iJRASET
International Journal For Research in Applied Science and Engineering Technology

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

**Volume:** 7 **Issue:** V **Month of publication:** May 2019

**DOI:** https://doi.org/10.22214/ijraset.2019.5227

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
*ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.177*
*Volume 7 Issue V, May 2019- Available at www.ijraset.com*

# Public Integrity Auditing for Shared Dynamic Cloud Data with Group User Revocation

Miss. Nayana A. Deshmukh[1], Prof. Umakant Mandawkar[2]
[1, 2]*Department of Computer Science, Sandip University, Nasik, India*

***Abstract:*** ***Now daily's distributed computing is one of the top innovation idea. Distributed storage servers assume a significant job in the buzz distributed computing innovation where customers can store their information at cloud servers and can get to this information from anyplace and whenever. This opens up the danger on the information which is the customer is putting away on the cloud as it needs to store and access from any gadget through web. This Paper manages the information Security which opens up part of zones like information respectability check, confirmation of customers and information security saving. In the proposed framework for the most part concentrate is on information respectability zone which is worried about information security and the Third Party Auditors (TPA) is used to check the legitimacy of the information without really getting to the information on cloud.***

***Keywords Cloud computing, big data, authorized public auditing, fine-grained updates, TPA.***

## I. INTRODUCTION

In the current framework there are heaps of issues identified with the effectiveness, for example, security hazards in unapproved inspecting demands and in productivity in handling little updates still exist. Alongside these referenced downsides of the current framework there are parcels of correspondence overheads are available in managing the customer and the server supplier in the current framework. The existing plan is additionally not adaptable to the substantial no of clients. Cloud clients may likewise need to part enormous datasets into littler datasets and store them in various physical servers for unwavering quality, protection safeguarding or productive handling purposes. Be that as it may, as referenced above in the framework it will expand the correspondence overhead at both the side customer side just as at the server side as well.

In the proposed framework engineering we will concentrate on the better help for little unique updates which benefits the adaptability and effectiveness of the distributed storage server. As in the current framework there isn't so work done yet has a place with these sort of QoS parameters here the work on the current module is done to improve the productivity too at the equivalent time greater security to the clients just as to the inspector for the trusted TPA. For this reason we are going to utilize the broadened open reviewing plan dependent on BLS signature what's more, Merkle hash tree (MHT) that can bolster fine-grained update demands. Contrasted with existing plans, our plot underpins refreshes with a size that isn't confined by the measure of record squares, in this way offers additional adaptability and adaptability contrasted with existing plans.

There are various security dangers related with cloud information administrations, not just covering customary security dangers, e.g., arrange listening stealthily, unlawful intrusion, and disavowal of administration assaults, yet in addition including specific distributed computing threats, e.g., side channel assaults, virtualization vulnerabilities, and maltreatment of cloud administrations. To throttle the dangers the accompanying security prerequisites are to be met in a cloud information administration.

Information confidentiality is the property that information substance is not made accessible or revealed to illicit clients. Redistributed information is put away in a cloud and out of the proprietors' immediate control. Just approved clients can get to the delicate information while others, including CSPs, ought not increase any data of the information. In the interim, information proprietors hope to completely use cloud information administrations, e.g., information look, information calculation, and information sharing, without the spillage of the information substance to CSPs or different foes.

Access controllability implies that an information proprietor can play out the specific confinement of access to his information out sourced to cloud. Legitimate clients can be approved by the proprietor to get to the information, while others can't get to it without authorizations. Further, it is alluring to uphold fine-grained get to control to the re-appropriated information, i.e., diverse clients ought to be conceded distinctive access benefits with respect to various information pieces.

Plans in presence experience the fill effects of a few normal down sides. Initial, a vital approval /verification process is absent between the reviewer and cloud specialist organization, i.e., anybody can challenge the cloud specialist co-op for a proof of uprightness of certain file, which possibly puts the nature of the alleged evaluating as-an administration in danger; Second, albeit a

portion of the ongoing work dependent on BLS mark would already be able to help completely unique information refreshes over fixed-measure information blocks, they just help refreshes with fixed -sized squares as essential unit, which this task call coarse-grained refreshes. Accordingly, every little update will cause re-calculation and refreshing of the authenticator for anent re-file block, which thusly causes higher capacity and correspondence overheads. This task will work for diminishing these downsides from the framework.

## II. LITERATURE SURVEY

This section introduces existing related work and describes the dissimilarities and differences from this project work. Although current development and proliferation of cloud computing is rapid, debates and hesitations on the usage of cloud still exist. Data security/privacy is one of the major concerns in the adoption of cloud computing. Compared to conventional systems, users will lose their direct control over their data. In previous approach, there is the problem of integrity verification for big data storage in cloud. This problem can also be called data auditing when the verification is conducted by a trusted third party. From cloud users perspective, it may also be called auditing-as-a-service. Compared to traditional systems, scalability and elasticity are key advantages of cloud As such, efficiency in supporting dynamic data is of great importance. Security and privacy protection on dynamic data has been studied extensively in the past. In this approach, this project will focus on small and frequent data updates, which is important because these updates exist in many cloud applications such as business transactions and online social networks (e.g. twitter). There is a lot of work trying to enhance cloud data security/privacy with technological approaches on CSP side.

2.1.1 Authorized Public Auditing of Dynamic Storage on Cloud with Efficient Variable Fine-Grained Updates with MHT, 2016 The advent of the cloud computing makes storage out sourcing become a rising trend, which promotes the secure remote data auditing a hot topic that appeared in the research literature. Recently some research considers the problem of secure and efficient public data integrity auditing for shared dynamic data.

S. Nepal, S. Chen, J. Yao, and D [1]. Thilakanathan, ‘‘DIaaS: Data Integrity as a Service in the Cloud,’’ in Proc. 4th Int’l Conf. on Cloud Computing (IEEE CLOUD), 2011, pp. 308-315.Migrating the virtual machines from on-premise to Cloud raises new security challenges for a company. A potential threat to the tenants is not only the Internet hackers, but also the cloud service provider and the other co-tenants, due to the multi-tenancy feature. The cloud service provider’s security is challenged by the tenants, as well. Deploying the open source cloud raises additional challenges since the intruders have access to the cloud source code and can assess its vulnerabilities. In this paper, we thoroughly assess the security vulnerabilities of Open Stack cloud framework, one of the most used open source cloud frameworks today. We assess the vulnerabilities of Open Stack server node, virtual machine instances and Open Stack’s Dashboard, the web management interface. The security assessment shows that Open Stack cloud has security vulnerabilities that need to be secured by developing the patch [1].

E. None, [2] ‘‘What Twitter Learns from All Those Tweets,’’ in Technology Review, Sept. 2010, accessed on: March 25, 2013. [Online]. Available: http://www.technologyreview.com/view/420968/whattwitter-learns-from-all-those tweets/ Cloud Computing is the long dreamed vision of computing as a utility, where users can remotely store their data into the cloud so as to enjoy the on demand high quality applications and services from a shared pool of configurable computing resources. By data outsourcing, users can be relieved from the burden of local data storage and maintenance.

### *A. Algorithm Used*

*1) Message Digestion (MD5):* The MD5 algorithm is a widely used hash fanon producing a 128-bit hash value. Although MD5 was initially designed to be used as a cryptographic hash function, it has been found to suffer from extensive vulnerabilities.

Steps

*a)* It Is Designed To Run Effectively On 32 Bit Processor.
*b)* Generate Unique Hash Value For Each Input.
*c)* It Produce Fixed Length 128 Bit Hash Value With No Limit Of Input Message.
*d)* Advantage Is Fast Computing And Uniqueness.
*e)* Also Known As Hashing Function [2].

X. Zhang, L.T. Yang, C. Liu, and J. Chen,[3] ‘‘A Scalable Two-Phase Top-Down Specialization Approach for Data Anonymization Using Map Reduce on Cloud, ’IEEE Trans. Parallel Distributed. Syst., vol. 25, no. 2, pp. 363-373, Feb. 2014. In this paper, we propose a scalable two-phase top-down specialization (TDS) approach to anonymalize large-scale data sets using the Map Reduce framework on cloud. In both phases of our approach, we deliberately design a group of innovative Map Reduce jobs to concretely accomplish the specialization computation in a highly scalable way. This approach get input data’s and split into the small data sets.

*ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.177*
*Volume 7 Issue V, May 2019- Available at www.ijraset.com*

Then we apply the ANONYMIZATION on small data sets to get intermediate result. Then small data sets are merge and again apply the ANONYMIZATION. We analyze the each and every data set sensitive field and give priority for this sensitive field. Then we apply ANONYMIZATION on this sensitive field only depending upon the scheduling.[3]

S.E. Schmidt,[4] ‘‘Security and Privacy in the AWS Cloud,’’ presented at the Presentation Amazon Summit Australia, Sydney, Australia, May 2012, accessed on: March 25, 2013. [Online]. Available: http://aws.amazon.com/apac/awssummit-au/. Possession at untreated stores," in Proc. of ACM CCS, Virginia, USA, Oct. 2007, pp. 598–609. With resource virtualization, cloud can deliver computing resources and services in a pay-as-you-go mode, which is envisioned to become as convenient to use similar to daily-life utilities such as electricity, gas, water and telephone in the near future [1]. These computing services can be categorized into Infrastructure-as-a-Service (IaaS), Platform-as-a Service (PaaS) and Software-as-a Service (SaaS) [3]. Many international IT corporations now offer powerful public cloud services to users on a scale from individual to enterprise all over the world; examples are Amazon AWS, Microsoft Azure, and IBM Smart Cloud [4].

*i) Algorithm*

TREEHASH (start, maxheight)

*a.* Set leaf = start and create empty stack.
*b.* Consolidate: If top 2 nodes on the stack are equal height:
*c.* Pop node value P (n right ) from stack.
*d.* Pop node value P(n left) from stack.
*e.* Compute P(n parent) = f(P (n lef t || P ( n right )).
*f.* If height of P(n parent) = maxheight output P(n parent).
*g.* Push P(n parent) onto the stack.
*h.* New Leaf: Otherwise:
*i.* Compute P (n l) = LEAF CALC (leaf).
*j.* Push P (n l) onto the stack.
*k.* Increment leaf.
*l.* Loop to step 2.

*ii) Algorithms*

SHA1- with the reference from – rfc3174 for SHA1 algorithm

```
#ifndef _SHA1_H_
#define _SHA1_H_
#include
<stdint.h/*
* If you do not have the ISO standard
stdint.h   header file, then you
* must typdef the following:
* name meaning
* uint32_tunsigned 32 bit integer
* uint8_tunsigned 8 bit integer
i.e., unsigned char)
* int_least16_t integer of >= 16 bits
*
*/
#ifndef _SHA_enum_
#define _SHA_enum_
 enum
{
 ShaSuccess = 0,
 ShaNull,        /* Null pointer    parameter */
 Sha Input Too Long,   /* input data too long   */
 shaStateError       /* called Input after Result */
};
```

```
#endif
#define
SHA1HashSize 20 /* 3
* This structure will hold context information for the
SHA-1
 * hashing operation
 */
 typedef struct SHA1Context
 {
uint32_Intermediate_Hash[SHA1HashSize/4];
/* Message Digest */          uint32_t Length Low;    /* Message length in bits
  */
uint32_t Length High;   /* Message length in bits */
/* Index into message block array */
int_least16_t Message_Block_Index;
uint8_t Message Block[64];
/*512-bit  message blocks  */
int Computed;      /* Is the digest computed?
  */ int Corrupted; /* Is the message digest corrupted? */ }
SHA1Context;
/*
* Function Prototypes
*/
int SHA1Reset( SHA1Context *);
int SHA1Input( SHA1Context *,
 const uint8_t *,
 unsigned int);
int SHA1Result( SHA1Context *,
uint8_t Message_Digest[SHA1HashSize]);
#endif
```

*iii)* *Description:* SHA1 calculation is very much depicted in RFC 3174 - US Secure Hash Algorithm 1 (SHA1), see http://www.ietf.org/ rfc/rfc3174.txt. The following is a snappy review of the calculation.

SHA1 calculation comprises of 6 undertakings:

Undertaking 1. Annexing Padding Bits. The first message is "cushioned" (broadened) with the goal that its length (in bits) is consistent to 448, modulo 512. The cushioning rules are: The first message is constantly cushioned with one piece "1" first. At that point at least zero bits "0" are cushioned to bring the length of the message up to 64 bits less than a numerous of 512. Errand 2. Attaching Length. 64 bits are attached as far as possible of the cushioned message to demonstrate the length of the first message in bytes. The guidelines of annexing length are: The length of the first message in bytes is changed over to its parallel arrangement of 64 bits.

In the event that flood occurs, just the low-request 64 bits are utilized. Break the 64-bit length into 2 words (32 bits each). The low-request word is annexed first and pursued by the high-request word.

Assignment 3.Getting ready Processing Functions. SHA1 requires.

80 handling capacities characterized as:

f(t;B,C,D) = (B AND C) OR ((NOT B) AND D)

( 0 <= t <= 19)

f(t;B,C,D) = B XOR C XOR D

(20 <= t <= 39)

f(t;B,C,D) = (B AND C) OR (B AND D)

OR (C AND D)

(40 <= t <= 59)
f(t;B,C,D) = B XOR C XOR D (60 <= t <= 79)
Undertaking 4. Getting ready Processing Constants. SHA1 requires.
80 preparing steady words characterized as:
K(t) = 0x5A827999 ( 0 <= t <= 19)
K(t) = 0x6ED9EBA1 (20 <= t <= 39)
K(t) = 0x8F1BBCDC (40 <= t <= 59)
K(t) = 0xCA62C1D6 (60 <= t <= 79)
Undertaking 5. Instating Buffers. SHA1 calculation requires 5 word supports with the accompanying beginning qualities:
H0 = 0x67452301
H1 = 0xEFCDAB89
H2 = 0x98BADCFE
H3 = 0x10325476
H4 = 0xC3D2E1F0
Errand 6. Preparing Message in 512-piece Blocks. This is the principle undertaking of SHA1 calculation, which circles through the cushioned and added message in squares of 512 bits each. For each info hinder, various activities are performed. This errand can be depicted in the accompanying pseudo code somewhat adjusted from the RFC 3174's technique
Information and predefined capacities
M[1, 2, ..., N]: Blocks of the cushioned and added message
f(0;B,C,D), f(1,B,C,D), ..., f(79,B,C,D): Defined as
above K(0), K(1), ..., K(79): Defined as above
above K(0), K(1), ..., K(79): Defined as above
H0, H1, H2, H3, H4, H5: Word cradles with starting qualities.

---

*iv) Algorithm*

For loop on k = 1 to N
(W(0),W(1),...,W(15)) = M[k] /* Divide M[k] into 16 words */
For t = 16 to 79 do:
W(t) = (W(t-3) XOR W(t-8) XOR W(t-14) XOR W(t-16)) <<< 1
A = H0, B = H1, C = H2, D = H3, E = H4
For t = 0 to 79 do:
TEMP = A<<<<5 + f(t;B,C,D) + E + W(t) + K(t) E
= D, D = C, C = B<<<30, B = A, A = TEMP
End of for loop
H0 = H0 + A, H1 = H1 + B, H2 = H2 + C, H3 = H3 + D, H4 = H4 + E
End of for loop
Output: H0, H1, H2, H3, H4, H5: Word buffers with final message digest
Step 5. Output. The contents in H0, H1, H2, H3, H4, H5 are returned in sequence the message digest

---

*B. ECC Algorithm*

*1)* ECC is a block cipher with 128 bits block length.

*2)* ECC allows for 3 different key lengths: 128, 192, or 256 bits. Our discussion Primarily will assume that the key length is 128 bits.

*3)* Encryption is going to be 10 rounds of processing for 128 bit keys, 12 rounds for 192 bit keys and 14 rounds for 256 bit keys.4) All the remaining rounds are identical with exception for the last round in each case in Computer and Network Security.5) One single-byte based substitution step is include in each round of processing, a row-wise permutation step, a column-wise mixing step, and the addition of the round key. These four steps executing order is different for encryption and decryption. 6) To enhance the processing steps used in a single round, it is best to think of a 128-bit block as consisting of a 4 × 4 matrix of bytes, arranged. 7) Therefore, the first four bytes of a 128-bit input block will occupy the first column in the 4 × 4 matrix of bytes. The next four bytes occupy the second column, thus it will continue. 8) The 4 × 4 matrix of bytes is referred to

as the state array. 9) ECC also has the perception of a word. A word consists of 4 bytes means 32 bits. Hence each column of the state array is a word, as is each row. 10) Each processing round works on the input state array and produces an output state array.11) The output state array produced by the last round is rearranged into a 128-bit output block. 12) Unlike DES, the decryption algorithm differs largely from the encryption algorithm. Somehow the same steps are used in encryption and decryption, the order in which the steps are carried out is different, as mentioned previously. 13) ECC, notified by NIST as a standard in 2001, is a slight variation of the Rijndael cipher invented by Belgian cryptographers Vincent Rijmen, Joan Daemen. 14) Whereas the block size that ECC requires at 128 bits, the original Rijndael cipher works with any block size (and as well any key size) that is a multiple of 32 as long as it will exceeds 128. The state array for the different block sizes still has only 4 rows in the Rijndael cipher. But the number of columns depends on size of the block. For instance, when the block size is 192, the Rijndael cipher requires a state array to consist of 4 rows and 6 columns.

## III. MATHEMATICAL MODEL

In this project assumption for the best this project proposed system architecture we will assume that in the auditor part for the TPA will responds to each query in the proper way and honestly. Consider a set of CSS (Cloud storage services) denoted as follows: Cs ={Cs1,Cs2,Cs3,...Csn} Where Care presents all the space sofa cloud to provide the services to the end users where at the end a TPA auditor can fire query to CSS, to verify the authentication. Let us say that this set of query is represented as follows: Qr ={Qr1,Qr2,Qr3,...Qrn} These are the n no of queries to the CSS from the TPA for the verification purposewherecurrentlytherearemanydrawbacksintheexistingschemeasmentioned above. So to overcome those drawbacks let us assume the association between the above defined two sets as shown in the following figure where it overcomes the associated drawbacks of the system

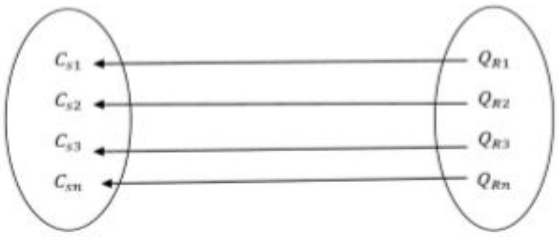



Figure 5.1: One to One mapping between sets

As this project is handling every query and giving reply to it honestly from the CSS where this work flow is handle in this project proposed system separately for each and every query towards the CSS. Means it will not create more communication overheads at the server side and also increases the security too. Now consider the small grained updates denoted by the set U where the elements of the declared set are as follows: Up ={U1,U2,U3,...Un} So these are all the small up dates of the system where it belongs to the set Upwhere handling of these small updates in the less no of overheads is the important part in this project proposed system. So these part is handled in this project proposed system very efficiently. These updates are said to be the subset of set S where S denotes the services overall in the system.

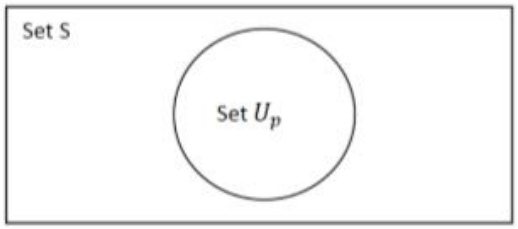



Figure 5.2: Association between two sets Up and S

As a service may have n no of small updates its a subset of the set S where each service consists of the small updates the relations between the above two sets can be shown in the following figure the mapping between these two sets are as follow:

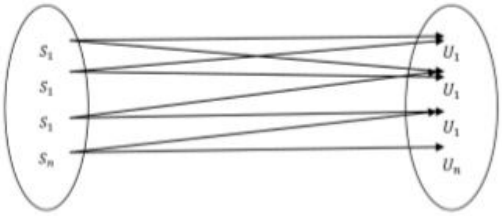



Figure 5.3: Association between two sets Up and S

So, this projects second main of the proposed system architecture where these are the important parts in the mathematical module development also as well as the in the core implementation of the proposed system architecture. So this is all about the overall modules of this project.

*A. Proposed System Modules*

*1)* User Registration (Sign In / Sign Up)

*2)* Uploading Files

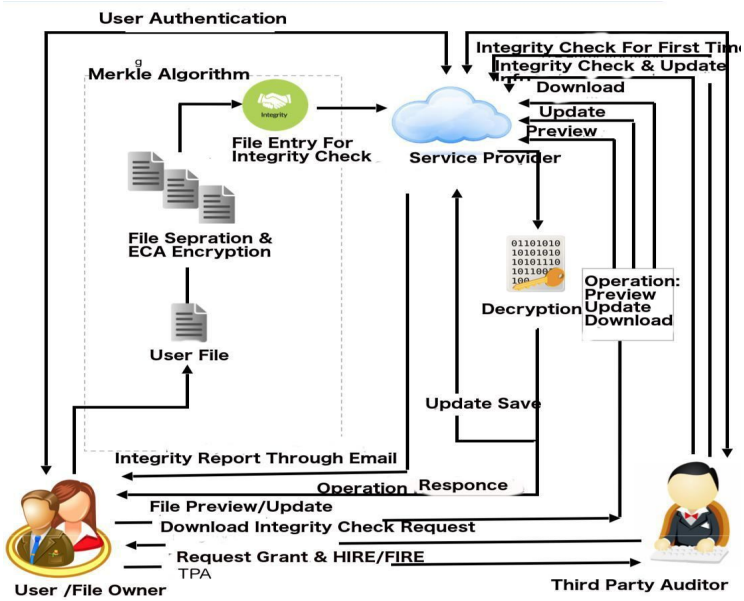
*3)* File Preview and Downloading

*a)* *User Registration:* In this module first client register with our application by including his own data like his name, secret word, address and his diversions etc. After enlisting with our application he can login with us utilizing user id and secret key. Framework verify this data and enables client to login into the framework.

*b)* *Uploading Files:* After login, the client can transfer the records to the cloud server space. The Uploaded records are getting isolated in to numerous parts and those parts are getting scrambled utilizing the ECC calculation. Here the Working of Merkle Hash tree begins. The Leaf hubs of the Merkle Hash Tree will be the Document Parts.

*c)* *File Preview and Downloading:* After login, the client can transfer the records to the cloud server After record Uploading has been done client can Update and Review the File. The client can Directly get to the predetermined part for the refreshing. The Updating will be done in that explicit part rather than the Whole document. The will be the fine grained refreshes. For every activity like Preview and Updates the documents are getting Decrypted , Encrypted and Converge for every activity.

Architecture



*ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.177*
*Volume 7 Issue V, May 2019- Available at www.ijraset.com*

## IV. SYSTEM DESIGN

In system design there are following modules such as Authentication, File Uploads, Integrity check , Request for integrity check And File Operation They are explained as follows:

### *A. Authentication*

The Cloud Service clients, similar to File Owner and Third Gathering Auditor are verified by the Cloud Service supplier. Indeed, even in the wake of enrolling with the application the client isn't permitted to get to any piece of use until the client is qualified by the CSP. Clients should be initiated by the CSP, and correspondingly can be deactivated as well.

The TPA is in charge of checking the trustworthiness of the document and report the record proprietor about status. So the TPA is additionally confirmed by the document proprietor to get to the record. Also even the TPA is getting to the document for checking the honesty yet really TPA will get the figment of getting to the record however there is in no way like document that TPA is getting to. The Values on which TPA finds the consequence of trustworthiness is only the Hash estimations of portions of encoded records processed all things considered. So the genuine unique record is absent on the server for what it's worth.

### *B. File Uploads*

The client transfers the documents on CSP space. The records are parceled in the parts which are then scrambled with the 256bit ECA calculation. These records are utilized by the SHA1 calculation to register its hash key. In the event that any progressions done by the client in any part will reflect in hash an incentive for the entire document. In any case, as a general rule, the entire document in real arrangement is absent on the administration space. This comprises the Merkle Hash Tree Algorithm which is the primary piece of our venture.

### *C. Integrity check*

When the user uploads the files that need to be checked by the Hired TPA. This will be done by the Integrity check functionality. First-time integrity check IV. Integrity Preserved. Integrity Lost.

This is the possible outcomes of the integrity check by the TPA.

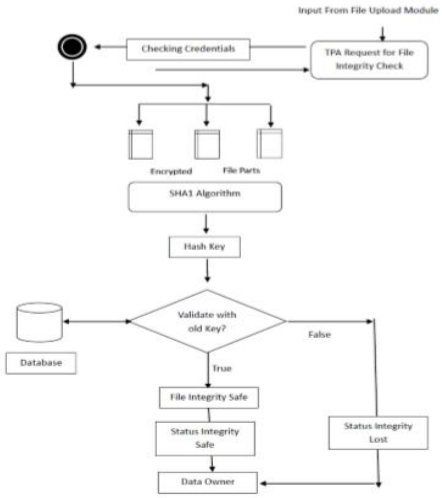



Figure 8.1: Integrity check module

### *D. File Operation*

1) *File Update:* Document Updating is an ordinary capacity of the client. In any case, the inventive thought here is that the client is refreshing just a required part and the part is in a scrambled organization. In the wake of refreshing the part will again be scrambled in its area and jam its honesty. The Encryption and unscrambling expected to take every necessary step.
2) *File Preview:* Document Preview additionally needs a record to encode before seeing.
3) *File Download:* The record should be in unscrambled before downloading. Here the AES Decryption keeps running for a specific record of Particular User. The File will be downloaded

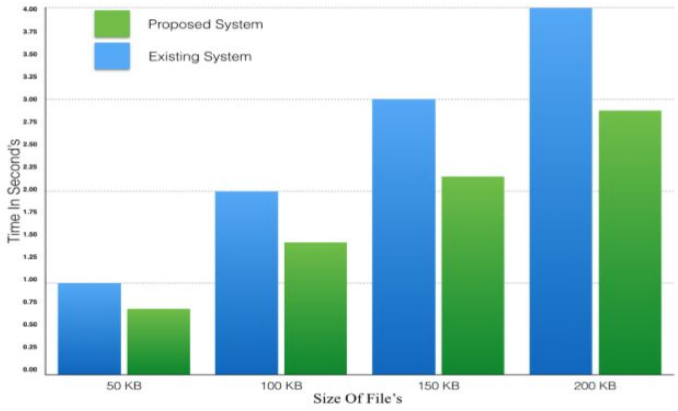
*E. Request for Honesty Check*

The outsider Auditor is only the organization which will gives the administration of checking and announcing the honesty status of the client. The Third Party examiner will see the rundown of cloud administration client. The TPA will send the solicitation to the record proprietor to enlist me as a TPA for your archives. At that point the client will choose whether to Grant or overlook the demand. This will forestall the document proprietor to sent solicitation to fake TPA to check the documents. By along these lines the record proprietor will be in key spot to permit TPA for honesty check or then again not. Moreover if after some time document proprietor feels that he/ she needn't bother with the TPA to check records trustworthiness any longer. He/She can HIRE or FIRE the TPA.

*F. Priority Based Scheduling*

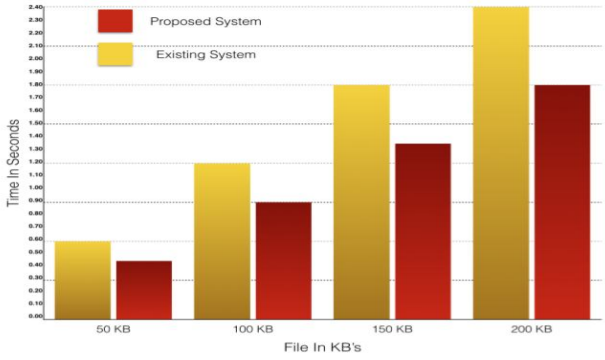
As the asset might be shared by the various substances at possibly a similar time or diverse time, so there ought to be the component for apportioning asset. As JSP Servest is as of now a string safe which will dispense the different case of the article to the requester? Moreover, we added the Priority based Algorithm to keep up its up rightness. The higher need provides for the File proprietor errands, similar to record Updating. The lower need is given to TPA undertaking, for example, checking the respectability of the document. The TPA isn't permitted to gain admittance to the record when the client is refreshing the document. 3)It takes out the need to visit different sites to get to data about train plans and furthermore kills the need to remain in a line to book a ticket.4)Improved client experience It's intuitive and progressively!

Result 1




The graph will show us the time V/S size files, that is how much time the file needs to get updated (As in actual we are updating the part of the file) This will lower the memory. As a result, every small update will not cause re-computation and updating of the authenticator for an entire file

Result 2



## V. FUTURE SCOPE

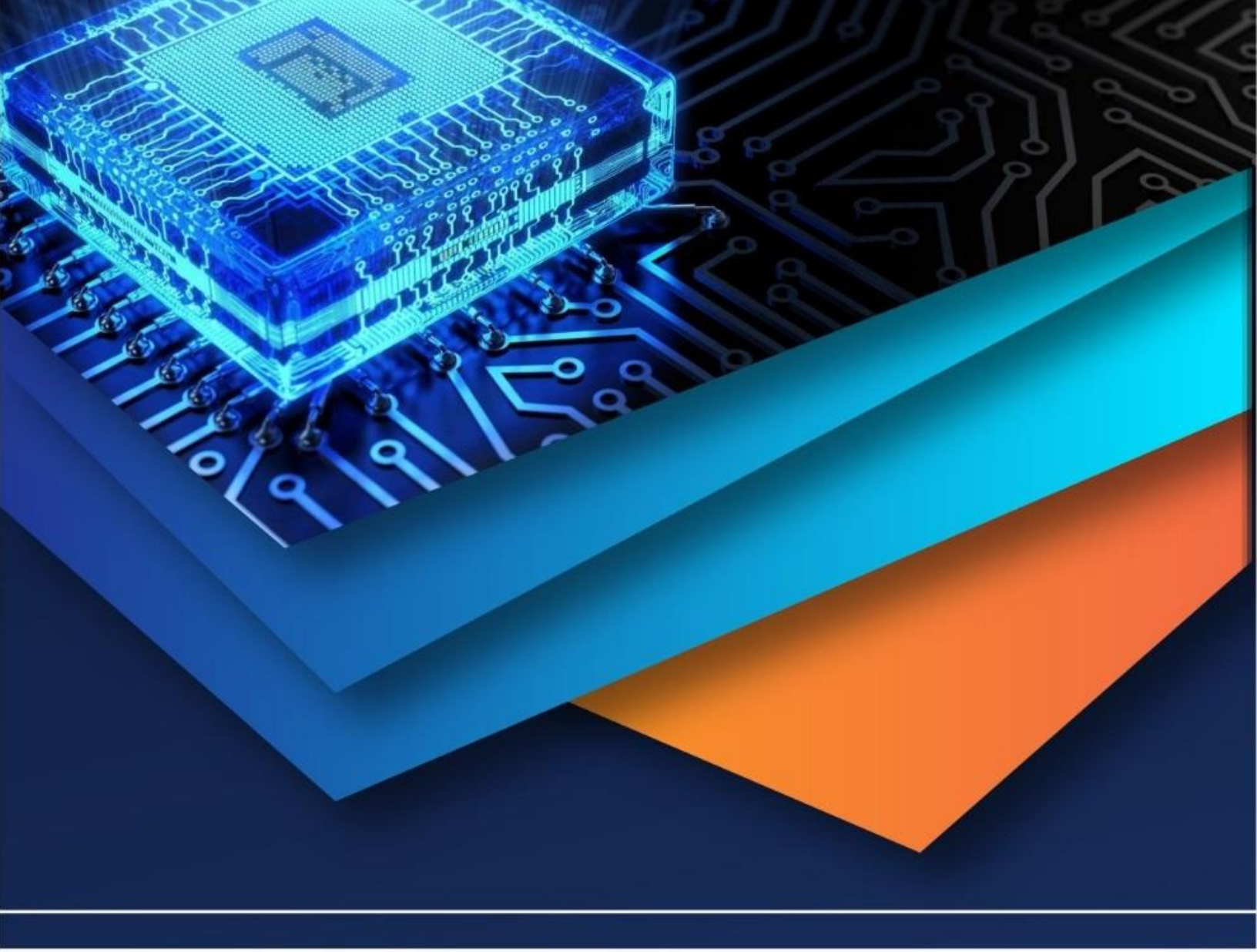
The proprietary file format cannot be updated with the application. Such as MSWord docs as it needs vendor specific software to open the file format. As for the security reason the original file will not be stored anywhere on the cloud, if some changes done by the user that changes cannot be undone once committed.

## VI. CONCLUSION

In the proposed method, data integrity check is done by TPA and the system also provides a scheme that can fully support authorized auditing and fine-grained update requests. The system also reduces communication overheads for verification of small updates. The security and experimental analysis show that, the proposed method requires less time to upload file on the cloud. The efficiency of this system almost increased by 5% than the existing system.

## REFERENCES

[1] Authorized Public Auditing of Dynamic Big Data Storage on Cloud with Efficient Veritable Fine-Grained Updates;, IEEE-, 2016
[2] Trust Public Auditing of Dynamic Big Data Storage with Efficient High Memory Utilization and ECC Algorithm, IEEE-, 2015
[3] Privacy-Preserving Public Auditing using TPA for Secure Cloud Storage;,IEEE-, 2014
[4] Trust Privacy-Preserving Public Auditing using TPA for Secure Cloud Storage, IEEE-, 2014
[5] Trust Based Security Service Mechanism For Client End Security Using Attribute Based Encryption At Cloud Platform IEEE-, 2013
[6] Trust A Systematic Approach For Ensuring Security And Efficiency In Cloud Computing,IEEE-, 2013
[7] Cooperative Provable Data Possession for Integrity Verification in Multi-Cloud Storage, IEEE-, 2012
[8] Remote Data Checking Using Provable Data Possession, ACM Trans .Inf .Syst. Security, vol. 14, no. 1, 2011
[9] Dynamic Provable Data Possession Compute. and Commun. Security (CCS), IEEE-, 2009
[10] TMR-PDP: Multiple-Replica Provable Data Possession, in Proc. 28th IEEE Conf. on Distribute. Compute. Syst. (ICDCS), 2008
[11] Scalable and Efficient Provable Data Possession, in Proc.4thIntlConf.Security and Privacy in Commun. Netw. (Secure Comm), 2008
[12] Provable Data Possession at Untrusted Stores, in Proc. 14th ACM Conf. on Compute. And Commun. Security (CCS), 2007



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY