# A NOVEL EFFICIENT REMOTE DATA POSSESSION CHECKING PROTOCOL IN CLOUD STORAGES

**[1] DR. CH. NARASHIMA CHARY, [2]R. VINOD KUMAR, [3]G. MANASA**

*[1]Associate Professor, Department of Computer Science and Engineering,*
*[2, 3] Assistant Professor, Department of Computer Science and Engineering, Sri Indu College of Engineering and Technology, Hyderabad, Telangana-501510*

## ABSTRACT:

*Cloud computing is widely accepted and used in many real applications because of its reliable, scalability and low cost outsources storage services. The service providers may hide some misbehavior from data owner to maintain a good reputation. Remote data possession checking is an effective technique to ensure the integrity for data files stored on cloud servers which uses hash value generated using homomorphic hash function to verify the data integrity and operation record table to monitor the data modification and changes by the data owners and differentiate them from unauthorized data tampering and data removal done by cloud servers as a point of misbehavior. We use public auditors for data possession checking because user validating the integrity adds extra workload and maintenance to them. We have many public auditors in existing where user privacy may compromise when user provides the file index to auditor for validation so we encrypt the hash value and store in cloud, during validation the auditor is provided with only old and new hash values because the hash value is encrypted. Because cloud can't make any changes and auditor can't get the file index so privacy is preserved.*

***Keywords:* *Remote Data Possession Checking, homomorphic hash function, operation record table, integrity, validation***

## INTRODUCTION

Cloud storage has become a part of collecting the number of information for millions of users and organizations. Large number of users attracted to use the cloud storage service for storing their data. Cloud services providers promise users for providing a reliable, scalable and low cost outsourced storage services. The important challenges are data tampering and data lost still exist in cloud storage [1]. Every cloud service provider tries to provide a quality of service to their client for data storage but these services still suffer various security issues like data tampering, data loss of outsourced files. In this case the service providers may hide these kinds of misbehaviors from data owner to maintain a good reputation. Therefore, it is crucial for the data owner to utilize an efficient way to check the integrity for outsourced data. We need a validation process to check the quality of service provided by the cloud service providers [2].

In existing system, validates data integrity of files stored in cloud without retrieving it using file properties and attributes. For example, an index is created which hold the file attributes like file name, date of creation, date of modification etc. This index is checked with the current file attributes to validate its integrity [3]. In multi cloud environments, data replication feature is used as a validation factor where the replica is used to compare to validate the integrity. In this method, Some of the disadvantages are

1. Validation based on file attributes, user can't find any modifications.

2. Using replica to validate integrity is very complex. It is possible only in multi cloud environment.

3. Other complex schemes provide data integrity check but they didn't allow the dynamic updating process.

## SYSTEM ANALYSIS:

### Proposed System:

This proposed scheme uses hash function to create a hash value generated based on the file content. These hash values are used to validate integrity. Validation process compares the previous hash value of the file with current hash value of the file. If the hash values mismatch then the data is said to be tampered or corrupted, else the data is safe. To allow authentic user dynamic file updating use an operation record table which record the operation, if file is updated then the previous hash value is updated automatically [4].

### ADVANTAGES:

1. Hash value based integrity check is effective in validation, when a small bit of information inside the file changes, hash value also gets changes.

2. It allows dynamic content updating while validating the file in cloud.

3. It is a simple process to reduce the load of integrity checking process.

### System Architecture:

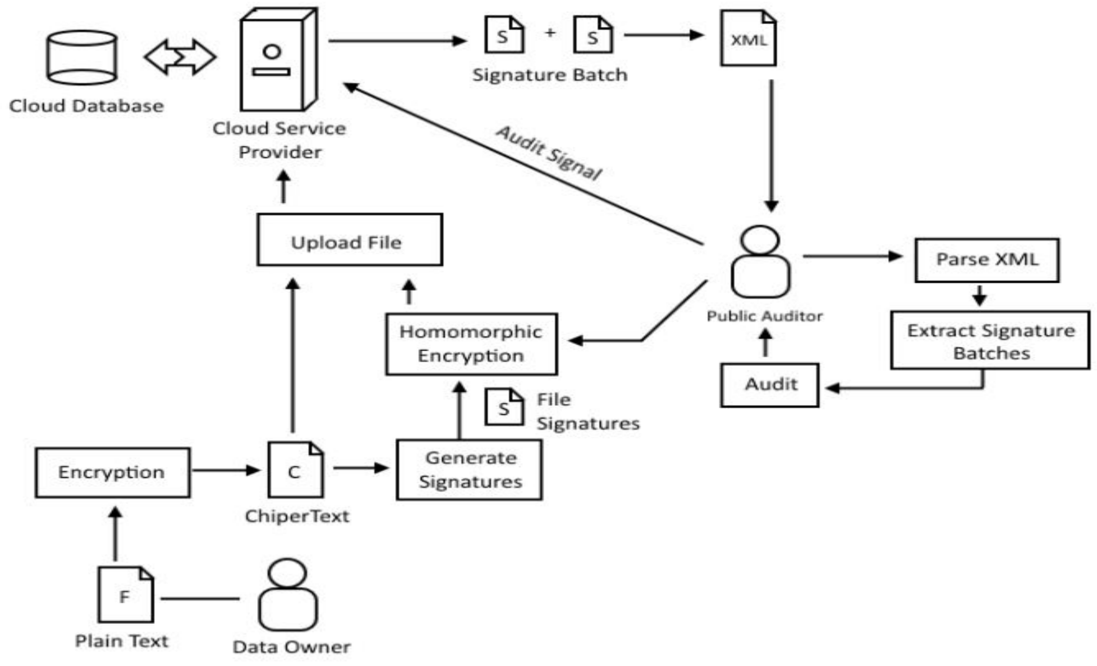

**Figure.1 ARCHITECTURE DIAGRAM**

## MODULE DESCRIPTION:

### Symmetric Encryption:

Due to privacy issues, user needs to encrypt their content before uploading them to the cloud server. We have to use Symmetric encryption scheme because it is cost effective when compared to asymmetric encryption standard. There is no need for distribution of symmetric key, only user need to access the file so he holds the key. After encryption, user needs to generate signature for integrity validation which is done by the next module [5].

## Signature Generation:

This module used to compute file signature which is 32 character length unique code representing the document using SHA-3 algorithm [18]. A file identifier is included within the signature to identifying the file during the validation process. Then store the signatures in cloud server to ensure privacy and allow access to encrypt signature using auditor's public key for public auditor. After encryption, the signature and the file are uploaded to the cloud server [6].

## Key Generation:

Key generation module is executed by public auditor because needs to forward the encryption key using RSA and send it to users. For secure key, distribution key allows to encrypt by the public auditor and forward to user end. A key copy is stored in the auditor end which will be used during the proof verification process [7].

## Proof Generation:

Public auditor sends an audit challenge to the cloud server to generate an auditing proof for each shared data. Cloud server runs the proof generation algorithm and sends the audit proof to the public auditor. Proof contains old signature which is encrypted by the auditor's key and a new signature of the file [8].

## Proof Verification:

The public auditor audits the integrity of shared data by verifying the proof. The correctness of the scheme can be proved by verifying the correctness of equation (1) as

*User Generated Block Signatures = Cloud Generated Block Signatures* (1)

If (1) holds, then the public auditor believes that the integrity of blocks in shared data is correct. Otherwise, the integrity of blocks in shared data is incorrect.

## METHODOLOGY:

### Homomorphic Hash Function:

Our scheme adopts the homomorphic hash function defined as the basis, which is described as following:
First, the algorithm HKeyGen(P,Q,m,s) $\rightarrow$ K is utilized to obtain the homomorphic key. It takes four security parameters as inputs, in which P and Q are two discrete log security parameters, m is the sector count of the message and s is a random seed. It outputs the homomorphic key K = (bp,bq,g) , where bp and bq are two random big primes with the properties of |bp| = P , |bq| = Q and bq | (bp-1) , g $\rightarrow$ [g1,g2,….,gm ] is a 1×m row-vector composing of m random values in $Zp^*$ with order q . The detailed process of this algorithm is shown in the below pseudocode, in which the function f (x) is the pseudo-random number generator with seed s and outputs the next number in its pseudo-random sequence, scaled to the range {0,….,x-1}.
The computation of the hash value X of the message S represented by H(S) $\rightarrow$ X is defined as following [9].
S is divided into m sectors S = (s1,s2,…,sm) , the hash value is calculated as:

$$H_K(S) = \Pi\, g_i^{si} \bmod gp.$$

For any two messages Si and Sj , where Si = ($s_{1i}$ ,$s_{2i}$ ,….,$s_{mi}$ ) , Sj = ($s_{1j}$ ,$s_{2j}$ ,….,$s_{mj}$ ) , we define

$$Si + Sj = (s_{1i} + s_{1j}, s_{2i} + s_{2j}, \ldots, s_{mi} + s_{mj}) \bmod gq$$

The homomorphic property of H(.) can be verified by:

$$H(S_i+S_j) = \Pi\, g_t^{sti + stj} \bmod gp = g_t^{sti} \bmod gp \,.\, g_t^{stj} \bmod gp = H(Si)\,.\,H(Sj) \text{ for } (1<=t<=m).$$

## Homomorphic Key Generation Algorithms

```
Function HKeyGen (P,Q,m,s)
    do
            gq ← gqGen(Q)
            gp ← gpGen(gq,P)
    while gp=0 done
    for i=1 to m do
            do
                    x ← f(gp-1)+1
                    $g_i$ ← x $^{(p-1)/q}$ (mod gp)
            while  $g_i$ = 1 done
    done
return (gp,gq,g')

Function gqGen(Q)
do
    gq ← f($2^Q$)
while gq is not prime done
return gq

Function gqGen(gq,P)
for i=1 to 4P do
    x ← f($2^P$)
    c ← X(mod 2gq)
    gp ← X-c+1        // note: gp ≡ 1(mod 2gq)
if gp is prime then return Pf done
return 0
```

## Operation Record Table

To support dynamic operations on file blocks, we introduce a simple flexible data structure named operation record table (ORT). The table is reserved on the data owner side and used to record all the dynamic behaviors on file blocks. ORT has a simple structure with only three columns, that is Block Position(BP) , Block Index(BI) and Block Version (BV). The BP represents the physical index for the current block in the file, normally its value is incremented by 1. The BI represents the logical index for the current block, which is not necessary equal to BP but relevant with the time when the block appears in the file. The BV indicates the current version for the block [17]. If the data file is initially created, the BV values for all blocks are 1. When one concrete block is updated, its BV value is incremented by 1. It is noted that using the ORT table will increase the storage overhead of the data owner by O(n), where n is the count of blocks [10]. However, this extra storage cost is very little. For example, a 1GB-file with 16KB block size only needs 512KB space to store an ORT realized by linked list (< 0.05% of the file size).

## RDPC (Remote Data Possession Checking) Scheme:

The cloud storage system including two participants: Cloud Storage Server (CSS) and data owner. The CSS has powerful storage ability and computation resources, it accepts the data owner's requests to store the outsourced data files and supplies access service [16]. The data owner enjoys CSS's service and puts large amount of files to CSS without backup copies in local. As the CSS is not assumed to be trustable and occasionally misbehave, for example, modifying or deleting partial data files, the data owner can check the integrity for the outsourced data efficiently [11].

A RDPC scheme includes the following seven algorithms:

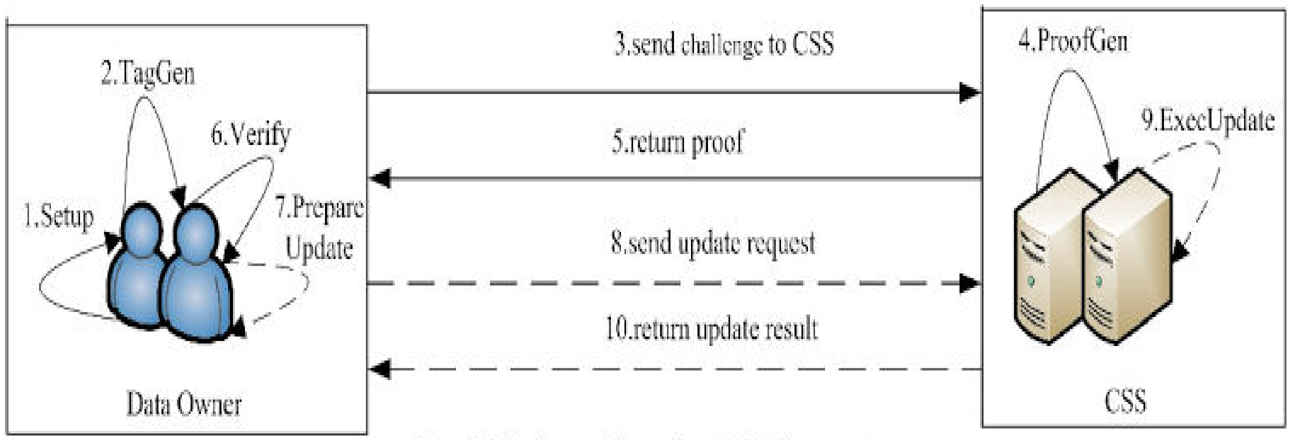

Fig. 2. Work procedure of our RDPC protocol

**KeyGen (1k,P,Q,m,s) → (K,Pk):** The data owner executes this algorithm to initialize the system and generate keys. It inputs security parameters k, P, Q, the message sector number m and a random seed s , and outputs the homomorphism key K and private key Pk. Here the seed s serves as a heuristic "proof", which the hash parameters are selected truthfully [12].

**TagGen(K,Pk,F) → T**: This algorithm is executed by the data owner to produce tags of the file. It inputs the homomorphic key K , private key Pk and file F , and outputs the tag set T which is a sequential collection for tag of each block [13].

**Challenge(c) → chal:** The data owner executes the algorithm to generate the challenge information. It takes the challenged blocks count c as input and outputs the challenge chal .

**ProofGen(F,T,chal) → P:** The CSS executes this algorithm to generate the integrity proof P . It inputs the file F , tag set T and the challenge chal and outputs the proof P .

**Verify(K,Pk,chal,Pf) → {1,0}**: The data owner executes the algorithm to check the integrity of the file using the proof P returned from CSS. It takes homomorphism key K, private key Pk, challenge chal and proof Pf as inputs, and outputs 1 if P is correct, otherwise it outputs 0 .

**PrepareUpdate(Fi,i,UT) → URI**: The data owner runs this algorithm to prepare dynamic data operations on data blocks. It takes new file block Fi , the block position i and the update type UT as inputs, and outputs the update request information URI . The parameter UT has three optional elements: insert, modify and delete [14].

**ExecUpdate(URI) → {Success,Fail}:** The CSS runs this algorithm to execute the update operation. It inputs URI and outputs execution result. If the update operation is finished successfully, it returns Success , otherwise returns Fail .

The complete work procedure of RDPC protocol is illustrated in the figure 2, in which solid lines and dash lines represent the processes of data integrity checking and data dynamic operations respectively [15].

## CONCLUSION:

In this paper, we study the issue for integrity checking of data files outsourced to remote server and propose an efficient secure RDPC protocol with data dynamic. Our remote data possession checking protocol to access the files in cloud environment securely with the help of cloud storage server and data owner. In addition to that, only some auditors are permitted with authorized signature and key generation to check the modification of files such as deleting, changing and steal the data by anyone, it can be notified by using homomorphic hash function and operation record table. So, it will provide good quality of service by provider in the cloud environment, now the non trusted environment provide efficient as trusted environment.

## REFERENCES:

*[1] F. Sebé, J. Domingo-Ferrer, A. Martinez-balleste, Y. Deswarte, and J. Quisquater, "Efficient Remote Data Possession Checking in Critical Information Infrastructures," IEEE Trans. Knowledge and Data Eng., vol. 20, no. 8, pp. 1034-1038, Aug. 2008.*

*[2] H. Wang, ''Identity-Based distributed provable data possession in Multicloud storage,'' IEEE Trans. Service Comput., vol. 8, no. 2, pp. 328-340, 2015.*

*[3] G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, ''Scalable and Efficient Provable Data Possession,'' in Proc. 4th Int'l Conf. Security and Privacy in Commun. Netw. (SecureComm), 2008, pp. 1-10.*

*[4] K. Yang and X. Jia, ''An efficient and secure dynamic auditing protocol for data storage in cloud computing,'' IEEE Trans. Parallel Distrib. Syst., vol. 24, no. 9, pp. 1717-1726, 2013.*

*[5] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic,"Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," Future Gener. Comp. Sy., vol.25, no. 6, pp. 599 – 616, 2009.*

*[6] H. Qian, J. Li, Y. Zhang and J. Han, "Privacy preserving personal health record using multi-authority attribute-based encryption with revocation," Int. J. Inf. Secur., vol. 14, no. 6, pp. 487-497, 2015.*

*[7] J. Li, W. Yao, Y. Zhang, H. Qian and J. Han, "Flexible and fine-grained attribute-based data storage in cloud computing," IEEE Trans. Service Comput., DOI: 10.1109/TSC.2016.2520932.*

*[8] J. Li, X. Lin, Y. Zhang and J. Han, "KSF-OABE: outsourced attribute-based encryption with keyword search function for cloud storage," IEEE Trans. Service Comput., DOI: 10.1109/TSC.2016.2542813.*

*[9] K. Bhargavi. An Effective Study on Data Science Approach to Cybercrime Underground Economy Data. Journal of Engineering, Computing and Architecture.2020;p.148.*

*[10]M. Kiran Kumar , S. Jessica Saritha. AN EFFICIENT APPROACH TO QUERY REFORMULATION IN WEB SEARCH, International Journal of Research in Engineering and Technology. 2015;p.172*

*[11] K BALAKRISHNA,M NAGA SESHUDU,A SANDEEP. Providing Privacy for Numeric Range SQL Queries Using Two-Cloud Architecture. International Journal of Scientific Research and Review. 2018;p.39*

*[12] K BALA KRISHNA, M NAGASESHUDU. An Effective Way of Processing Big Data by Using Hierarchically Distributed Data Matrix. International Journal of Research.2019;p.1628*

*[13] P.Padma, Vadapalli Gopi,. Detection of Cyber anomaly Using Fuzzy Neural networks. Journal of Engineering Sciences.2020;p.48.*

*[14] Kiran Kumar, M., Kranthi Kumar, S., Kalpana, E., Srikanth, D., & Saikumar, K. (2022). A Novel Implementation of Linux Based Android Platform for Client and Server. In A Fusion of Artificial Intelligence and Internet of Things for Emerging Cyber Systems (pp. 151-170). Springer, Cham.*

*[15] Kumar, M. Kiran, and Pankaj Kawad Kar. "A Study on Privacy Preserving in Big Data Mining Using Fuzzy Logic Approach." Turkish Journal of Computer and Mathematics Education (TURCOMAT) 11.3 (2020): 2108-2116.*

*[16] M. Kiran Kumar and Dr. Pankaj Kawad Kar. "Implementation of Novel Association Rule Hiding Algorithm Using FLA with Privacy Preserving in Big Data Mining". Design Engineering (2023): 15852-15862*

*[17] K. APARNA, G. MURALI. ANNOTATING SEARCH RESULTS FROM WEB DATABASE USING IN-TEXT PREFIX/SUFFIX ANNOTATOR, International Journal of Research in Engineering and Technology. 2015;p.16.*

*[18] J. Li, Y. Shi and Y. Zhang, "Searchable ciphertext-policy attribute-based encryption with revocation in cloud storage," Int. J. Commun. Syst., DOI: 10.1002/dac.2942.*