CONTINUOUS PUBLIC AUDITING AND DATA REGENERATION ON CLOUD STORAGE

R.Boob[1], Prof.S.M.Rokade[2]
[1] M.E., Computer Engineering Department, Maharashtra, India
[2] HOD, Computer Engineering Department, Maharashtra, India

ABSTRACT

CSP provides high level security and compliances for cloud services. But if we consider these services as a part of multi-year or ever-changing environment then it may put us into reliability issue of cloud service certifications. Continuous auditing i.e. CA required to be continuously reliable and secured by increasing trustworthiness of certifications. To avoid infancy strategy of CA, continuous cloud service auditing approach is proposed. In this various criteria is audited. It is continuous auditing process for cloud services which is launch to extend the concept of continuous cloud service auditing. Another aspect is there is need of data restoration (Backup) in case of any damage or leakage this system contributes proxy server at the cloud end which is responsible for data restoration i.e. code regeneration. With the experimental results we have to prove performance of system in terms of security and auditing of data.

Keyword: *Certification, cloud computing, continuous auditing, security*

1. INTRODUCTION

In this modern era, there is huge growth in outsourcing data to cloud server. Number of organizations outsourcing their data, applications and business processes to the cloud, allow them to achieve financial and technical benefits due to on-demand provisioning and pay-per use pricing. Organizations have still hesitance to adopt cloud services because of security, privacy and reliability concerns. Several CSC are good to address the concerns by establishing trust. Traditionally, several CSC have derive such as, CSA STAR or EuroCloud star audit. These attempts are to assure high level security, reliability and legal compliances for valid period of three years. CSP has ever-changing environment. Therefore, long term assurance of cloud service certification put in doubt and it will result into major security occasion as well as configuration changes. Therefore, there is need of continuous auditing CA to provide transparent assurance of cloud services which will be continuously reliable and secure and to establish a trustworthy CSC after the initial certification process is accomplished. Existing research has main focused on estimating information related to CA system. To monitor and system's information audit there are different methodologies, embedded auditing modules [7] and monitoring control layers have been introduce in[8].

In past review, CA is mostly inspected for internal purposes. To check data integrity, concurrence of data location and dynamicity of cloud infra third party auditor i.e. TPA has been introduced in the contexts of cloud certification [9]. In this research our main focused is to address the limitations of existing techniques by studying CA architecture, including main components, techniques and processes at the time of seeing all requirements of main stakeholders. Before, deciding CA architecture we defined the performance of CA and also to analyze where is applicable. Criteria of CSC is validated by performing establishment with cloud service auditors first. In proposed work, cloud service criteria of cloud services have been evaluated. CA methodologies build in extend to the previous work. There are three objectives on which we have concentrated, 1. To provide continuous auditing for cloud service certification i.e. CSC 2. To find efficient mechanism which can be applicable in the context of continuous auditing of cloud services 3. To prove loyal certification, security and reliable services, the scope of this research paper is to provide reliable services for end user by encouraging auditors and CSP to implement CA techniques, sequentially creating honorable certifications and services. To provide checklist for auditor to classify that high frequency auditing is required after basic certification level or process is accomplished. As part of contribution, this system contributes, regenerating code based cloud storage to solve the problem of data restoration or regenerating codes. Proxy is introduced to regenerate the authenticators, into the traditional public auditing system model.

2. RELATED WORK

P. Stephanow, M. Gall proposed language classes for cloud service certification. It simplifies the research in design and implementation of CA systems. Language classes are developed for signature based intrusion detection systems then it is applied to the CSC system. They have studied similarities between signatures based IDS and cloud service certification adapted to recyclable concepts for certification system. They defined six languages classes such as, event, response, and reporting, correlation, and detection policy as well as detection mechanism. The concept of six languages applicable to cloud service certification, and another two classes have been proposed as solution for remaining classes of the conceptual model [1].

P. Stephanow, et al., discussed about bottom-up approach. It uses low-level metrics available through widely delivered implementation of IaaS. Low level metric support validation of universal requirements from previous certificates such as, CSA STAR and ECSA etc. it is correlated to composite metrics development. Complex metrics similar to the certificate requirements they have defined continuous certification process for future work [2].

M. Alles, et al, identified the management of audit alarms and the prevention of alarms floods as it is critical task in CMBPS implementation process. In this paper they construct an approach to solve the problem of implementation of hierarchical structure of alarms. In this paper, only diverse practical experience will provide the facts necessary for identifying trade-offs between effectiveness, efficiency and timeliness of audit procedures and determining how to make CMBPC implementations worthwhile [3].

J. Woodroof and D. Searcy discussed about continuous audit implications technique. EDI is the Electronic Data Interchange import symbolic efficiencies and reductions in cost to supply chains. “Cold fusion” is used to design and demonstrated system that uses agents and alarm triggers that sent over internet to sequentially to observe actual values of client’s variable. [4].

S. Zawoad, et al.[5], suggested Secure-Logging-as-a-Service (SecLaaS) system. It saved virtual machine’s logs and gives access to forensic investigators by ensuring the confidentiality of cloud users. Proposed work preserves the proof of existing logs and it can protect integrity of logs from doubtful investigators. SecLaas is implemented to fetch logs from open stack and evaluate feasibility of proposed scheme.

J. R. Rajalakshmi, et al.[6], proposed homomorphic encryption scheme. It identifies the issues in secure cloud-based management. Anonymisation process is utilized for reduction of replaces the identity information from communication or record. This communication and records may be pseudonymous that same subject always have the replacement identity but it cannot be identified as separately. In future work they were planning to evaluate the performance overhead using alternative option such as, pair of partial homorphic algorithm. Also performance of system planned to calculate using anonymzing networks like, Ultra surf, Freegate etc.

H. Ye, Y. He. [7], proposed a continuous auditing model that utilizes web service technology. It leverages the power of XML and their related technologies. XML and web services can be utilized for the process of CA. XML web services include SOAP, WSDL, UDDI protocols those are discussed in this paper. Web services technology stay in auditee’s accounting system.

In future research work author looks towards a security mechanism. It automatically audit new risks faced in providing on-demand, real-time assurance.

S Schneider, et al, [8] concentrating on CC and ITO context. To identify determinant factors of sourcing decision in CC context they were concentrating on rich body of research on ITO. In this paper, authors inherits then set of determinant factors of cloud sourcing.

J. R. Kuhn,[9][10], proposed a continuous auditing ERP system. Development in ERP system provides critical infrastructure which required for effective evaluation of the assurance functions from periodic event to ongoing process through the merging of auditing process. An embedded auditing methodology is developed for combining continuous auditing functionality internally.

C.E Brown, et al.[11], introduced Continuous auditing is based on multiple research streams. They state that is made in this paper is that continuous auditing requires more than changes in hardware and software, it requires changes in the control environment and in the behavior of management and auditors. Vasarhelyi and Halper outline the key concepts and components of a continuous process auditing system.

N. Mahzan et al,[12], discussed about the use of computer-assisted audit techniques and tools (CAATTs) is a part of many professionally recommended audit procedures. This paper aims to argue that obtaining a better understanding of the factors underlying successful CAATTs adoptions would be helpful to aid wider development of these technologies in internal audit functions. In this paper author explores the successful adoption of GAS in ten cases to draw out the general factors that appear to be essential elements that lead to successful adoptions. From this basis, the paper proposes an initial model, built on existing theories of IT adoption more generally, as a theoretical basis for GAS adoption by decision-makers in an internal audit setting to better understand what may be essential factors to their adoption decisions to be likewise successful. Results suggest that two constructs from UTAUT (performance expectancy and facilitating conditions) appear to be particularly important factors influencing successful adoptions of GAS in this domain.

R. Nithiavathy [13], proposed rCloud computing technique. It gains the promotion in I.T. cloud computing world. It considered as the second thing after internet. They have proposed a mechanism of distributed integrity auditing that utilizes homomorphic token and erasure coded data for dynamically storing data. It allows TPA to audit cloud storage at very low computation cost. An efficient and dynamic operations i.e. update & delete are provided on block of data. Byzantine failure, malicious data modification attack, and even server colluding attacks the proposed technique is highly efficient and resilient. They to aim to achieve the goals like: Storage accuracy, data error in fast localisation and dynamic data support, lightweight etc. The proposed scheme accomplishes the integration of correctness of storage and corruption of data. For auditing procedure TPA allows cloud storage without expecting probability and time.

K. Yang, et al. [14], proposed efficient and privacy preserving protocol of auditing. It supports to data dynamic operations. They have extended their work to support batch auditing. The proposed approach generates the proof to solve the problem of data privacy. The proof is generated with challenge stamp and it utilizes the bilinear property such that auditor can only verify the correctness of proof without decrypting it. At the time of batch auditing process for multiple clouds proposed method does not required trusted organizer. The generated proof is serving as intermediate value of verification. In the process of auditing protocol contains the two-way communication such as: challenge and proof.

S. Lins, et al [15], enables data auditor to verify integrity of data, compliance in data and the dynamic infrastructure of cloud. To address the gap between continuous auditing a conceptualize architecture of CA has been introduced. It supports data auditor to to classify whether or not a high frequency auditing of their CSC criteria is needed. From the proposed approach CA, high level security as well as reliability is achieved in the cloud environment. But the methodologies to efficiently and continuously audit cloud services are remains immature.

3. PROBLEM DEFINITION

To design “Continuous public auditing scheme for regenerating cloud based secure cloud storage”.

4. PROPOSED SYSTEM

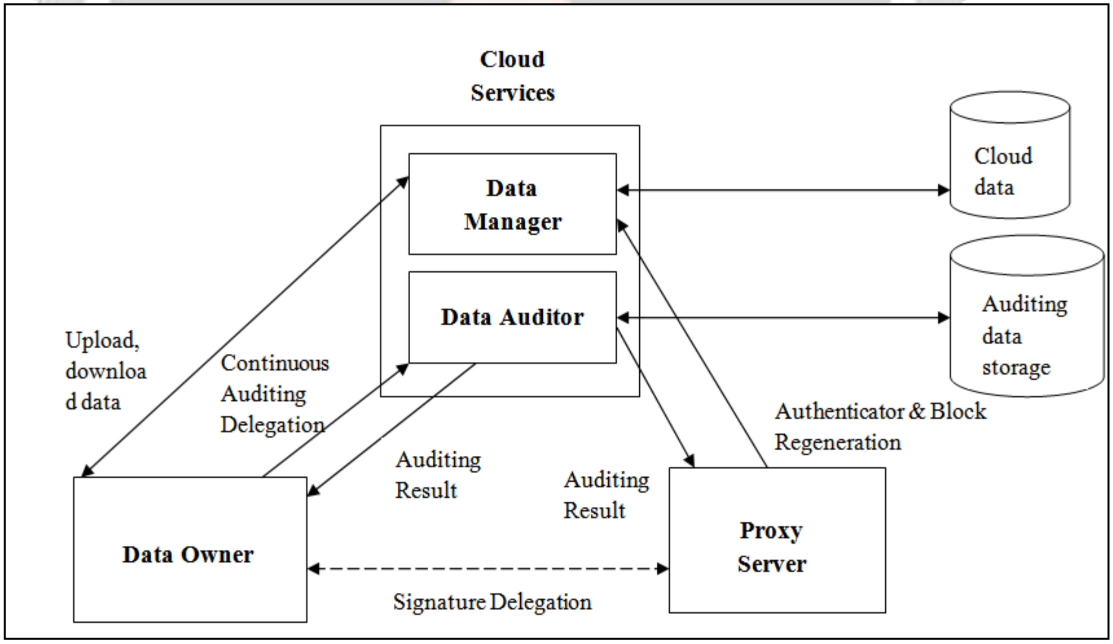


Fig -1: System Architecture

Fig. 1 represents the system architecture of proposed system. There are three types of entities are given in following figure such as, User who uploads as well as download the data on or from cloud. At the user end we proposed RSA algorithm for data encryption and decryption. In another entity such as cloud services, two types of users are presented i.e. data manager to handle the proper management of user uploaded data. For auditing of data we have proposed SHA-1 algorithm and the third entity is proxy, having purpose of authentication and block regeneration for that we have implemented Ramp secret sharing algorithm. Detail description of proposed system is given below in algorithmic steps.

a) Key Generation Algorithm:
This polynomial-time algorithm is run by the data owner to initialize its public and secret parameters by taking a security parameter κ as input.

b) Delegation Algorithm:
This algorithm represents the interaction between the data owner and proxy. The data owner delivers partial secret key x to the proxy through a secure approach.

c) Signature and Block generation algorithm:
This polynomial time algorithm is run by the data owner and takes the secret parameter sk and the original file F as input, and then outputs a coded block set Ψ, an authenticator set Φ and a file tag t.

d) Auditing Algorithm:
The cloud servers and TPA interact with one another to take a random sample on the blocks and check the data intactness in this procedure

e) Challenge:
This algorithm is performed by the TPA with the information of the file Finfo as input and a challenge C as output

f) Proof Generation:
This algorithm is run by each cloud server with input challenge C, coded block set Ψ and authenticator set Φ, then it outputs a proof P.

g) Verification:
This algorithm is run by TPA immediately after a proof is received. Taking the proof P, public parameter pk and the corresponding challenge C as input, it outputs 1 if the verification passed and 0 otherwise

h) Repair:
In the absence of the data owner, the proxy interacts with the cloud servers during this procedure to repair the wrong server detected by the auditing process.

i) Claim for Reparation:
This algorithm is similar with the Challenge () algorithm in the Audit phase, but outputs a claim for repair Cr.

j) Generate For Reparation:
The cloud servers run this algorithm upon receiving the Cr and finally output the block and authenticators set BA with another two inputs

K) Block and Signature Re-Generation:

The proxy implements this algorithm with the claim Cr and responses BA from each server as input, and outputs a new coded block set Ψ'' and authenticator set Φ' if successful, outputting $\perp$ if otherwise.

5. ALGORITHMS

5.1 RSA Algoritm

Input:

-2-large prime numbers p & q

-M: numeric block of plaintxts

Output:

-C: ciphertexts

-OM: Recovered plaintexts

Processing steps:

1. Select two prime numbers p & q to get product of p&q variable i.e. ‘n’
2. Find the totient of ‘n’
   T= (p-1)*(q-1)

3. Obtain list of possible integers which result into 1 mod totient
4. Calculate encryption and decryption time using exact two prime factors.
5. Encryption
6. Decryption

5.2 Shamir Secret Algorithm

Input:

- secret ’k’

- No. of parts ’n’

- Shares ’k’

Output:

a_0 i.e. secret’s’

Processing steps:

Phase 1:

1. Define two random numbers
2. Generate polynomial as, $f(x)=s+a_0^x+a_1^2$
3. Construct ’n’-points as,
   $Da_{x-1}=(x, f(x))$

4. Generate ’n’ shares and send
5. Select ’k’ shares i.e. $k(x_1,x_2,x_3)$

Phase 2:

1. Consider ’k’ shares i.e. $k(x_1,x_2,x_3)$
2. Such as,
   $(x_0, y_0)=x_1$

   $(x_1, y_1)=x_2$

   $(x_2, y_2)=x_3$

3. Compute ’L’ i.e. Lagrange Basis Polynomial.
   Such as, $l_0=(x-x_1/x_0-x_1)* =(x-x_2/x_0-x_2)$

   $L_1=(x-x_0/x_1-x_0)* =(x-x_2/x_1-x_2)$

   $L_2=(x-x_0/x_2-x_0)* =(x-x_1/x_2-x_1)$

   Therefore,

   $F(x)=\Sigma_{j=0}^{2} y_j.l_j(x)$

$=a_0+a_1+a_2$

5.3 SHA-1 Algorithm

Input:

-key, message m

Output:

Hash Value h

Processing:

```
1. ml = message length in bits
   Initialize h0 = 0x67452301,h1 = 0xEFCDAB89,h2 = 0x98BADCFE,h3 = 0x10325476,h4 = 0xC3D2E1F0

2. If characters $\le$ 8 bits Then
   Append bit 1 to message OR Add 0x80

3. for each chunk
   Break chunk into sixteen 32-bit big-endian words w[i], $0 \le i \le 15$

4. for i from 16 to 79
   w[i] = (w[i-3] xor w[i-8] xor w[i-14] xor w[i-16])

   leftrotate 1

   a = h0

   b = h1

   c = h2

   d = h3

   e = h4

5. for i from 0 to 79 If $0 \le i \le 19$
   Then

   f = (b and c) or ((not b) and d)

   k = 0x5A827999

6. Else If $20 \le i \le 39$ Then
   f = b xor c xor d

   k = 0x6ED9EBA1

7. Else If $40 \le i \le 59$ Then
   f = (b and c) or (b and d) or (c and d)

   k = 0x8F1BBCDC

8. Else If $60 \le i \le 79$ Then
   f = b xor c xor d

   k = 0xCA62C1D6
```

```
temp = (a leftrotate 5) + f + e + k + w[i]
e = d
d = c
c = b leftrotate 30
b = a
a = temp
h0 = h0 + a
h1 = h1 + b
h2 = h2 + c
h3 = h3 + d
h4 = h4 + e
hh = (h0 leftshift 128) or (h1 leftshift 96) or (h2 leftshift 64) or (h3 leftshift 32) or h4
```

6. MATHEMATICAL MODEL

System S can be defined as:

S = {U, C, DA, P} Where,

1. **U = {UI, UF, UO} is A User**

   UI= {UI1, UI2, UI3, UI4} ,A set of input

   UI1-User Authentication details

   UI2-File Data

   UI3-Update File request

   UI4-File Verification Request

   UF={UF1, UF2, UF3, UF4, UF5, UF6}, A set of functions

   UF1=User Registration

   UF2=User Login

   UF3=File Upload

   UF4=Edit File

   UF5=Verification Request

   UO={UO1,UO2,UO3}, A set of output

   UO1=Notification

   UO2=File Notification

UO3= Verification result

2. Cloud:

CI={CI1, CI2, CI3, CI4, CI5, CI6, CI7}, A set of input

CI1-User Registration Details

CI2-Block Data

CI3- Update File request

CI4=Restore request

CI5-Delete File request

CI6- Revoke User request

CI7-Challenge Message

CF= {CF1, CF2, CF3, CF4, CF5}, A set of functions

CF1= Register User

CF2=Login

CF3=Save Blocks

CF4=Generate Metadata

CF5=Generate Proof

CO=Output of cloud

CO= {CO1, CO2, CO3, CO4, CO5} ,A set of output CO1=Notification

CO2=Set of Blocks = {b1, b2..bn}

CO3=Metadata Notification

CO4=Update file notification

CO5=Generated Proof Notification

4. Data Auditor:

DAI= {DAI1, DAI2, DAI3, DAI4} , A set of input

DAI1-User Registration Details

DAI2-User Sig. Details

DAI3-Block Details

DAI4-Metadata of Block

DAF= {DAF1, DAF2, DAF3, DAF4, DAF5},A set of functions

DAF1=User Registration

DAF2=Validate User

DAF3=Save Metadata

DAF4=Generate Challenge

DAF5=Verify Proof

DAO= {DAO1, DAO2, DAO3},A set of output

DAO1=Notification

DAO2=Challenge Message

DAO3=Verification Message

5. Proxy:

P = {PI, PF, PO} is A Proxy Server

PI = {PI1, PI2}, A set of Input

PI1 = key

PI2= ClaimForRpair Request

PF = {PF1, PF2}, A set Of Function

PF1 = Generate new coded block set i.e. Regenerating Codes

PF2 = Get Regenerating codes

PO ={PO1}, A set Of Output

PO1 = Set Of Re-generating Codes

7. EXPERIMENTAL SETUP

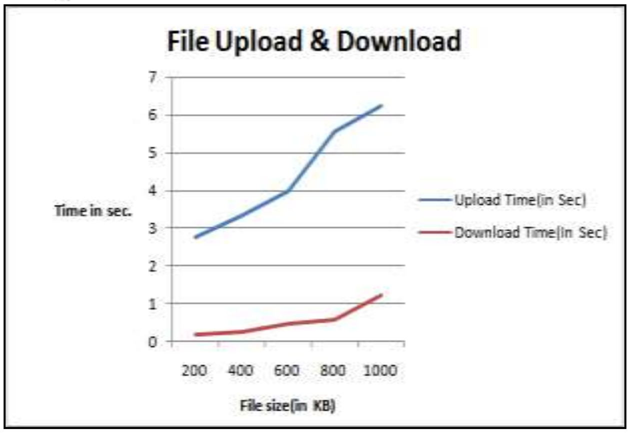
Client server system is developed using apache tomcat-6 and jdk 1.7 with mysql 5.3 as a database. Core I5 – machine with 4GB RAM is used for development and testing. Cloud server and proxy server are 2 server side entities created using Eclipse IDE as a server side project. Data owner entity is created using Netbeans-IDE.

Synthetic Dataset: Text based document verification and data regeneration is proposed this system. Various text files are collected from different sources of various sizes form 1 KB to 1 MB. For Text files following extensions are allowed: .txt, xml, java, .cs, .json, .csv

8. RESULT TABLES AND DISCUSSION

Table-1: Efficiency Evaluation

File Size (In KB)	Upload Time(in Sec)	Download Time(In Sec)
200	2.76	1.12
400	3.34	1.76



600	3.98	3.02
800	5.56	3.66
1000	6.23	3.99

Chart-1: Efficiency evaluation

Table I represent the time analysis of file upload and download. To test system performance for file uploading and downloading files size from 200KB to 1000KB are used. The readings are taken into seconds. As per observations, time required for file uploading is more than time required to file download. File uploading time involves file encryption as well as metadata generation time.

Chart 1 depicts efficiency evaluation in graphical format. X-axis represents the file size in KB whereas, Y-axis represents the time in seconds.

Table-2: Metadata generation Time

File Size (in KB)	Metadata Generation Time(in milisec.)
200	22
400	34
600	36
800	42
1000	49

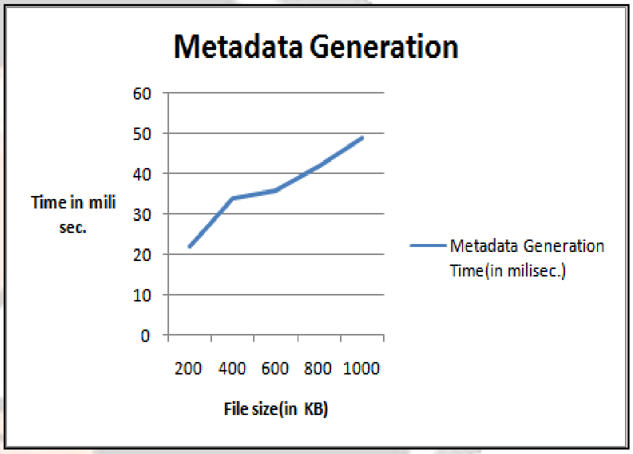


Chart-2: Metadata Generation time

Table 2 represents the evaluation of time for file metadata generation. It is given in milliseconds.

Chart 2 represents the performance of file metadata generation in graphical format. In this, X-axis represents file size in KB whereas, Y-axis represents the time in milliseconds

Table-3: File verification & Regeneration

File (in KB) size	Time req. for verification (in milisec.)	Time req. for Regeneration (in milisec.)
200	29	17
400	41	29
600	44	31
800	50	41
1000	56	45

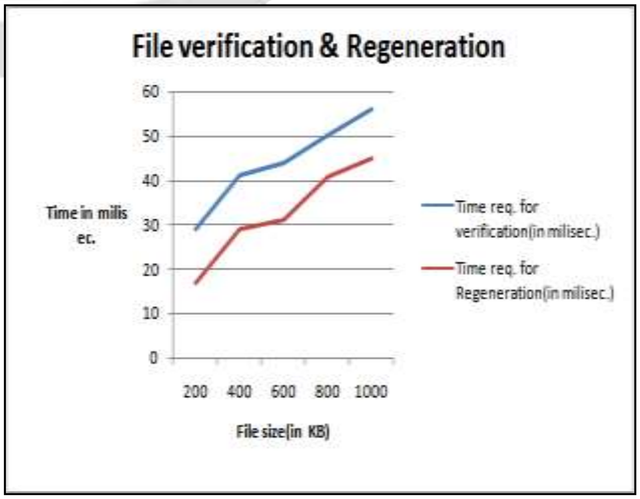


Chart-3: File verification & code re-generation

Table 3 represents the time evaluation for file data verification and data regeneration. To test the system performance in terms of file verification as well as data regeneration, we have used files of size 200KB to 1000KB. As per reading observation from table III, time required for file verification is more than time required for file data regeneration as system contributes continuous data auditing after file uploading to cloud server. Regeneration is required only if there exist some mismatch between file metadata and verification result.

Chart 3 represents the graphical format of file verification and data regeneration. It contains X-axis and Y-axis. X-axis represents the file size in KB whereas, Y-axis represents the time in millisecond.

9. CONCLUSIONS

Continuous cloud auditing service is proposed to determined many business applications which demanding for reliable cloud services for data as well as their personal privacy and security perspective. It focused on continuous auditing of cloud services which provides guarantee of high level security as well as reliability. Proposed system contributes data restoration facility in case of any data damage or corruption or leakage etc.

10. ACKNOWLEDGEMENT

11. REFERENCES

[1] P. Stephanow and M. Gall, “Language Classes for Cloud Service Certification Systems”, in 2015 IEEE 11th World Congress on Services (SERVICES), 2015.

[2] P. Stephanow and N. Fallenbeck, “Towards continuous certification of Infrastructure-as-a-service using low-level metrics”, in Proc. ATC, Beijing, China, 2015.

[3] K. Singh, P. J. Best, M. Bojilov, and C. Blunt, "Continuous auditing and continuous monitoring in ERP environments", Inf. Syst. J, vol. 28, no. 1, pp. 287–310, 2013

[4] J. Woodroof and D. Searcy, “Continuous audit implications of internet technology”, in Proc. HICSS, Outrigger Wailea Resort, Island of Maui, 2001, pp. 1–8.

[5] S. Zawoad, A. K. Dutta, and R. Hasan, “SecLaaS”, in Proc. ASIA CCS, Hangzhou, China, 2013, pp. 219–230.

[6] J. R. Rajalakshmi, M. Rathinraj, and M. Braveen, “Anonymizing log management process for secure logging in the cloud”, in Proc. ICCPCT, India, 2014, pp. 1559–1564.

[7] U. S. Murthy and S. M. Groomer, "A continuous auditing web services model for XML-based accounting systems", International Journal of Accounting Information Systems, vol. 5, no. 2, 2004.

[8] A.Sunyaev and S. Schneider, "Cloud services certification", Commun ACM, vol. 56, no. 2, pp. 33–36, 2013.

[9] Kuhn Jr, John R. and S. G. Sutton, "Continuous auditing in ERP system environments", Inf. Syst. J, vol. 24, no. 1, pp. 91– 112, 2010.

[10] K. Singh, P. J. Best, M. Bojilov, and C. Blunt, "Continuous auditing and continuous monitoring in ERP environments", Inf. Syst. J, vol. 28, no. 1, pp. 287–310, 2013.

[11] C. E. Brown, J. A. Wong, and A. A. Baldwin, "A review and analysis of the existing research streams in continuous auditing", Journal of Emerging Technologies in Accounting, vol. 4, no. 1, 2007.

[12] N. Mahzan and A. Lymer, "Examining the adoption of computer-assisted audit tools and techniques", Managerial Auditing Journal, vol. 29, no. 4, pp. 327–349, 2014.

[13] R. Nithiavathy, “Data integrity and data dynamics with secure storage service in cloud”, in Proc. PRIME, Salem, Germany, 2013

[14] 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.

[15] S. Lins ,S. Schneider, A. Sunyaev, "Trust is Good, Control is Better: Creating Secure Clouds by Continuous Auditing ", IEEE Trans. On cloud computing, vol. 27, no. 9, pp. 1717–1726, Jan 2016.