International Journal of Advanced Research in Engineering Technology & Science

ISSN: 2349-2819

Email: editor@ijarets.org Volume-8, Issue-6 June- 2021 www.ijarets.org

# MECHANISMS FOR CLOUD ACCESS CONTROL FOR SECURE STORAGE

**Kamlesh Kumar,**
Research Scholar, Dept. of CSE,
The Glocal University, Mirzapur Pole, Saharanpur (UP)

**Dr. Manoj Kumar,**
Associate Professor, Dept. of CSE,
The Glocal University, Mirzapur Pole, Saharanpur (UP)

## ABSTRACT
Emerging storage cloud systems provide continuously available and highly scalable storage services to millions of geographically distributed clients. A secure access control mechanism is a crucial prerequisite for allowing clients to entrust their data to such cloud services. The seamlessly unlimited scale of the cloud and the new usage scenarios that accompany it pose new challenges in the design of such access control systems. In this paper we present a capability-based access control model and architecture appropriate for cloud storage systems that is secure, flexible, and scalable. We introduce new functionalities such as a flexible and dynamic description of resources; an advanced delegation mechanism and support for auditability, accountability and access confinement. The paper details the secure access model, shows how it fits in scalable storage cloud architecture, and analyzes its security and performance.

***Keywords*** **:** *Cloud systems, storage, mechanism and support*

## INTRODUCTION
The rapid growth in the amount of personal and organizational digital data is a major characteristic of information systems in this decade. This growth is accompanied by increased demand for availability, as users wish to run their software and access their data, regardless of their type and location, from any web-enabled device at any time. Cloud computing addresses these challenges by providing virtualized resources as an online service and by allowing them to scale dynamically. Services offered over the cloud include applications, virtual machines (Infrastructure as a Service), and data storage (Storage as a Service). The latter service, along with the storage cloud upon which it resides, is the topic of this paper. Storage cloud systems are required to provide continuous availability and elastic scalability while serving multiple tenants accessing their data through the Internet. The storage cloud infrastructures are comprised of tens of geographically dispersed data centers (DCs), where each DC is comprised of many thousands of compute and storage nodes, and should be able to efficiently serve millions of clients sharing billions of objects. Furthermore, to achieve availability and scalability, each data object is replicated at multiple data centers across the cloud. Each object replica should be accessible for reads and writes, and to optimize access latency and data throughput, it is preferable that clients be directed to the replica closest to them.

## USER / CLIENT INTERFACE
Through a cloud server, the user interface communicates with the cloud temporal database. The user interface is used in this model to carry out user interaction. When a user submits a query, the system receives it and checks it using a validation agent. The user interface identifies queries from common users. The cloud databases manager, on the other hand, recognizes requests from malicious users and sends them to the temporal information manager

and constraint manager, who then check the key provided to them and filter and drop the queries if the key does not match.

**CLOUD DATABASE SERVER**

In this work, data storage and manipulation tasks are supported by a cloud database server. If there is any doubt about the user's history, the cloud database server receives queries from the user interface and sends them to the other constraint manager and temporal information manager for key verification. Before accessing a cloud database, these users must pass checks based on identity restrictions, time restrictions, and user status levels.

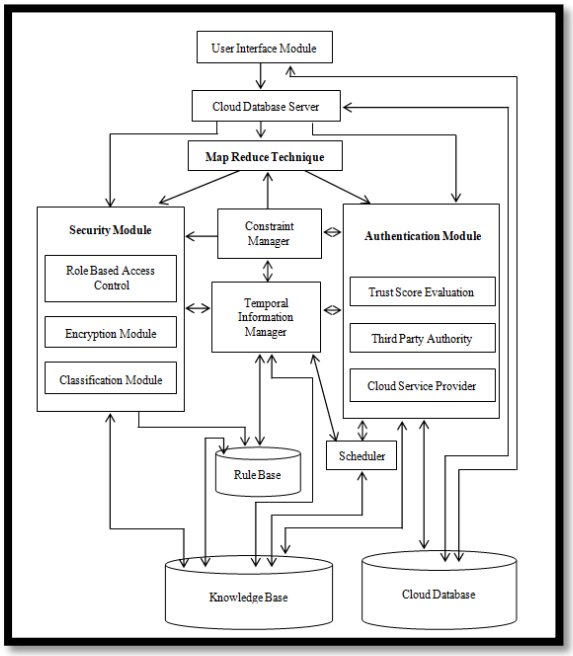


**Figure-1: System Architecture**

**MAP REDUCE TECHNIQUE**

Temporal Secured Cloud Map Reduced Algorithm (TSCMRA), a new algorithm, is used by this component to boost speed. In order to improve security, it is in charge of periodically altering the roles while taking time restrictions into account. It is also in charge of performing access control based on temporal restrictions through temporal sorting, temporal searching and indexing, temporal categorization, and temporal joining.

**SECURITY MODULE**

Role-based access control, encryption, and classification are the three submodules that make up this module. In the event that the user's history is questioned, the role-based access control sub module is in charge of Temporal Information Manager for key verification. Before accessing a Cloud database, such users are put through checks based on temporal limitations, identity constraints, and user status levels. The database is accessed by RBAC, and the user interface receives the query result. Reputable users receive status level 5, which allows them to advance directly to the Speed and Access Control Manager. Second, the database is effectively secured using the Hill Cipher Technique, which is employed in both the encryption and decryption procedures. The domain knowledge is kept in the rule base as IF...THEN statements.

**AUTHENTICATION MODULE**

The evaluation of trust scores, third party authorities, and cloud service providers are the three subcomponents that make up the authentication module. First, the newly proposed trust computation technique is used in this work to evaluate trust scores. It connects with trusted TPAs while also being able to do so with cloud service providers and temporal information managers. To calculate and uphold confidence for safe data, trusted TPA is utilized.

The cloud database uses the Temporal Information Manager to check for temporal constraints. Additionally, brand-new rules that intelligent agents pick up from user behavior are also encoded as rules and kept in the rule base.

This system uses a frame-based method of knowledge representation to store the whole collection of rules in the rule base.

**CONSTRAINT MANAGER**

The task of the power constraint manager is to employ temporal constraints to examine the user's historical access patterns and the length of each access. The power constraint manager uses the rule base, which contains a collection of active and passive rules as well as event information. Under the specified circumstances, active rules are automatically triggered in response to anomalous events. In addition to managing power, the power constraint manager is also in charge of managing roles, which includes adding, removing, and modifying rules in response to agent feedback.

**SCHEDULER**

The primary duties of the scheduler include creating user access schedules based on data from the knowledge base and inputs from the temporal information manager about the users.

**TEMPORAL INFORMATION MANAGER**

The key duties of the temporal manager include handling temporal restrictions and rules and providing details about user access patterns on the cloud database during the time period when data access attempts are being made. To assign appropriate roles based on user requirements and status, this manager consults the rule manager and temporal information manager. Additionally, it assigns user roles based on roles from the role base and is in charge of giving users different privileges on tables.

**RULE BASE**

In the form of rules, user knowledge is stored in the rule base. After being properly validated, the information learned from the experts on system circumstances is entered into the Rule Base. Additionally, brand-new rules that intelligent agents pick up from user behavior are also encoded as rules and kept in the rule base. For efficient construction and maintenance of the Rule Base in this system, the complete sets of rules are stored in the Rule Base utilizing a frame-based knowledge representation technique. Utilizing this knowledge representation has several benefits, but its main benefit is that it facilitates efficient retrieval from Cloud databases to improve performance and security.

**CLOUD DATABASE**

For data storage, a cloud database is linked to a cloud server. In this work, data in the form of organized, semi-structured, and unstructured data structures are stored in the cloud database. It utilizes different data formats in addition to Structured Query Language. Cloud data centers house the cloud data, which is organized into servers, racks, clusters, data bases, and files with unique identifiers. Utilizing map reduce functions, it may be handled efficiently.

**KNOWLEDGE BASE**

Many rules, data, and details regarding the previous history of user behaviors are available in the knowledge base. The knowledge base is in charge of formulating the rules and offers pertinent data regarding the user for the scheduler to use when scheduling their activities. Additionally, the authentication module provides the necessary data for making judgments on cloud users. Additionally, the security module is useful for protecting the information. Finally, it offers temporal information that can be used by a temporal information manager to make decisions about cloud user behaviors.

**Erasure codes**

Cauchy Reed Solomon code

Erasure codes are the fundamental method for preventing data loss in distributed storage systems.The several types of distributed storage include cloud storage (Wang et al. 2011), peer-to-peer storage (Zhengnet al. 2009), network applications, and data domains (Zhiet al. 2011).

The Vandermonde Reed-Solomon algorithm, in which m represents the original data block and n represents duplicate blocks stored in a spread fashion, was previously used by several communication and storage systems.

The only significant challenge is determining the n Galois Field product for each code in the block, which is smaller than the machine word; this can need up to 8 n multiplications per word, which is economically prohibitive for big data blocks (Planks 2005). With the minor upgrade of the Vandermonde Reed - Solomon code by (Blomeret al.1995), a Cauchy Reed- Solomon is included. The following two modifications are made: XOR is used in place of the Galois field GF (2w) encoded operation conversion. The matrix is utilized in place of the Galois Field GF (2w), which was employed in the Vandermonde matrix, as the second change. The distributed matrix generates invertible n x n matrices. Cauchy matrix with parity block is used to derive the distribution matrix A.

$$A = (I/P)^T = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1/(p_{1,0}) & 1/(p_{1,1}) & 1/(p_{1,(m-1)}) \\ 1/(p_{2,0}) & 1/(p_{2,1}) & 1/(p_{2,(m-1)}) \\ 1/(p_{n-1,1}) & 1/(p_{n-1,1}) & 1/(p_{n-1,(m-1)}) \end{pmatrix}^T$$

**Figure-2: Cauchy Distribution Matrix**

Figure depicts the Cauchy matrix A, which consists of an identical matrix in the m columns and a (m x n) Cauchy distribution matrix in the n columns. The projection is used by the binary distribution matrix (BDM) to enlarge the Cauchy distribution matrix. The bit matrix is encoded by multiplying binary distribution matrix A by wm vector. When the files F and A are multiplied, the client creates the encoded file, where w(m+n) is a vector element.Cauchy reed-Solomon code uses a sequence of bit-wise XOR operations to encrypt data files, which is faster than the Vandermonde reed Solomon code. For the larger files, the aforementioned Cauchy Reed-Solomon code performs less well.

**TORNADO CODES**

The data file stored in the distributed files can be encoded and decoded using any erasure code. Effective block encoding is restricted and becomes sluggish for larger files. A Tornado code hence relies on the atypical bipartite graphs. The recovery of the entire file can be done at variable rates even with a portion (Ashish 2003). Assumed generator matrices are a series of irregular random bipartite graphs that are cascaded to produce tornado code. The original data is stored in the nodes on the left, and duplicate or redundant data is stored in the nodes on the right, which are XORed with the input nodes on the level to the left. The number of exclusive OR operations for encode and decode determines how many edges there are in the graph.

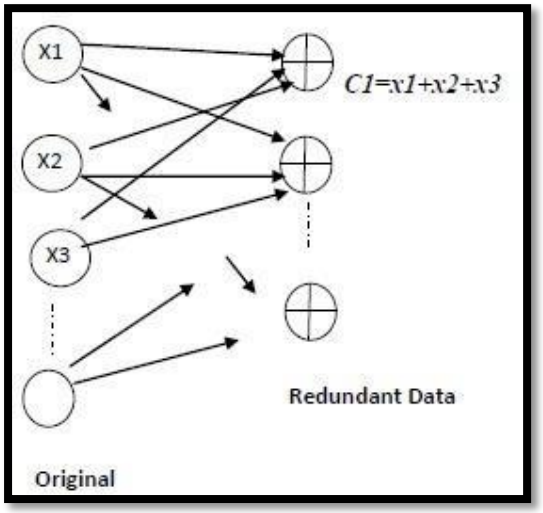


**Figure-3: Tornado Code Structure**

The code word contains the n message bits, and at each stage of the cascade, check bits are generated.

**SOBOL SEQUENCE**

A quasi-random sequence of points with low-star discrepancy known as the Sobol sequence was developed in 1967. From a set of binary fractions vi, where i=1, 2..w is referred to as a direction number, the sequence creates values that are binary fractions, that is, 0 and 1 with length of w bits. It uses base two for the numbers that are generated progressively in order to fill in the gaps between the uncorrelated sequences.Sobol sequence exhibits greater regularity than pseudorandom sequence.

**UNIVERSAL HASH FUNCTION (UHF)**

According to Carter et al. (1979), the universal hash function is a function that relies on the key where k K and compresses file elements and information messages into packaged into a digest or hash. When pairing messages between x and y with an independent element k, the function is resistant to collision. The likelihood that the third input matches the hash of the first two inputs for the given three input nearly exclusive OR universal (AXU) hash function feature is very low.

**HOMOMORPHIC DISTRIBUTED CHECK PROTOCOL (HDCP)**

Homomorphic distributed check protocol is used to ensure the consistency and availability of the files stored on cloud servers. There are three stages to it:

Set up

To maintain availability, the data files are encoded, and initially, prior to distribution, the owner creates compact verification tokens. During this phase, the following techniques are primarily used:

- Encode
- Key gen
- Meta data gen

Encode HDCP

Using an erasure code based on the Cauchy Reed Solomon (CRS) algorithm, the entire file is distributed redundantly over several servers (Blomeret al. 1992, Planks 2005). the collection of k=m+n cloud servers that guarantees data availability despite possible data loss and byzantine failure. In the Cauchy Reed-Solomon code, (m+ n, n) is the number of redundancy or parity blocks produced by the code from the file containing m data blocks.By using the m data block, m out of (m+n) can be rebuilt. The original data can be recovered from any m out of (m+n) failures without a loss, although there is a space overhead (m/n) mentioned in (Lillibridge et al. 2003) due to the storage of files across several servers.

***Algorithm Encode*** *Step 1: Procedure start*

*Step 2: for i=0 to n-1*
*Step 3: do*
*Step 4: for k=0 to w-1; do*
*Step 5:* $C^i k = C^i k = [0, 0]$
*Step 6: for j=0 to m-1 do*
*Step 7: for l=0 to w-1 do*
*Step 8:* $C^i k = C^i k \oplus fi^k$ *,j,*dj*
*Step 9: end for*
*Step 10: end for*
*Step11: end for*
*Step 12: end for*
*Step 13: end procedure*

The following assumptions are made: Let Di=(d1i,d2i,d1i) and F= [D1,D2,...,Dm]The data word (which may be 8 or 16) is T, W.Diis column vector, T is the abbreviation for transposition, and l stands for block vector size. All of the Galois field's elements can be obtained by merely multiplying F by the client-encoded file C.

C=F.A

Key Gen HDCP

After the data has been encrypted, the owner or client will generate a challenge key x and a master permutation key y at random in order to process the file later. To generate keys a and y, the Sobol random function (SRF) is utilized.

x= f (i) and y= f (i)

The key is calculated by key : f :$\{0,1\}^* \times$ key- $GF(2^w)$

*Algorithm*
*Step 1: procedure key Gen*
*Step 2: generation of any random change key Ksrf*
*Step 3: master permutation key is generated by sobol sequence;*
*Step 4: x and y are derived*
*Step 5: end for*
*Step6: end procedure.*

Metadata Generation HDCP

All protocols rely on the metadata because it is one of the most crucial components in preserving the integrity of data in the cloud. Before sending the file to the server, the client pre-calculates the token. On each individual data block C(k) employing the Sobol sequence, there are brief verification tokens referred to as meta data. For each token cover, a random set of blocks is used.

Algorithm Metadata Gen
*Step 1: Procedure*
*Step 2: for vector G(k), k←1,n do*
*Step 3:for round i←1, t do*
*Step 4 : compute random indices Iq*
*Step 5: Compute the meta data* $V^i j$.
*Step 6: end for*
*Step 7: end for*
*Step 8: store all* $V_i$ *locally*
*Step 9: End procedure.*

Using the universal Hash function (UHF) is employed to preserve the homomorphic properties which incorporated with the erasure code.

Check phase
To ensure security, the client, owner, or TPA frequently verifies the server for data integrity. It also guarantees file recovery and detects error and data loss. Three techniques are used to elaborate on this check phase. A) contest HDCP, B) response, and C) verify authenticity
Challenge: HDCP
To check the accuracy of the data on the cloud server, the client sends out a random sample challenge. The master key and challenge key are permutated using the Sobol sequence to generate a challenge key, which is then sent to the service provider.
Algorithm challenge: HDCP
*Step1: Procedure challenge*
*Step 2: Regeneration of the random challenge key* $K_{SRF}$ *and permuted masterkey* $K_{SRP}$
*Step 3: derive x and y*
*Step 4: end procedure.*
Response: HDCP
After receiving a request from the client, the server will respond by computing the integrity proof signatures on a random block and sending them to the client.In the following technique, the CSP computes the answer for each block during the ith response algorithm process check over the k servers.
Algorithm Response: HDCP
*Step 1: Procedure Check Integrity*
*Step 2: for j←1, k, do*
*Step 3: if (Response= pre computed token) then*
*Step 4: Ready for next challenge*
*Step 5: else*
*Step 6: return the data corrupt*
*Step 7 end if*
*Step 8: end for*
*Step 9 end procedures.*
If the data's originality is confirmed, HDCP will take on the following new task.
Check Originality: HDCP
The client verifies the accuracy of the data by comparing the server's answer with the meta data that the data's owner has already computed for verification.
*Algorithm for check originality: HDCP*
*Step 1: Procedure: Check Originality*
*Step 2: for j←1, k do*
*Step 3: if (Response = metadata) then*
*Step 4: Accept and ready for new challenge*
*Step 5: else*
*Step 6: Return*
*Step 7: End if*
*Step 8: end for*
*Step 9: End procedure*
If the data integrity is upheld, HDCP is prepared for any new challenges or for when an error or loss needs to be corrected. In the following phase, this protocol's dynamic data operation is discussed.

**DYNAMIC OPERATIONS AND CHECK PHASE: HDCP**

When they retrieve the file, all dynamic operations are carried out, updated constantly, and data files are accessible with their corrections and updates. The cloud storage effectively manages the dynamic data storage by often

updating things like images, emails, log files, etc. To improve data integrity and ensure availability with dynamic operations like edit, append, and insertion, it is crucial to update accurate data. To facilitate dynamic functioning, the clients must download the entire file throughout the verification process. The upgrading of the data and distribution of it once more to the server is completed, although it is extremely chaotic and unsafe for the larger files. With the methods outlined below, the HDCP protocol supports dynamic operation effectively: A) Planning B) Execution C) Verification.

**Preparation :HDCP**

The client adds, removes, and appends data that is subject to dynamic operations.In light of this, the client creates an update request for dynamic operations, which is then sent to cloud service providers. The client makes the following update requests:

- To request a modification of the data in the data block, the client simply multiplies the new value by the current value. With no contact to the unaltered data, the client created the update parity blocks using Cauchy Reed-Solomon algorithm. It automatically replaces the old data occurrence with new owns and has an impact on the remaining metadata that is handled by the client.
- When deleting data, the deleted portion is typically replaced with a zero or with reserved data symbols. When it affects the tokens, it uses the same process as the modification operation.
- Adding a new file at the end of an existing one to increase the size of a cloud-based file is known as an append operation. The client receives blocks in their largest possible size.
- The data file saved in the cloud is same for insertion update operations as it is for append operations, but the data can be put at the desired location while maintaining the block level structure for the entire file. When a new one is introduced, the location is changed by relocating the blocks and recalculating the token for those spots. As a result, completing an insertion operation is difficult. When the update is finished being prepared, it is sent to the CSP.

**EXECUTION UPDATE: HDCP**

The CSP changes the operation in accordance with the client's request and saves the most recent version in the file as explained in the following algorithm.

Algorithm Execution update

*Step1: Procedure Execution update*
*Step 2: if the (update == append or modification)*
*Step 3: the new block is update along appended ormodified block with position*
*Step 4: Else if (updates == delete)*
*Step 5: the block exclude the blocks mention and update*
*Step 6: move all block one step behind after $i^{th}$ block*
*Step 7: end if*
*Step 8: end procedure.*

Check update : HDCP

After any modifications or operations are made to the data, the client checks to see if the files have been updated. It determines whether updates have been successfully finished. The dynamic operation of verification continues to perform a verification procedure as an algorithm to check for originality. It computes vi rather than hashing the blocks.The adversary uses XOR to obtain the output of the single block's hashing. The simulator's capacity to extract the blocks questioned during the challenge procedure remains unchanged.

**SECURITY ANALYSIS OF HDCP**

The analysis show how security of the proposed is efficient than the existing system as in Integrity. The analysis of the corruption detection likelihood reveals that the HDCP is a better method for dealing with random blocks, not necessarily all of them. Thus, despite the high chance of data corruption detection, it keeps computational costs low for both client and server. The suggested approach displays stronger and more effective integrity than current

probabilistic schemes. HDCP assists in identifying data corruption using random sampling with high probability for integrity check evidence when threats are limited to a portion of the file. In cases when the current results are unsatisfactory, the Sobol sequence is used for the random sample generations with the complete file being equally and uniformly dispersed in the file block. It is unreliable because the pseudorandom sequence may not use the entire file to produce an integrity verification.

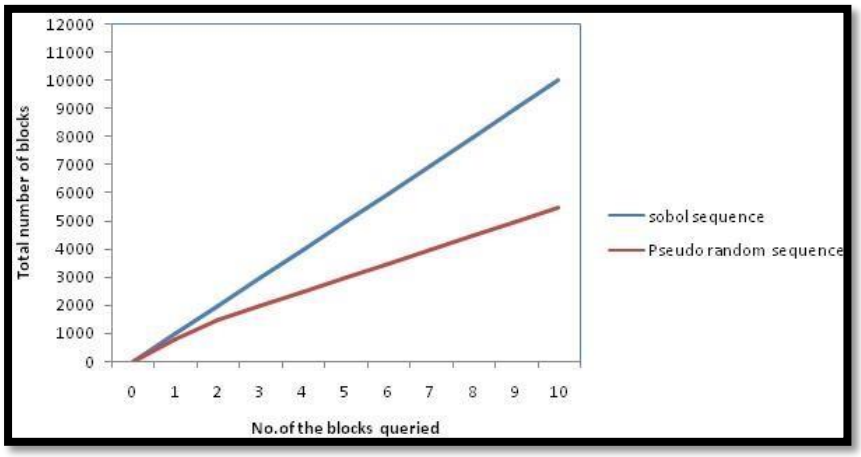


**Figure-4 The Probability Detection Against the Adversary Data Corruption**

**Efficiency**

When analyzing the current system using HDCP, the client or verifier notices that more random samples are being taken in order to achieve high probability, rendering the current system no longer be a simple procedure. However, HDCP finds the fault with high probability in a small number of samples taken at random. Because the current approach is based on a pseudo-random number generator (PRNG), which tends to exhibit cluster effects, the results have poor integrity. To get over this issue, Sobol employs a generation random number that exhibits uniformity in distribution random sequence, producing a better and more reliable outcome in maintaining the data's integrity.

**Availability**

The cloud-stored data file should be accessible to clients and end users around-the-clock and free of corruption. If the suggested system detects a case of data loss or theft, corrective action is then done to make the data available by recovering and restoring it. The HDCP uses an erasure code to guarantee data retrievability by creating parity n blocks from m numbers of blocks. The original data can be recreated from (m, n) blocks in the event of any data loss or server failure.

**PERFORMANCE ANALYSIS OF HDCP**

The performance focuses primarily on the execution of encoding, production of meta data, and CSP computation that yields experimental results with Encode

Data integrity is ensured during HDCP encoding. The comparison of vandermonde and Cauchy Reed Solomon codes yielded findings that demonstrate the average encoding expense utilizing 1GB.In both tables, the parity n is fixed blocks are 2 and increased m data blocks in set I, while m is fixed blocks are increased by 10 times in set II. As the length of the m rises, the data block gets smaller, necessitating fewer visits to the Cauchy reed Solomon encoder.

**Table-1**

**Vander Monde Reed Solomon Encoding Cost**

| Set 1 | m=4 | m=6 | m=8 |
|---|---|---|---|
| n=2 | 111.01 s | 82.06 s | 64.45s |
| Set 11 | n=2 | n=4 | n=6 |
| m=10 | 50.1 s | 82.3s | 138.33s |

**Table-2**
**Cauchy Reed Solomon Encoding Cost**

| Set 1 | m=4 | m=6 | m=8 |
|---|---|---|---|
| n=2 | 81.3s | 63.07s | 46.56s |
| Set 11 | n=2 | n=4 | n=6 |
| m=10 | 31.21 s | 58.58s | 100.34s |

**META DATA GENERATION**

It is measured how quickly metadata is computed. The metadata of the proposed homomorphic distributed check protocol are highly dynamic. According to Carter et al. (1979), the client first creates a permutation and then calculates a random key using a sobol sequence and UHF information.Because the input is so small, compared to hashing functions, the average token pre-computation cost by implementing HDCP is 0.2.

**CLOUD SERVICE PROVIDER COMPUTATION**

By assessing the integrity proof of the challenge response that CSP accepted, the CSP computation is measured. Because only the integrity check is produced for the block selected challenge, the HDCP produces responses faster than the current protocol. Take into account 20000 blocks, where 2% of them are contaminated. The current system calculates c=920 blocks to achieve 99% probability detections whereas the CSP produces responses with c=720 blocks at random.

**Table-3**
**Probability detection & data corruption out of 20000 blocks**

| Probability detection | No. of sample / total samples needed | |
|---|---|---|
| | HDCP | Existing system |
| 0.1 | 80 | 220 |
| 0.8 | 180 | 240 |
| 0.9 | 240 | 460 |
| 1.0 | 320 | 560 |
| 1.20 | 580 | 640 |

The homomorphic distribution check protocol (HDCP) is a suggestion that would improve the availability and integrity of cloud servers. To ensure availability, erasure codes are used in the duplicated area. Tokens are pre-calculated using the Sobol sequence in order to examine and confirm the data's uniqueness. The HDCP is more likely to be effective and guarantee authenticity thanks to security and performance analysis.

However, the dynamic data operations are less robust due to the building of the meta data and the file index, which necessitates the computation of the meta data with each new index and increases computational cost. Additionally, it lacks imagination and power because it does not offer public verifiability for huge data files. The dynamic public audit protocol is suggested to fix HDCP's vulnerability.

**CONCLUSION**

The prevalence of cloud storage throughout the world has led people to rely more on a variety of online storage systems to back up their data or use it immediately, providing access from anywhere at any time. In a day where data growth is exponential, data storage is a crucial component. Huge amounts of sensitive data are kept in the cloud. Information that is modified asynchronously as new updates become available is referred to as dynamic data. There is no guarantee that the data that has been outsourced and stored on remote storage servers is secure or hasn't been lost or altered. These services raise worries about security risks for all the services they offer because the user's data are maintained and stored outside of the user's premises. To address the problems with service providers and cloud security, extensive study is done. Since there is less control over the consumers' data and it is outsourced to external servers, security becomes the main concern. Due to this vulnerability, data can be lost or changed.

**REFERENCES**


- Ahlem Bouchahda, Nhan Le Thanh, Adel Bouhoula & FatenLabbene 2010, _RBAC+: Dynamic Access Control for RBAC-Administered Web-Based Databases', In Proceedings of the Fourth International Conference on Emerging Security Information, Systems and Technologies, pp. 135-140.
- Arputharaj Kannan 2020, _Fuzzy logic based unequal clustering for wireless sensor networks', Wireless Networks vol. 22, no. 3, pp. 945-957.
- Boubaker Souha, Mohamed Graiet & Nejib Ben Hadj- Alouane 2015, _Event-B Based Approach for Verifying Cloud Resource Allocation in Business Process', IEEE International Conference on Services Computing, pp. 538-545.
- Dario Pompili & Jingang Yi 2020, _Dynamic Collaboration Between Networked Robots and Clouds in Resource- Constrained Environments', IEEE Transactions on Automation Science and Engineering, vol. 12, no. 2, pp. 472-479.
- Huaqun Wang, Debiao He & Shaohua Tang 2016, _Identity-Based Proxy-Oriented Data Uploading and Remote Data Integrity Checking in Public Cloud', IEEE Transactions on Information Forensics and Security, vol. 11, no. 6, pp. 1165-1176.
- Kaitai Liang, Willy Susilo, Jianghua Liu & Yang Xiang 2019, _Two-Factor Data Security Protection Mechanism For Cloud Storage System', IEEE Transactions on Computers, vol. 65, no. 6, pp. 1992-2004.
- Nagpure Mahesh, Prashant Dahiwale & Punam Marbate 2015,_An Dynamic Resource allocation Strategy for VM Environment in Cloud', International Conference on Pervasive Computing (ICPC), pp. 1-5.
- Yung-Hsiang Lu & Thomas J Hacker 2018, _Adaptive Cloud Resource Allocation for Analysing Many Video Streams', IEEE 7th International Conference on Cloud Computing Technology and Science, pp. 17-24.