Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.18576/amis/13S106

Verified Data Integrity with Dynamic Data in the Cloud Server

R. Nithiavathy[1,*], *K. Srihari* [2] *and S. Karthik*[3]

[1] Department of Computer Science and Engineering, Arjun College of Technology, Tamil Nadu, India
[2] Department of Computer Science and Engineering, SNS College of Engineering, Tamil Nadu, India
[3] Department of Computer Science and Engineering, SNS College of Technology, Tamil Nadu, India

Received: 7 Oct. 2018, Revised: 12 Nov. 2018, Accepted: 30 Nov. 2018
Published online: 1 Aug. 2019

Abstract: In cloud storage system, security of data stored by the owner in the cloud is exposed to risk toward the integrity of data like unauthorized data modification, server failure, misbehaving server. The proposed system guarantees security by auditing which is performed by the owner or by hiring third- party auditor who does the inspection on behalf of the owner. To preserve privacy from the third- party auditor, the complete user data blocks is not derived; instead certain signature techniques are used. It works efficiently for dynamic data operation like update, appends, etc. and also for distributed storage by using auditing mechanism which involves homomorphic authenticator and erasure code. It is a lightweight process and it has very low computation cost. The performance evaluation and security analysis show that the system works efficiently for dynamic data which ensure integrity of the data with low cost and low overheads.

Keywords: Secured storage, Error Localization, Recovery, Data dynamics, Auditing

1 Introduction

The fast-growing economy with huge population connected through internet for various reason like business solutions, social media, financial aspects, etc.,this leads to store large amount data in cloud remotely irrespectively of the geographically location without the burden of local hardware and software.In this digital era information technology plays a vital role in the world economy,almost in all enterprise.Huge data have to be stored and access from various geographical locations. The latter is achieved by cloud storage services where the users outsource the information to the cloud service providers and access anywhere through internet without deployment of local storage devices. In spite of beneficial storage and services, there are critical securities breaches existing while outsourcing data. Integrity of data check has to be verified in regular bases in order to retain the originality of the data. Service provider may cover up data loss which has occurred when there is long term storage, these are internal threats. There are external threats like hackers, intruders, unauthorized user who try to access or modify the data stored in cloud server. The reliability on the data originality is at risk which may cease the growth of cloud storage. To solve problem auditing of data in the cloud is done [[1]-[8]].Provable Data Possession(PDP) and Proof of Retriviability (POR) are two systems which are the foundations to audits the cloud server. To achieve the data integrity either the cloud user audits the files which are stored in cloud or hires a third-party auditor to do the work for them. The problem is the security breach when third- party retrieves data blocks to verify its originality. Privacy-preserving audit should be done; where third-party auditor can audit the data without knowledge of what data are stored [9].The proposed work is for the flexibility and effective verifying distributed stored data in the server for dynamic data support with correctness and availability of data to the user in cloud.

Using erasure code along with homomorphic authenticator,distribution of files in the cloud server ensures error localization, rebuilds the lost data caused by server failure, and this maintains integrity of data.Quality of cloud storage is enforces the availability of data with assured security from internal and external threats. Many previous security techniques ensures for static data and fail to secure dynamic data whose operations include edit,append, etc.Quality of Service (QOS) testing is

* Corresponding author e-mail: nithiavathyphdcse@gmail.com

achieved and yet fails when there is a system crash.The proposed work enhances distributed stored data verification scheme for dynamically changing data. The observation of the results shows that there is a decrease of computational and communication overhead compared to the traditional method of error localization data integrity check with binary result. The proposed work ensures the following features.

1.Integrated storage correctness, fast error localization with binary result for distributed data in cloud.
2.Data assurance for dynamic data.
3.High performance compared to predecessors. .

The paper is organized in section which explain the related works in sections where, the 2 section explains the methodologies and the flaw of the existing system, in section 3, the problem definition, system design and it goals of integrity of the data and error localization are explain for the proposed work, in section 4, the security of the cloud is maintenance of the files are distributed, handled, algorithms are explained with the audit mechanism is elaborated. In Section 5 explains the dynamic operation of the data in cloud; section 6 explains the analysis and performance of the proposed work.

2 Related Works

Two phases involved in all auditing protocols used to check integrity of the data are setup phase and verify phase. To verify the originality of data information stored in the cloud server using Message Authentication Code (MAC)[10] was traditionally done. For files F finite set of block $B_1, B_2, \ldots, B_n$,.The user generates a secret key and store along with data which is outsourced to the cloud server. The user challenges with MAC by utilizing security keys. Its cons are the fact that it does not work for dynamic data and number of keys are limited as computational cost increases. Hashing technique is used for large files. Each time the user checks the file by retrieving leads to communication overheads.

Signature is used for auditing mechanism instead of MAC, it calculates signature for each block F,selects random block for challenges with signature by the user for integrity check. Communication overhead is encountered and it most likely works for static data.Bilinear Signature (BLS) uses Provable Data Possession with RSA. Linear authenticator is used to authenticate without drawing out the block of data stored in the cloud. Server- aggregated authentication [12] is a linear combination of blocks which is produced by the server and stands for integrity check.All encrypted data is secured and unencrypted data is prone to risk.

Merkle Hash Tree [MHT] [13] used to check integrity of data stored in cloud server by the user or third-party auditor who is hired by user to do integrity check on behalf of them. The information is calculated from the path from root through the neighbor nodes. Cost is high for large files and does not support unencrypted data.

Alteniese et al [14] describe how Provable Data Possession (PDP) works in a probability by checking the data originality without retrieving the data from the server and does not need the whole file to access, which guarantees security. Generation of proof with small portion of the file is done. RSA-based homomorphic authenticator tags are used for block verification, local metadata and proof of meta data produced during the user challenge is compared. Once the verification is over, the local metadata is deleted.

Juels and kaleski [15] introduced the concept of proof of retrievilibity which emphasizes that the user can retrieve the data file from distributed storage remotely. Using BLS signature in Oracle mode (POR protocol) where sentinel value checks if each block is sandwiched with encrypted file. Computational cost is higher than the above mention concept.

Jin Li et al. represented a notion of two-cloud server which efficiently provides POR [16],one server for storage and other is for auditing audit. To ensure security Preprocessing is done using hash function and bilinear maps.

Yan Zhu [17] proved an interactive POR (IPOR) for dynamically-changing data. Authorized Application (AA) given by the user or third -party auditor (TPA) hired to perform audit on server. The data are fragmented into sectors and tagged a secret key which is generated and stored along with the indexed hash table is used for dynamic data.

Chris Erway et. al. [18] modeled a Dynamic Provable Data Possession (DPDP) for dynamically-changing data auditing. Ranking information is used for authentication and is formed as dictionary.

Qian Wang et. al. [19] establishes the concept of auditing publically which works for dynamic flow data in the server. Signatures for new indices are recomputed by Bilinear Signatures (BLS) or RSA (Rivest-Shamir-Adleman) based authentication. To position the data block and authentication values are issued by MHT, it suffers very low overhead for dynamic data.

Yang et. al.[20]showed how dynamic auditing is done for batch auditing. It is used for multiple clients and owners. The communication cost is reduced by using homomorphic tags. After completing successful auditing no copy is retained, all copies are deleted.

3 Problem Statement

The client hires the cloud data storage system for their data storage which is no longer stored in their local systems. Once the data is stored in cloud servers, the client have no control over it, which leads to external attacks or byzantine failure making the system more vulnerable. The adversary can erase or corrupt the

original data in the cloud data storage leading to data loss or corrupted data as the client is uncertain of the storage pool. Hence the auditing mechanism ensures the availability, integrity and the confidentiality of the client data with minimum storage, computation and communication overheads. In section, the design, threat model and notation are discussed.

3.1 System Design

There are four main entities in the cloud network

1.End user: The one who need of storage and easy retrieval of data file in the cloud environment, it may be a person, company, industry, etc.
2.Cloud server: The servers which provides storage for large data managed by cloud service provider (owner of the cloud).
3.Data Owner: Owner of the data and stores information in cloud for the end user and for its own.
4.Third-party Auditor (TPA): Groups who are expertise and capability to audit the cloud server.TPA is a trusted party hired by the user to do work for them. User must ensure correctness assurance of the data which is stored in the cloud without maintaining local copy. If the user does not have time to monitor the data stored,the work is delegated to the third party to audit for them provided no leakage to the TPA happens which is illustrated in Figure 1.

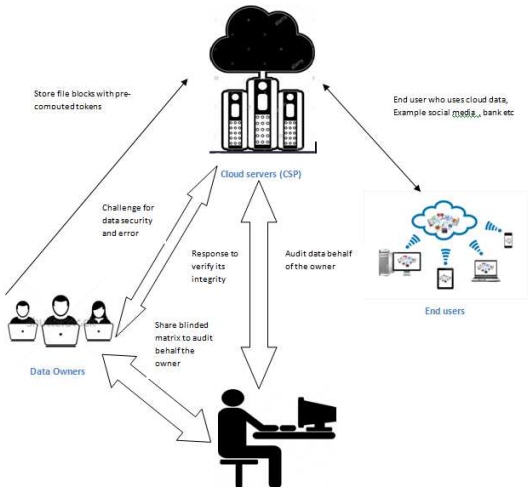

Fig. 1: Architecture of the proposed cloud storage system

3.2 Threat Model

The model captures the threat which affects both internally and externally. The data stored in cloud is prone to attack from inside like service provider can cheat or hide the data loss, move the data which is rarely used to lower storage facilities, and these are called internal attacks. Data security is at high risk from external threats like intruders who modify the data, and hackers for corrupting the whole database to create chaos and economic advantage. Data-error identification is a great significance, as of how fast data error is found and recovered from attacks.

To solve the above, the distribution of the data across the cloud server must be reviewed. The token computation usage, homomorphic tokens, along with universal hash functions used for computation of tokens, is done before storage of data in cloud [23]. Erasure-code is sandwiched with homomorphic properties[21],[22] for verification of storage uniqueness and spotting out the misbehaving servers. Data recovery and replacement of data loss is done by using erasure- correcting code. The user challenges the service provider to prove the integrity of the data stored by techniques randomly. If the user does not have time and resource to the integrity check, they hire a third party to audit the data on behalf of them. There is a possibility of security breach of data to TPA, hence privacy preserving is done.

3.3 Design Goals

A dependency of cloud-stored data to survive the adverse condition with dynamic operation requires the following must to be achieved conditions.

1.Accurate data integrity maintenance.
2.Quick flaw identification.
3.Effective dynamic data operation.
4.Data availability.
5.Minimum overhead.

3.4 Notations

1.**D**-Data File stored in cloud server
2.**E-equal** size data vector in l blocks
3.**B**-Dispersal matrix which implements Reed-Solomon coding it denoted by Galois Field.
4.H-encoded file matrix.
5.**N=s+t** for l Block
6.**v**-Version number with initial value 0
7.PRP-Pseudo random permutation
8.PRF-Pseudo random function
9.r_{ij}-seed for PRF, with index number i and j position.

4 Enhanced Cloud Securities

No local data is maintained once the data is stored in the cloud. Distributed cloud server should ensure correctness and availability anytime .Two main problem are

encountered, the first is unauthorized data access, and the second is the random byzantine failure. The error must be detected and recovered as soon as possible to avoid reading corrupted data. To ensure that the storage is safe and available, code theory is reviewed which is required to the file distribution in the cloud server. Hash function conserves the homomorphic properties to create token for computation [23]. Erasure verification code is embedded with the homomorphic properties, finding of misbehaving server, it is important to exercise the challenge –response protocol. The following notation is used in below algorithms and methods **D**-Data File stored in cloud server **E** is size data vector in l blocks, **B** is Dispersal matrix which implements Reed-solomon coding it denoted by Galois Field **G**. **H** is the encoded file matrix, $N = s+t$ for l Block,v is the Version number with initial value 0 ,PRP is Pseudo random permutation ,PRF is Pseudo random function r_{ij}- seed for PRF, with index number i and j position .

4.1 Handling Distribution of files

To handle the multiple failures in distributed data in the cloud server, erasure correcting code is used.

Consider $D =$ data file is dispersed across a set of $N = s+t$. An (s,t) in the reed Solomon erasure correcting code is used to generate t which is a redundant parity vector from s. The reconstruction of lost data with s out $s+t$ vector from other servers, thereby reducing the failure of any data loss is retrieved. Again, the original data which unchanged file vector along with t parity vector is distributed to $s+t$ server. The parity is achieved with dispersal matrix information D , derived for Vandermonde matrix [24].

The matrix D is derived after row transformation $D = (I/P)$, where I is (mxm) identity matrix and p is secret parity matrix with sxt. To form invertible matrix $s+t$ column, where D is derived by vandermonde matrix. $G = D$. B is an encoded file. . Hence there by reducing the original file vector of$D$ and s parity vector generator is based on D.

4.2 Token pre-computation and challenge

The user compute short token for verification on an individual vector. Each one contains a covered random subset of data block .The user challenges the cloud server to verify the data integrity with pre-computed tokens for random data blocks. The server gives response with signature to which the matching is done with pre-computed token. A secret P matrix is determined for all servers which operate over same subset of the indices and integrity of information stored is verified. The pre-computed token can be stored locally or in encrypted form in the server.

The algorithm1 chooses the required parameters for the token pre-computation of the data which is outsourced. $R$is the row of indices of data files per verification process.The row of indices R of data files per verification process calculated and generation of master key and K_{PRP} and challenge key K_{chal} is done from the Galois field, using the set of vector a random value for challenge is created a_i.The key is at i position using master key is generated. The token is generated for block of data which is to be stored in cloud server using random value, master key$i$ and server j and pseudorandom function. The pre-computed tokens are stored at client side locally.

Algorithm 1:

1.Procedure
2.Choose parameters l, N and function to calculate pseudo random function and pseudo random permutation
3.Choose number of T tokens
4.Choose number of indices R per verification.
5.Generate master key and challenge key, K_{prp} and K_{chal} .
6.For all position j and indices in random set of blocks calculate with l and T
7.Derive the random value challenge a_i and K_{prp} for position i from master key K_{PRP}.
8.Compute token for each block and store locally.
9.End Procedure.

Blinding each parity block is the final step before the distribution throughout the server. All encoded vectors are dispersed by the user across the cloud server after blinding parity information.

4.3 Flaw localization and Retaining Originality

To eradicate the error in the cloud server, it is very important to identify from which mode threats are generated, whether from internal source or external source. Earlier system [25], [26] is not fully involved to eradicate data-error localization and maintains integrity. Challenge and response protocol solve the above by giving the exact location of error which causes threat to data confidentiality. The following is the procedure to cross check over number of servers using challenge –response protocol.

1.The random value ai is revealed by the user along with i^{th} permutation key to each server.
2.A linear combination response is formed by aggregating the rows R with specified in-dices and position j and then sent to all the servers.
3.The user takes away blinded values from the response received from the server and verifies that the received

values are valid codes generated by the secret matrix P.
4.If the challenge is passed, the data remains with its integrity or specified rows where there exists corrupted block.

To find the misbehaving server and verify the correction of data, the following steps are involved

1.Calculate the random-value challenge to the server.
2.The user challenges the server randomly by selecting the block along with the master key to the cloud servers.
3.On receiving the key, each block computes response value for all rows.
4.If the response is equal to the pre-computed value, then the data integrity is preserved and gets ready for next challenge.
5.Otherwise it finds out the position of the misbehaving server in the cloud server. For each challenge, only the aggregated value is sent back over a set of data block. Cost of the bandwidth is low compared to the previous methods, which requires downloading all the data blocks that are challenged for integrity check.

For each challenge, only the aggregated value is sent back over a set of data block. The cost of the bandwidth is low compared to the previous methods, which requires downloading all the data blocks that are challenged for integrity check.

4.4 Retrieval and Recovery

The original file is reconstructed by downloading the vector from s server with high probability with response which are genuine values because our technique is based on random set checking. Choosing the parameter (R,N,T), the detection of data corruption is with successful file retrieval by conducting number of verifications. The value of pre-computed token and the received response assure that misbehaving servers are identified with high probability which is done by using erasure-correction implementation on blocks of the row challenged and the data loss is regenerated. The newly-recovered blocks are again distributed to server to maintaining availability and correctness. The following steps involved in error recovery technique explained algorithm 2. The following steps involved in error recovery

Algorithm 2:

1.Procedure
2.Percentage of corruption is assumed to be detected from the randomly-drawn rows from block of data file.
3.Misbehaving rows are R downloaded from the servers.
4.Server is treated with erasure correction and locks are recovered.
5.The recovered blocks are updated in server.
6.End procedure.

4.5 Audits with Privacy Preservation

The user stores the confidential data in the cloud server .Frequently, the user has to challenge the CSP to verify the integrity of the data. When the user does not have time and resource to inspect the integrity of the data ,then the user hire the auditor who is a third-party to do the work for the them [27],[28]. The parity-vector blinding process with linear property is incorporated for the new design. Secret matrix P is protected by the blinding process against cloud server. Before file distribution encoding, blinding the data vector is done followed by storage verification that can be delegated to auditing to the trusted third-party with preservation of privacy. Steps involved in private preservation in third- party auditing.

1.The individual user blinds each block of data file where k_j is secret key.
2.User generates parity vector based on blinding data vector through the secret matrix p.
3.The token is calculated with the index and the position
4.The secret matrix P, token set, permutation and challenge key are transferred to auditor.

TPA is unaware of the data content during auditing, there is only a change of the sequence of encoding precomputing token and blinding parity bits.

5 Working of Dynamic Operation

Dynamically-changing data is stored in cloud- like documents, log files, social media, business, etc., such data may be frequently updated like inserting new data, erasing and adding more information to existing data. Dynamic data also has to maintain its integrity which is very challenging without revealing secret key.

Only the user knows the secret matrix P .The change on the block is captured and storage verification token is generated, challenge response protocol is employed for the new updated file blocks. According to the changes in the data file, the user needs to correct the storage verification of token which holds the changed data block. For dynamic data operation request, the updated verification of token works effectively and executed correctly for which CSP is audited.

5.1 Update

Modification of data blocks are stored in the cloud from old data D_{ij} to modified data $D_{ij}+\Delta D_{ij}$. Reed Solomon

code with linear properties can be modified and updating the data along parity bit ΔD_{ij} constructing an updated matrix ΔD leaving behind the unedited data blocks. The user multiplies ΔD by B and generates the modified or updated both data vector and parity vector. By using homomorphic construction update to new data block without any hindrance to the old data makes token verification very easy. The seed is used for the updating the new edited data blocks; it blends with secret parity matrix and this helps to tract which version, as of how many updates have been done, is explained in the below algorithm 3.

Algorithm 3:

Assume the data block D_{ij} changed to ΔD_{ij}

1.Procedure :Update
2.If (update == append /modification).
3.For i to t do.
4.Derive x and y, where $x = f_{kSRF}$ and f_{Ksrp} (i).
5.For each vector calculate $G^{(j)}$ $j \longleftarrow 1$ to r do.
6.Calculate the Token T with i and j ,
$T^{(j)_i} \leftarrow T^{(j)_I} + x * \Delta G^j[I_s]$
7.End for
8.End for
9.Else if (update == delete)
10.For $i = 0$ to r do
11.if (i(l)== j) then
12.$T^{(j)_i} \leftarrow T^{(j)_I} + x * \Delta G^j[I_s]$ (D block)
13.End for
14.Store $T^{(j)_I}$ locally
15.Version v_i is updated
16.End procedure.

5.2 Delete

The stored data block can be de-allocated, if it is no longer needed from the cloud storage. The blocks are replaced with zeros or reserved data symbols .This also a kind of update operation where zeros are placed for which original data is to be deleted. The same method is employed for all deleted blocks like blinding the parity information.

5.3 Append

To add more data to the end of the existing information is append operation. In the cloud computing, the storage system, append the data frequently by uploading large number of data blocks at one time. In the matrix$D$ the distribution of the file is prepared by placing the appended file at the end of the file by concatenating rows at the bottom for data file D. Zero padding is used to create a row of m, again secret matrix is calculated directly by the user for appended block, the new block appended also stands for the integrity challenge by attending slight change in token pre-computing explained in algorithm for dynamic operation mentioned above.

5.4 Insertion

By adding a new file, in the desired index position in the data block without disturbing the other data blocks which is already stored in cloud server is executed in the insertion method. A block is inserted $D(j)$ which shifts the blocks with one step ahead after insertion $j+1$ .All the rows after the new data is affected by rearranging matrix and renumbering for each block shift. With challenge-response protocol and token computing are done for new positions. Supporting insertion is quite difficult, but certain studies enhance with hash trees like Merkle Hashing tree (MHT)[15],[16],[32] to extract block information. Additional data structure ensures physical block-index mapping where all block insertion is considered as append operation which is efficient. The only drawback it is to maintain the whole data structure information in user's local server.

6 Analyses and Performance Evaluation

Our Goal is to ensure efficient correctness, errorless and highly availability. Based on adversary model, security is analyzed and its performance evaluation is calculated based on file distribution with pre-computed and verification of its integrity as illustrated in Figure 2 and 3. The verification procedure ensures the correctness of data by employing challenge-response protocol. The responses from the servers which are challenged by the user are obtained, and blinded values are eradicated from them to calculate the user token which is previously computed. If it matches then the data originality is maintained or it has been prone to risk. Determination of the misbehaving server or error is identified and recovered. Sampling is done by selecting a set of rows in which challenge response protocol is executed. The computation cost is greatly reduced in the server when data maintains high detecting probability for corruptions.

Our sampled data is checked to identify the attacks of high probability. Determination of which server is malfunction is done during the modification and this is found by comparing the tokens stored along data file with the response from various servers. For the false negative result, it calculated by the product of matching the probability of complementary event and probability for sampling check.

6.1 Worst-case scenario

During the file distribution, the parity-block blinding is done to handle the worst-case scenario in the adversary

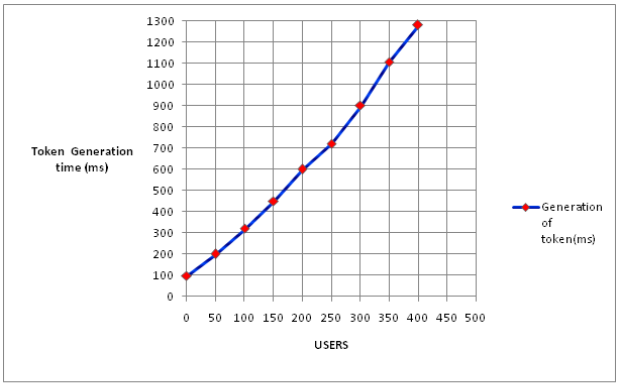

Fig. 2: Token generation Time

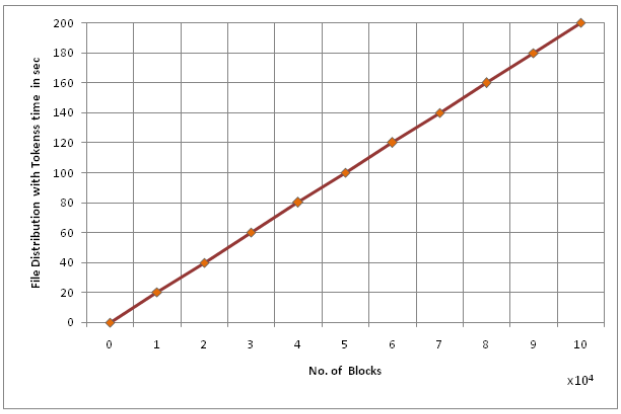

Fig. 3: File Distribution tagged with tokens in blocks

model. The redundancy-parity vectors are calculated by the product of data field and P secret matrix for storage assurance. Without blinding the data files with tokens, the intruders, cloud service, third-party auditor have the chance of interfering and reconstructing the original matrix from vector tokens. By selecting the same rows in the data file in the data block in the server, the parity which solve the set from Galois field and the set of$(s.t)$ by linear equation with entry of the parity generation P. The malicious server with the parity generates, the whole corresponding blocks. By adding noise to the linear equation, it is very difficult to solve the secret matrix to make the computation infeasible. The misbehaving servers have no enough information to access secret matrix P.

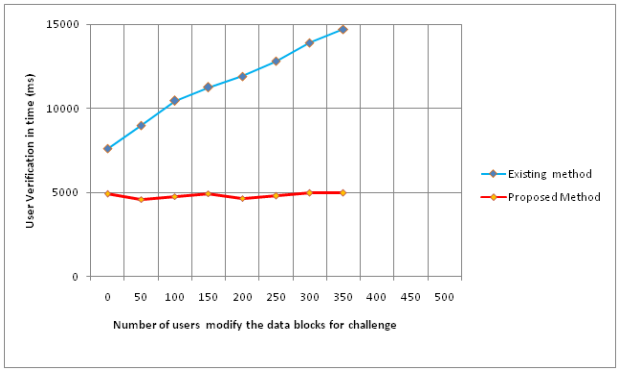

Fig. 4: User verification time on different users

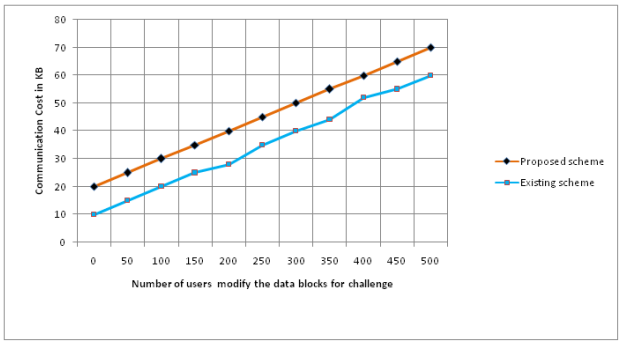

Fig. 5: Communication cost on different users

6.2 Performance

TThe performance depends on the audit mechanism .The cost of distribution file and generation of token determine its performance .Open source is used to implement the algorithm erasure coding library[29] with 30 trials. The distribution of files generally includes parity vector generation then it blinds the parity vector. The Two parameters (s,t) are involved in Reed Solomon encoding in Galois set. The determination of the parity vector required before outsourcing the data in to cloud server .As the t is incremented, the generation of parity vector increases, hence to estimate how many parity required to blinds clearly shows the generation of PRF by HMAC [30] which is implemented to minimize the cost and improved performance as shown in (Figure 4,5 and 6)

Compared to the entire predecessor scheme, the proposed scheme efficiently handles the data integrity, error finding and handles it for both static and dynamic data files in the cloud storage. Good balance is maintained for data dynamic and error recovery. Communication and computational cost complexity are very low as the number of verification tokens is fixed before conducting distribution for faster performance using Horner algorithms [26], the token is calculated from previous archive data.

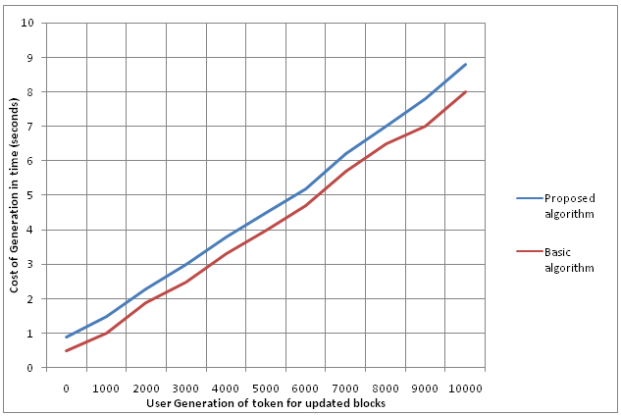

Fig. 6: Cost of User regeneration of tokens for updates data blocks

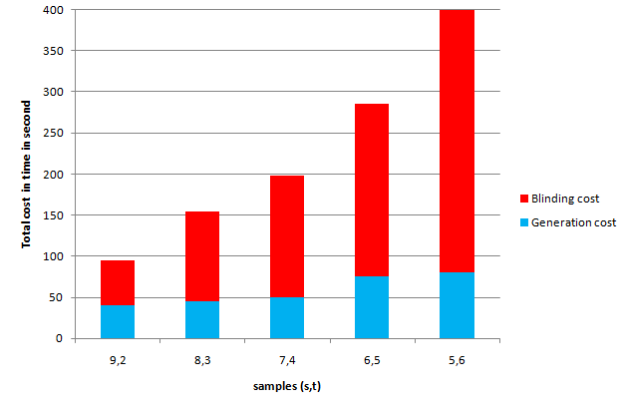

Fig. 7: Comparison between two parameters setting for file distribution.The chosen (s,t) parameters done by reed Solomon coding. Sample (20,4) files are divided into 20 data vectors and generate 4 redundant parity vector s is fixed and t is decreasing

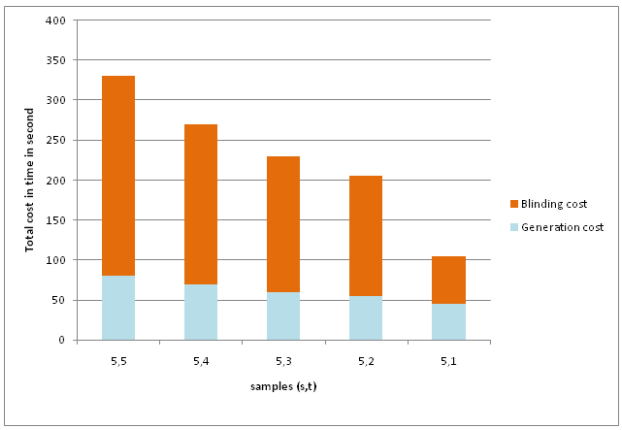

Fig. 8: Comparison between two parameters setting for file distribution. the chosen (s,t) parameters done by reed Solomon coding. Sample (20,4) files are divided into 20 data vectors and generate 4 redundant parity vector both are fixed $(s+t)$

7 Conclusion

In this proposed work, the storage verification design ensures integrity of data stored in cloud server and the privacy is preserved from third- party auditor, investigating the possible security breaches in cloud data storage effectively. The dependence of the data with high quality and availability is achieved effectively. The dynamic operation for distributed storage is like to add, update, append and erase. Erasure -correcting code in distribution of files gives us redundancy parity vector which solves the error localization and recovering lost data which guarantees our data integrity and availability with low computation and communication overhead. The time taken for the computation and the resources utilized which may be a burden to the online users to overcome this audit by third- party is done, where the user can delegate the checking process safely. With the experimental result, the proposed work shows high performance than the previous work with secured storage, durable-to-byzantine failures, and external and internal threats along with data dynamics.

Compliance with ethical standards

Conflict of Interest: All the authors do not have any conflict of interest in publishing this research article.
Ethical approval: This article does not contain any studies with human participants or animals performed by any of the authors.

References

[1] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z.Peterson, and D. Song, Provable Data Possession at Untrusted Stores, Proc. 14th ACM Conf. Computer and Comm. Security(CCS 07), 598-609 (2007).

[2] M.A. Shah, R. Swaminathan and M. Baker, Privacy-Preserving Audit and Extraction of Digital Contents, Cryptologye Print Archive, Report 2008/186 (2008).

[3] A. Juels and B.S. Kaliski Jr., Pors: Proofs of Retrievability for Large Files, Proc. 14th ACM Conf. Computer and Comm.Security (CCS 07), 584-597 (2007).

[4] H. Shacham and B. Waters, Compact Proofs of Retrievability, Proc. 14th Intl Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology(ASIACRYPT 08), 90-107 (2008).

[5] K.D. Bowers, A. Juels and A. Oprea, Proofs of Retrievability: Theory and Implementation, Report 2008/175, Cryptology ePrint Archive,(2008).

[6] M. Naor and G.N. Rothblum, The Complexity of Online Memory Checking, Proc. 46th Ann. IEEE Symp. Foundations of Computer Science (FOCS 05), 573-584 (2005).

[7] E.-C. Chang and J. Xu, Remote Integrity Check with Dishonest Storage Server, Proc. 13th European Symp. Research in Computer Security (ESORICS 08), 223-237 (2008).

[8] M.A. Shah, R. Swaminathan and M. Baker, Privacy-Preserving Audit and Extraction of Digital Contents, Report 2008/186, Cryptology ePrint Archive, (2008).

[9] C. Wang, S.S.M. Chow and Q. Wang, Privacy-Preserving Public Auditing for Secure Cloud Storage[J], IEEE Transactionson Computers,**62**(2), 362-375 (2013).

[10] M. Bellare, R.Canetti and H.Krawczyk, Keying hash functions for message authentication. Adv. Cryptol, **1109**,1-15, Springer LNCS (1996).

[11] B. Dan, L.Ben and S.Hovav, Short signatures from the Weil pairing. Adv. Cryptol, **2248**,514-532, Springer LNCS,(2011).

[12] G. Ateniese, S. Kamara and J. Katz, Proofs of storage from homomorphic identification protocols. Adv. Cryptol. **5912**,319-333, Springer LNCS, (2009).

[13] A. Shoufan and N.Huber, A fast hash tree generator for Merkle signature scheme. International symposium on circuits and systems, (2010).

[14] G. Ateniese, R. Burns and R. Curtmola, Provable data possession at untrusted stores. In: Proc. 14th ACM conf. computer and comm. security (CCS'07), 598-609 (2007).

[15] A. Juels, Jr. Kaliski and BS. PoRs, Proofs of retrievability for large files. In: Proc. 14th ACM conf. computer and comm. security (CCS'07),584-597 (2007).

[16] L. Jin,T. Xiao and C. Xiaofeng, An efficient proof of retrievability with public auditing in cloud computing. In: International conference on intelligent networking and collaborative systems IEEE, 93-98 (2013).

[17] Y. Zhu, H.Hu and G.J Ahn, Zero-knowledge proofs of retrievability, Sci. China. Inform. Sci., **54**,1608-1617 (2011).

[18] C. Kupcu Erway, C. Papamanthou and R. Tamassia, Dynamic provable data possession. In: Proc. 16th ACM conf. computer and comm. security (CCS'09) (2009)

[19] Q. Wang , C. Wang and K.Ren, Enabling public auditability and data dynamics for storage security in cloud computing, IEEE Trans. Parallel Distrib. Syst., **22**, 847-859 (2011).

[20] K.Yang, X.Jia, An efficient and secure dynamic auditing protocol for data storage in cloud. IEEE Trans. Parallel Distrib. Syst., **24**, 1717-1726 (2013).

[21] T. Schwarz and E.L. Miller, Store, Forget and Check: Using Algebraic Signatures to Check Remotely Administered Storage, Proc. IEEE Int'l Conf. Distributed Computing Systems (ICDCS '06), 12-12 (2006).

[22] J. Hendricks, G. Ganger and M. Reiter, Verifying Distributed Erasure-Coded Data, Proc. 26th ACM Symp. Principles of Distributed Computing, 139-146 (2007).

[23] L. Carter and M. Wegman, Universal Hash Functions, J. Computer and System Sciences, **18**, 2, 143-154 (1979).

[24] J.S. Plank and Y. Ding, Note: Correction to the 1997 Tutorial on Reed-Solomon Coding, Technical Report CS-03-504, Univ. of Tennessee,(2003).

[25] K.D. Bowers, A. Juels and A. Oprea, HAIL: A High-Availability and Integrity Layer for Cloud Storage, Proc. ACM Conf. Computer and Comm. Security (CCS'09), 187-198 (2009).

[26] T. Schwarz and E.L. Miller, Store, Forget, and Check: Using Algebraic Signatures to Check Remotely Administered Storage, Proc. IEEE Int'l Conf. Distributed Computing Systems (ICDCS'06), 12-12 (2006).

[27] C. Wang, Q. Wang, K. Ren and W. Lou, Privacy-Preserving Public Auditing for Storage Security in Cloud Computing, Proc.IEEE INFOCOM, (2010).

[28] C. Wang, K. Ren, W. Lou and J. Li, Towards Publicly Auditable Secure Cloud Data Storage Services, IEEE Network Magazine, **24**, 4, 19-24 (2010).

[29] J.S. Plank, S. Simmerman and C.D. Schuman, Jerasure: A Library in C/C++ Facilitating Erasure Coding for Storage Applications-Version 1.2, Technical Report CS-08-627, Univ. of Tennessee, (2008).

[30] M. Bellare, R. Canetti and H. Krawczyk, Keying Hash Functions for Message Authentication, Proc. 16th Ann. Int'l Cryptology Conf. Advances in Cryptology (Crypto'96), 1-15 (1996).

[31] M. Castro and B. Liskov, Practical Byzantine Fault Tolerance and Proactive Recovery, ACM Trans. Computer Systems, **20**,4, 398-461 (2002).

[32] R.C. Merkle, Protocols for Public Key Cryptosystems, Proc. IEEE Symp. Security and Privacy, (1980).

[33] Q. Wang, K. Ren, W. Lou and Y. Zhang, Dependable and Secure Sensor Data Storage with Dynamic Integrity Assurance, Proc. IEEE INFOCOM, (2009).

[34] M. Bellare, O. Goldreich and S. Goldwasser, Incremental Cryptography: The Case of Hashing and Signing, Proc. 14th Ann. Int'l Cryptology Conf. Advances in Cryptology (CRYPTO'94), 216-233 (1994).

[35] D.L.G. Filho and P.S.L.M. Barreto, Demonstrating Data Possession and Uncheatable Data Transfer, Cryptology ePrint Archive, Report 2006/150, (2006), http://eprint.iacr.org

R. Nithiavathy received the B.E. Computer Science and Engineering and M.E. Computer Science and Engineering degree from Anna University, Chennai. She is presently working as an Assistant Professor in the Department of Computer Science and Engineering, Arjun College of technology, affiliated to Anna University- Chennai, TamilNadu, India. His research area includes cloud computing storage, security.

K. Srihari received the M.E. and Ph.D. degree from Anna University, Chennai. He is currently working as an Associate Professor in the Department of Computer Science and Engineering, SNS College of Technology, affiliated to Anna University-Chennai, Tamilnadu, India. Dr.K.Srihari published over 30 papers in international journals and his research area includes semantic search engines, big data and cloud computing.

S. Karthik is presently Professor and Dean in the Department of Computer Science and Engineering, SNS College of Technology, affiliated to Anna University-Chennai, Tamilnadu, India. He received the M.E. and Ph.D. degree from Anna University, Chennai. His research interests include network security, big data, cloud computing, web services and wireless systems. Dr.S. Karthik published more than 96 papers in refereed international journals and 125 papers in conferences and has been involved many international conferences as Technical Chair and tutorial presenter. He is an active member of IEEE, ISTE, IAENG, IACSIT and Indian Computer Society.