <generatedtoken_2>>,
.
.
<file1.txt, n, <generatedtoken_n>>,
]

Here, the token_id's generation varies according to file size. The generated output is hash value.

B. Batch Auditing

In case of batch auditing, TPA sends challenge message to CSP in the format of,
<User_id, token_id, file_id>
Here, for particular users' files TPA specifies selected random number of token_id's in challenge message and sends it to CSP, against which CSP has to generate response in form of generatedtokens for specified token_id's and send it to TPA for auditing. When TPA receives response from CSP, it now compares two metadata; one received from user and second which was send by CSP and generates audit report for multiple present on cloud server and forwards it to user.

C. File Replication

In case of file replication, initially file is uploaded on main cloud server and later the same is replicated from cloud server to backup server1 and backup server2.
Database representation seems as:
<cloud_server, backup_server1, backup_server2>
On initial upload of file x:
<x ,x ,x>
If uploaded file is later modified (from x to y), then database seems as:
<y, x, x>
If again same file is modified (from y to z), then database seems as:
<z, y, x> and so on.

Table 1: Comparison between RSA and RC4

File Size (in KB)	Metadata Creation Time(in ms)	
	RSA	RC4
1	5166	5564
5	13472	30271
10	14112	69338
15	32217	99347

VI. Conclusion

In this paper, we had tried to overcome most of the flaws of the existing systems. By utilizing the RC4 algorithm, users are assured that TPA server would not be able to learn any knowledge during auditing process about the original data content stored on cloud. Also, it not only eliminates users from huge auditing task but also users' fear of outsourced data leakage is being alleviated. It has been observed that for better efficiency, TPA can perform the multiple auditing

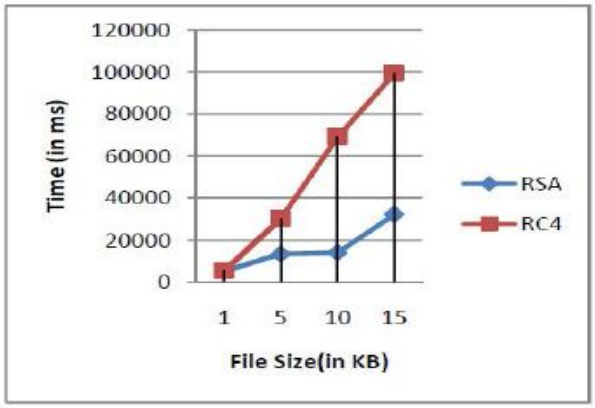

Fig 2: Analysis of RSA and RC4 algorithm

Table 2: Performance Comparison between RSA and MD5

File Size (in KB)	Metadata Creation Time(in ms)	
	RSA	MD5
1	5166	734
5	13472	5839
10	14112	8152
15	32217	8649

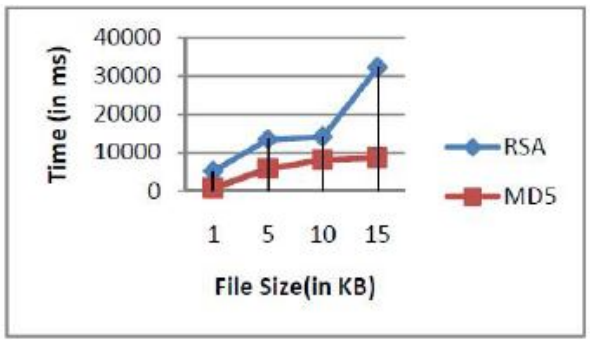

Fig 3: Analysis of RSA and MD5 algorithm

tasks in batch manner as TPA is capable of handling concurrently multiple audits of a user. Also, the scheme achieves and storage correctness assurance of dynamic data for better performance. Hence, privacy preserving auditing system provides the guarantee of cloud data correctness and availability. Here, the work using RC4 cryptographic algorithm which is stream cipher based on the use of a random permutation to ensure security and correctness of outsourced data but increases computational time; so later the results of MD5 algorithm proved that it is better than RSA and RC4. In this paper, along with public verifiability and storage correctness proof, recovery of corrupted data can be done via backup servers. Backup servers guarantee availability of data, thus the scheme also provides multi server hosting.

Acknowledgment

The authors wish to thank MET's Institute of Engineering Bhujbal Knowledge City Nasik, India for providing lab facilities. Authors are also thankful to the Cong Wang, Sherman S.-M. Chow, Qian Wang, Kui Ren and Wenjing Lou paper, "Privacy-Preserving Public Auditing for Secure Cloud Storage", for their work in this area. The mentioned paper is major guideline for this work.

References

[1] Dimple Bedmutha and P. M. Yawalkar, "A Review on User Privacy Preserving and Auditing for Secure Data Storage System in Cloud," IJCA, Dec 2014.

[2] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in Proc. of CCS'07, Alexandria, VA, October 2007, pp. 598-609.

[3] M. A. Shah, M. Baker, J. C. Mogul, and R. Swaminathan, "Auditing to keep online storage services honest," in Proc. of HotOS'07. Berkeley, CA, USA: USENIX Association, 2007, pp. 1-6.

[4] M. A. Shah, R. Swaminathan, and M. Baker, "Privacy-preserving audit and extraction of digital contents, "Privacy-preserving audit and extraction of digital contents," Cryptology ePrint Archive, Report 2008/186, 2008, http://eprint.iacr.org/.

[5] A. Juels and J. Burton S. Kaliski, "Pors: Proofs of retrievability for large files," in Proc. of CCS'07, Alexandria, VA, October 2007, pp. 584-597.

[6] H. Shacham and B. Waters, "Compact proofs of retrievability," in Proc. of Asiacrypt 2008, vol. 5350, Dec 2008, pp. 90-107.

[7] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing," in Proc. of ESORICS'09, volume 5789 of LNCS. Springer-Verlag, Sep. 2009, pp. 355-370.

[8] G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in Proc. of SecureComm'08, 2008, pp. 1-10.

[9] K. D. Bowers, A. Juels, and A. Oprea, "Proofs of Retrievability: Theory and Implementation," Cryptology ePrint Archive, Report 2008/175, 2008, http://eprint.iacr.org/.

[10] C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia,"Dynamic provable data possession," in Proc. of CCS09, 2009, pp. 213222.

[11] R. Curtmola, O. Khan, R. Burns and G. Ateniese, "MR-PDP: Multiple-replica provable data possession," in Proc. of ICDCS'08. IEEE Computer Society, 2008, pp. 411-420.

[12] C.Wang, Q.Wang, K. Ren, and W. Lou, "Ensuring data storage security in cloud computing," in Proc. of IWQoS'09, July 2009, pp. 1-9.

[13] C. Wang, Q. Wang, K. Ren, N. Cao, and W. Lou, "Towards Secure and Dependable Storage Services in Cloud Computing,"IEEE transactions on Services Computing, 06 May 2011.

[14] C. Wang, Sherman S.-M. Chow, Q. Wang, K. Ren and W. Lou, "Privacy-Preserving Public Auditing for Secure Cloud Storage,"IEEE Transactions on Cloud Computing , 2013.

[15] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the Weil pairing," J. Cryptology, vol. 17, no. 4, pp. 297-319, 2004.

[16] A. L. Ferrara, M. Greeny, S. Hohenberger, and M. Pedersen, "Practical short signature batch verification", in Proceedings of CT-RSA, volume 5473 of LNCS. Springer-Verlag, pp. 309-324.