International Journal of Future Generation Communication and Networking
Vol. 13, No. 2, 2020 pp.1614-1628

Verification and Protection of Big data Integrity in Cloud Computing (VPBICC)

[1]Abid, *Max Tech Developers Mingora, Pakistan,* abidbaseer@ymail.com , abidbaseer10@gmail.com
[2]Mohammad Faisal, *Department of CS & IT, University of Malakand, Pakistan* mfaisal@uom.edu.pk / mfaisal_1981@yahoo.com
[3]Rizwan Munir *Max Tech Developers Mingora, Pakistan* rizwanmtci@gmail.com
[4]Sajid Hussain, *Max Tech Developers Mingora, Pakistan* sajid8188@gmail.com
[5]Habibullah, *Department of statistics, University of Malakand,* habibullah858@gmail.com

Abstract

Data verification and protection are challenging issues in cloud computing. Multiple integrity schemes tried to address these issues. But none of the literature work get successful to get a single solution for both problems. Here we propose a novel scheme which can provide big data integrity proof (Verification) and secure big data storage (Protection) in cloud environment in a single paradigm. Our scheme will produce effective results w.r.t execution time as well.

1. Introduction

Data integrity is a key aspect for big data. The data cannot be modified by unauthorized parties to prevent from misuse and only modify by owner or authorized party is called integrity. To manage and store big data in cloud centre the cloud base application provide the data integrity feature for the user [1]. For any type of data and context related to computation the data integrity is very important factor. Data integrity provide quality of services and also very important in privacy and security [2].

In cloud centre to ensure the user data correctness is one of the must addressed main challenge and it is the responsibility of cloud that provide structure for user to check that their data is secure [1][3]. Many users and firm understand that their economic advantage their big data outsourcing to cloud storage servers. It is mean that the data owner moves their big data to cloud storage server which is manage by third party and the data owner will pay free for it and when the data is required back to data owner the third party will provide it back. Data outsourcing in cloud storage for small firm (organization) is costly because of updating their hardware and it is also difficult task to maintain the data. To store big data in cloud storage server also help the firms to minimize the cost of storage as well as cost of maintenance and also reduce chance of losing of data from hardware failure by keep many copies of the user data [4]. To store big data in cloud storage server is also many security concern (For find efficient solution it is to be investigated) one of the main problem in integrity. To ensure that how securely and efficiently the server of cloud storage returns required data to use. Similarly data which is return correct and complete in the response of their clients [5].

Increasing in the volume of data is called big data. When the volume of data is increase so for traditional technology it storing, processing and analysing is become difficult. For uncovering large hidden data value from huge amount of data that are divergent, complicated and of large scale the set of technologies in the new integration form is called big data [1][6] . For utilizing this large amount of scientific data for visualization is also called big data [7]. The big data is also referred to the volume of data that just away of the capability of the technology to manage, process and store in well-organized way [8].

One of the major important moves to modern technology is cloud computing and it becomes important architecture to complete massive-scale and complex computing. Cloud computing delivers a good platform to store big data [1]. Recently cloud platform is offering a diverse model for the user which are (i) Software as a Service (SaaS): A software Application running on cloud infrastructure and this

ISSN: 2233-7857 IJFGCN

application used by various clients through a thin client interface such as Web Browsers. (ii) Platform as a Service (PaaS): The cloud service provider offers a specific interface inside this model and the client works in this interface it is include some software, Programming languages and hosting. (iii) Infrastructure as a Service (IaaS): inside this Service model/technique the providers of cloud service offers some hardware to client and the client use the hardware for their purpose and it is include processors, storage and network etc. the client have not control on these hardware but control through some operating system[9][10].

Our scheme is combination of two schemes one of these scheme is "Data Integrity Proofs in Cloud Storage". If data owner want to store their big data in cloud so this scheme will provide the facility of data integrity verification. The data owner gets the proof that their big data is in correct form. But shortcoming of this scheme is it not provides the facility to secure big data from deletion or modification. So for this purpose the other scheme "Secure Big Data Storage and Sharing Scheme for Cloud" is used. This scheme provides the facility to secure the big data from deletion or modification. But shortcoming of this scheme is it not provides the facility of big data integrity verification.
In this research work proposed a novel scheme which are combination of two scheme and this scheme have two very important feature one is provide the integrity proof to data owner if he wish to store their big data in cloud data centre and the other feature is its provide a mechanism inside cloud computing for storage of big data security. With the help of this novel scheme some integrity issues of big data in cloud data centre is solved.

2. Literature review

Provable data possession is the first group which is responsible for outsourced big data integrity in cloud computing. There are four main phases in this method. (i) Setup (ii) Challenge (iii) Verification (iv) Proof [11]. The integrity verification of stored big data without retrieving big data in cloud by usage of RSA-Holomorphic Verifiable Tag(HVT) two provably secure scheme depends on server constructed proof HVT give permission to server have certain block even when the block is not accessible for the client[12]. First model in which the robust feature is added is Secure PDP which is also robust data possession guarantee which is succeed through added strong feature to PDP(Provable data possession) depend on Spot-checking technique. So the data owner will be responsible for the cost of computation. By assuming the possession of combined block to reduce computation cost they suggest an efficient PDP (E-PDP) [13]. Another PDP method which is depends on elliptic curves cryptosystem in this method the third party/data owner have ability to examine data remotely. Behind this method the main idea is producing a similar tag for concurrent public and private verifiability by identify for every block vector therefore data blocks is includes in the input file [14], [15].In All PDP method the TPA or big data owner is continuously in duty of outsourced big data examine but some situation like connection of internet is not available such in jail, battlefield, on the ocean vessel. On the other side without passing the verification step from owner the TPA have not ability to accomplish independently checking of data remotely[16][17]. To overcome this problem proxy PDP (P-PDP) method is proposed this method use bi-linear pairing mechanism in which according to a warrant the task checking of data integrity remotely is delegated to proxy[18][19].

Proof of Retrievability (POR) is another technique which is kind of cryptographic Proof of Knowledge (POK) without having a file to be download from the untrusted cloud to guarantee the integrity as well as privacy of the big data stored in the cloud server by using Forward Error-correctness Code it provide the feature of data recovery and reduce data corruption when a significant fraction of file is not corrupt then there will be capability with verifier that the file can recover by spot-checks. Client stored data completely on the server in POR approach. But The PDP base ensure that mostly data of client are stored in server and may be chance that some portion of data lost from the server. POR can differentiate from PDP on the security approach[20][21]. First POR model is proposed which is basis on the sentinel blocks. In this model before data blocks storing in cloud the sentinel blocks are hide in the other data blocks one

ISSN: 2233-7857 IJFGCN

way hash function has been used to compute sentinel blocks. Intact sentinel blocks will only require checking for verifier. On the number of the sentinel block embedded the amount of challenges is not enough is the main disadvantage of this scheme [22][23]. To increase the challenges in term of number and to improve security protocol and efficiency another POR scheme is proposed which is called Compact Proof of Retrievability (Compact POR)[24]. Using a constant size polynomial commitment technique with constant communication a new POR scheme called Public Proof of Retrievability is proposed. For analysing the cloud a polynomial with small string as a proof is generate by the polynomial commitment scheme to help the prover. And this is reducing the communication cost of POR scheme [25][26]. To select randomly indices of input files as challenge is use to reduce the challenge message complexity of Public POR sachem [27][28].

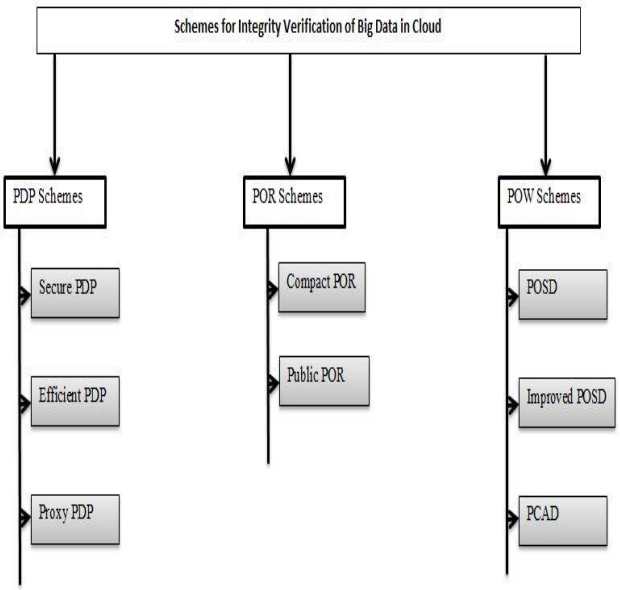


Figure 1 Big data integrity verification schemes in cloud

To provide efficient storage in term of cost and remove duplicates copies of data and redundancy single instance storage (data compression) technique is use which is called data De-duplication[29][30]. In such storage system which supporting data de-duplication a client give a copy of hash to server to give proof to server that client have a copy of outsourced data. If it is finding in database by server the server accept the client otherwise ask him for entire file uploading. This method is not good in attack of security because if someone gets the hash file its mean he is permitted to access the file[31][32][33]. Another de-duplication scheme in which the clients have ability to give proof to server without upload the entire file and consider these security issues is proposed. Mark Hash Tree and Collision-Resistant Hash function are used to construct this method and this called Proof of Ownership. In this scheme role of the verifier and proverb is reversed because the clients need to persuade the cloud [31]. To provide both security and integrity the two schemes Proof of Ownership and Proof of Data Possession are integrated and proposed the first one Proof of Storage with DE duplication method (POSD) scheme. In this method there are four phases which are “Key generation”, “Uploading”, “Auditing” and “De-duplication” [34]. To control key generation step by minimizing the client capability improved POSD (I-POSD) schemes is proposed[35]. To address the computation and linear communication cost proposed new data storage auditing that has capability of De-duplication which depends on polynomial based tag and homomorphic-linear authenticators. Public cloud

ISSN: 2233-7857 IJFGCN

auditing with De-duplication which is complexity of auditing in this method depend on transforming an challenge message [36][37].

To provide secure data storage in cloud for this security purpose of data in cloud there are already some existing schemes which are mostly designed on the basis of costly algorithm which are not efficient to apply for big data. We will discuss some of these schemes here [38][39].

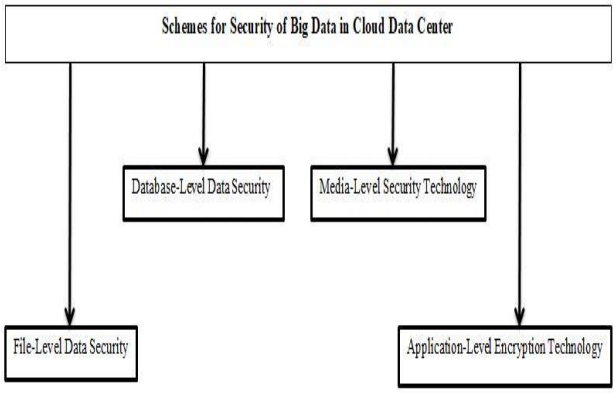


Figure 2 Schemes for Security of Big Data in Cloud Data Centre

File-level big data security scheme which can implement on the host this scheme can cause problem in performance in some application. In data backup operation mostly for the database back in this scheme have also some weakness [40]. The other scheme use for big data security has mainly focused on database level security when the data store in data base this scheme can apply. In this mechanism the data in each column of the database is encrypted and also knows as column-level encryption. When the companies stored their sensitive data in one or two column in data base in that situation this scheme is very economical. In this scheme the encryption is done by software not by hardware so it will effect on performance [41]. Media-level security technology is the other scheme use for data security in data centre. On storage devices such as hard disk and tape encryption of static data is includes in this schemes. For the user and applications high degree transparency is provide by media-level encryption. Media-level encryption will protect data after reaching and against the theft of physical storage device and it will not protect the data during transmission [42]. Another scheme use for security is application-level data security. end-to-end encryption solution provide by this scheme it ensure that the data access by certain users through some application. To maintain many parameters and structures this scheme is become very costly [43].

Table 1 Algorithmic complexities of the literature review

S.no	Title and Author	Pros	Cons	Algorithmic complexity
1	Provable data possession at untrusted stores. (G. Ateniese *et al.*)	Verification of stored big data without retrieving big data	Completely not provide secure data possession	O(mn)
2	Provable data possession at untrusted stores. (G. Ateniese *et al*)	Robust feature is added. Depend on Spot-checking technique	Computation Overhead is High	O(t)
3	Remote data checking using	Reduce computation cost.	Computation	O(t)

ISSN: 2233-7857 IJFGCN

	provable data possession. (G. Ateniese *et al*)	depends on elliptic curves cryptosystem	Overhead is High	
4	Proxy provable data possession in public clouds. (H. Wang)	Use bi-linear pairing mechanism in data integrity can check remotely	Information is lacking	O(t log n)
5	PORs: Proofs of retrievability for large files. (A. Juels and B. S. Kaliski Jr)	Without having a file to be downloaded from the untrusted cloud to guarantee the integrity as well as privacy of the big data stored in cloud.	The number of Challenges is limited for the embedded sentinel block	O(mn)
6	Compact proofs of retrievability. (H. Shacham and B. Waters)	The number of Challenges is increased	Just Appropriate for Private auditing	O(mn)
7	Proofs of retrievability with public verifiability and constant communication cost in cloud. (J. Yuan and S. Yu)	Reduce the communication cost of POR scheme	Only for static environment the efficiency in this is approved	O(t)
8	Proof of Ownership in Remote Storage Systems PoW Solution : A General Protocol Security-Efficiency Tradeoff. (S. Halevi, D. Harnik, B. Pinkas, and E. Haim)	Using Mark Hash Tree and Collision-Resistant Hash function.	Not Support Integrity if someone gets the hash file its mean he is permitted to access the file.	O(t log n)
9	Secure and efficient proof of storage with deduplication. (Q. Zheng and S. Xu)	Provide the integrity and also the security	Validity of key generation linear communication and Computational cost on the client	O((m+n)t)
10	Secure proof of storage with deduplication for cloud storage systems (Y. Shin, D. Koo, J. Hur, and J. Yun)	Control key generation step by minimizing the client capability	Lot of communication and Computational cost on the client	O(mn)

3. Methodology

In this section we will validate our scheme through algorithmic, programing, statistical meaning.

3.1 Algorithmic evaluation

Phase I

In first phase we perform the process of division of data into data blocks, selection of key, Generation of Metadata for each block, XOR of metadata with key and append the result of XOR with the data block these overall process will be perform through some steps which are following.

Step 1: In this step we will distribute data into data blocks. Data will be distributed with respect of pages. Calculate the total number of pages and divide by half the result will be the total number of blocks.

Step 2: In this step we will select the key which will be used in the process of encryption. Convert the key into binary form.

ISSN: 2233-7857 IJFGCN

Step 3: The metadata will be generated for each block in this step and the will be store in some particular position in each block which will be only known to verifier.
Step 4: Encryption of metadata will perform in this step. For encryption of metadata will take the XOR of metadata with the selected key.
Step 5: The encrypted metadata will append with each of their block.
Step 6: Concatenate all data blocks which have consist encrypted metadata and generate a new data file.

Phase II

In second phase of this scheme the file of big data is separated into different many sequenced parts before storing and then store these different parts into different storage server owned by different providers. The division of big data into different parts is performing under some principles.

Phase III

In third phase of this scheme is if the client / verifier want checking data integrity first of all big data parts which are stored in different data centre will be collected together and then it is restored into their original form on the basis of their sequence number and after this the verifier throw a challenge to the archive and ask for respond. After the response the response and the challenge is compared and after comparison the verifier/client decide about the verification of the integrity if the response is correct in then its mean the integrity is not lost.

3.2 Programing evaluation / Code

```
Static String generateRandomString(int n, String string) {
    // create StringBuffer size of strig
    StringBuilder sb = new StringBuilder(n);
    for (int i = 0; i < n; i++) {
       int index = (int) (string.length() * Math.random());
       sb.append(string.charAt(index));        }
    return sb.toString();    }
  static String encriptionXor(String data, char c) {
    String enc = "";
    for (int i = 0; i < data.length(); i++) {
       enc = enc + (data.charAt(i) ^ c);        }
    return enc;    }
 public static void main(String[] args) {
 String data = "the quick brown fox jumps fox fox over the lazy dog brown";
    String blocks[] = new String[11];
    for (int i = 0; i < 11; i++) {
       blocks[i] = "";        }
    int datalength = data.length();
    int blockdatasize = datalength / 10;
    int blockindex = 0;
    ///  code for dividing data in blocks starts
    for (int i = 0; i < data.length(); i++) {
       if (blockindex > 9) {
       } else if (blocks[blockindex].length() == blockdatasize) {
          blockindex++;}
       blocks[blockindex] = blocks[blockindex] + data.charAt(i);        }
    ///  code for dividing data in blocks ends
    ///  code for creating mata data from each block  starts
   int mataDataSize = Integer.parseInt(JOptionPane.showInputDialog("Enter size for mata data : "));
    String mataData[] = new String[11];
```


ISSN: 2233-7857 IJFGCN

```
for (int i = 0; i < mataData.length; i++) {
   mataData[i] = generateRandomString(mataDataSize, blocks[i]);       }
/// code for creating mata data from each block  ends
/// code for creating mata data from each block  starts
String key = JOptionPane.showInputDialog("Enter key for encryption : ");
String encryptedMataData[] = new String[11];
for (int i = 0; i < encryptedMataData.length; i++) {
   encryptedMataData[i] = encriptionXor(mataData[i], key.charAt(0));       }
/// code for creating mata data from each block  ends
/// code for joining/combining encrypted data with each data block starts
String encryptedDataWithBlocks[] = new String[11];
for (int i = 0; i < encryptedDataWithBlocks.length; i++) {
   encryptedDataWithBlocks[i] = blocks[i] + encryptedMataData[i]; }
/// code for joining/combining encrypted data with each data block ends
/// code for joining/combining all new data blocks again starts
String OvarallDataWithEncryption = "";
for (int i = 0; i < encryptedDataWithBlocks.length; i++) {
   OvarallDataWithEncryption = OvarallDataWithEncryption + encryptedDataWithBlocks[i];       }
//code for joining/combining all new data blocks again ends
/// code for spliting new generated data for servers starts
int dataSize = Integer.parseInt(JOptionPane.showInputDialog("Enter size to split data for different
servers : "));
int OvarallDataWithEncryptionLength = OvarallDataWithEncryption.length();
String OvarallDataWithEncryptionBlocks[] = new String[dataSize];
for (int i = 0; i < dataSize; i++) {
   OvarallDataWithEncryptionBlocks[i] = "";       }
int ovarallblockdatasize = OvarallDataWithEncryptionLength / dataSize;
int arrindex = 0;
for (int i = 0; i < OvarallDataWithEncryptionLength; i++) {
   if (arrindex != (dataSize - 1)) {
      if (OvarallDataWithEncryptionBlocks[arrindex].length() == ovarallblockdatasize) {
         arrindex++; }          }
   OvarallDataWithEncryptionBlocks[arrindex] = OvarallDataWithEncryptionBlocks[arrindex] +
OvarallDataWithEncryption.charAt(i);       }
/// code for spliting new generated data for servers ends
// code to display actual data starts
System.out.println("****** Step 1: Actual Data **********************");
System.out.println(data + "\n");
// code to display actual data ends
// code to display actual data blocks starts
System.out.println("******  Step  2:  Actual  Data  Divides  In  10  or  11  Blocks
**********************");
for (int i = 0; i < blocks.length; i++) {
   System.out.println(blocks[i]);       }
System.out.println("");
// code to display actual data blocks ends
// code to display mata data starts
System.out.println("******  Step  3:  All  Mata  Data  From  Each  Data  Block
**********************");
for (int i = 0; i < mataData.length; i++) {
   System.out.println(mataData[i]);  }
```


ISSN: 2233-7857 IJFGCN

```
System.out.println("");
// code to display mata data ends
// code to display Encrypted mata data starts
System.out.println("****** Step 4: All Encrypted Data Generated From XOR With Key " + key + " and Each Mata Data Block  **********");
for (int i = 0; i < encryptedMataData.length; i++) {
   System.out.println(encryptedMataData[i]);       }
System.out.println("");
// code to display Encrypted mata data ends
// code to display combined data blocks with encrypted data starts
System.out.println("****** Step 5: Combine Again Each Data Block with Encrypted Data Block **********");
for (int i = 0; i < encryptedDataWithBlocks.length; i++) {
   System.out.println(encryptedDataWithBlocks[i]);       }
System.out.println("");
// code to display combined data blocks with encrypted data ends
// code to display over all data with encrypted data starts
System.out.println("******  Step  6:  Combine  Again  All  New  Data  Blocks **********");
System.out.println(OvarallDataWithEncryption);
System.out.println("");
// code to display over all data with encrypted data ends
// code to display final data blocks for severs starts
System.out.println("****** Step 7: Finaly Agian Divide New Generated Data in Blocks For Different Servers  **********");
for (int i = 0; i < OvarallDataWithEncryptionBlocks.length; i++) {
   System.out.println(OvarallDataWithEncryptionBlocks[i]);       }
System.out.println("");}
// code to display final data blocks for severs ends
```

3.3 Statistical evaluation

As per the following specifications the results are shown in table

Computer System Details:

System: Core i5 3rd generation
Processor: Intel(R) Core(TM) i5-3337U CPU @ 1.80GHZ
Ram: 8GB

Table 2 VPBICC algorithm testing details

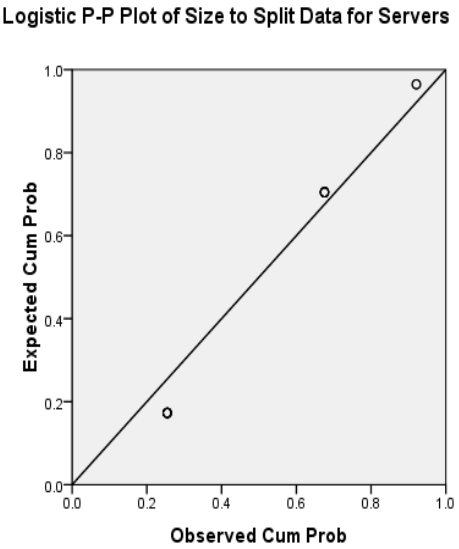
Number of Words	Size for Meta Data	Number of Keys For Encryption	Size to Split Data for Servers	Build and Execution Time
10	5	5	5	2 Sec
50	5	5	5	2 Sec
50	10	10	10	2 Sec
150	5	5	5	3 Sec
150	10	10	`0	3 Sec

ISSN: 2233-7857 IJFGCN

300	5	5	5	4 Sec
350	5	5	5	4 Sec
400	10	10	10	4 Sec
500	10	10	10	4 Sec
1000	5	5	5	4 Sec
1000	10	10	10	5 Sec
1000	20	20	20	5 Sec
1500	5	5	5	5 Sec
1500	10	10	10	5 Sec
1500	20	20	20	5 Sec

The following statistical results are achieved after applying the logarithmic function on SPSS tool using PPlot procedures.

Table 3 Model Description



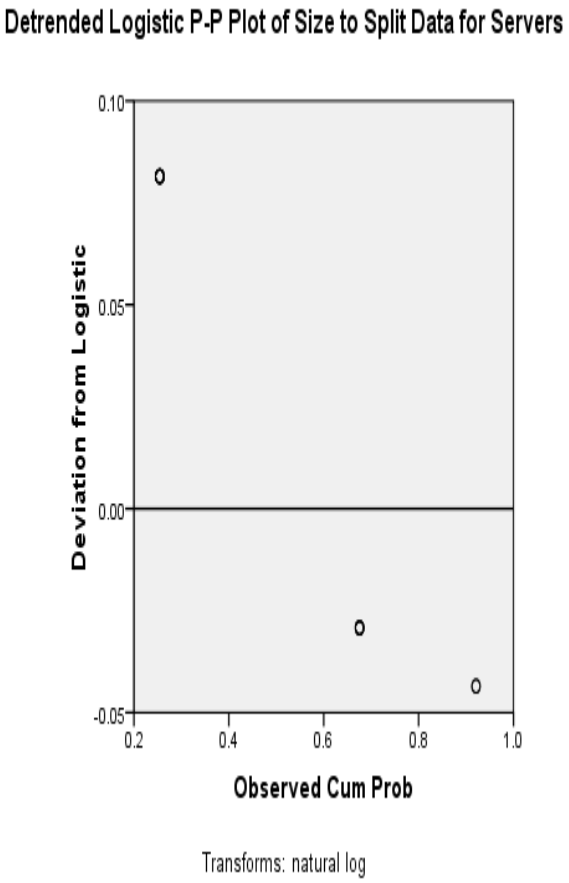
Model Name		MOD_2
Series or Sequence	1	Number of Words
	2	Size for Meta Data
	3	Number of Keys For Encryption

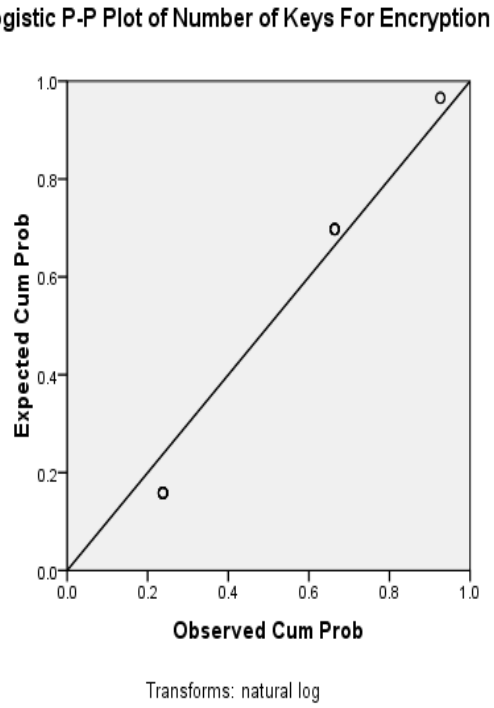
ISSN: 2233-7857 IJFGCN

	4	Size to Split Data for Servers
	5	Build and Execution Time
Transformation		Natural logarithm
Non-Seasonal Differencing		0
Seasonal Differencing		0
Length of Seasonal Period		No periodicity
Standardization		Applied
Distribution	Type	Logistic
	Location	estimated
	Scale	estimated
Fractional Rank Estimation Method		Blom's
Rank Assigned to Ties		Mean rank of tied values

Applying the model specifications from MOD_2

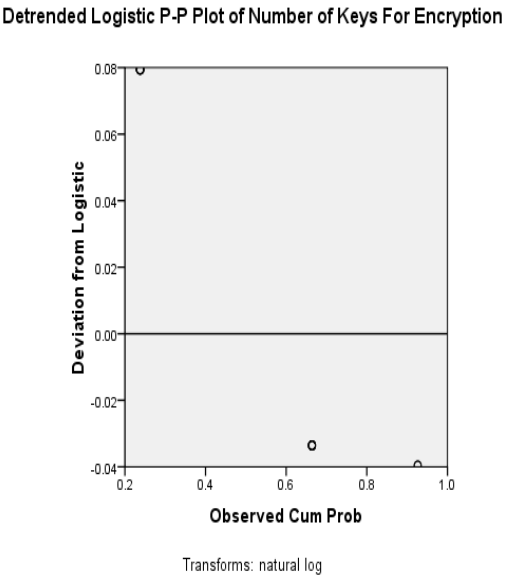
Size to Split Data for Servers



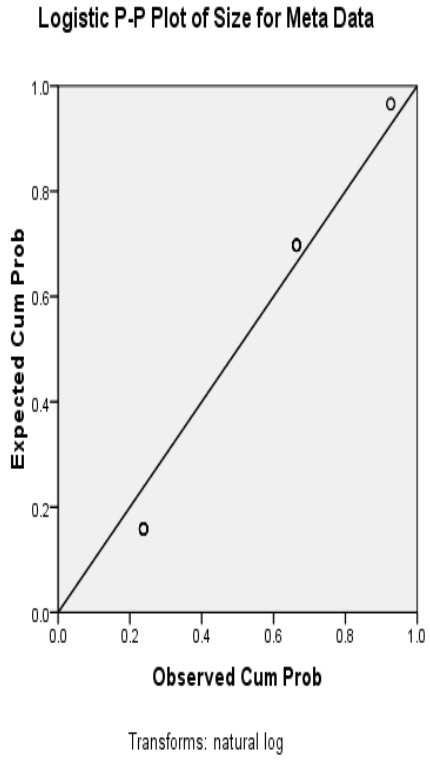


Number of Keys for Encryption

ISSN: 2233-7857 IJFGCN

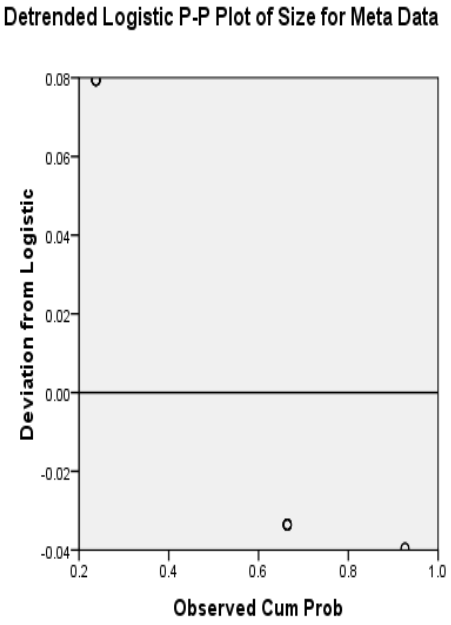


Size for Meta Data



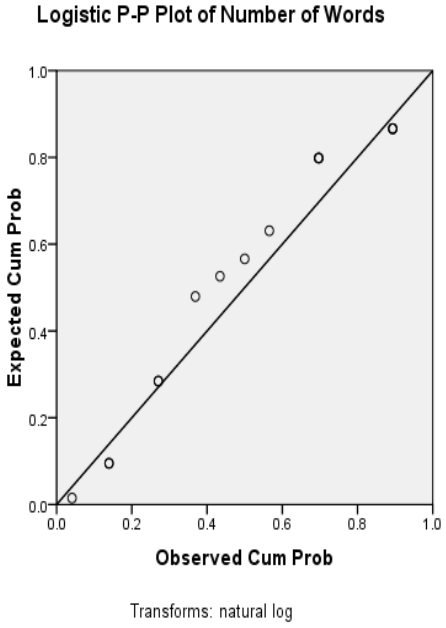
q

ISSN: 2233-7857 IJFGCN



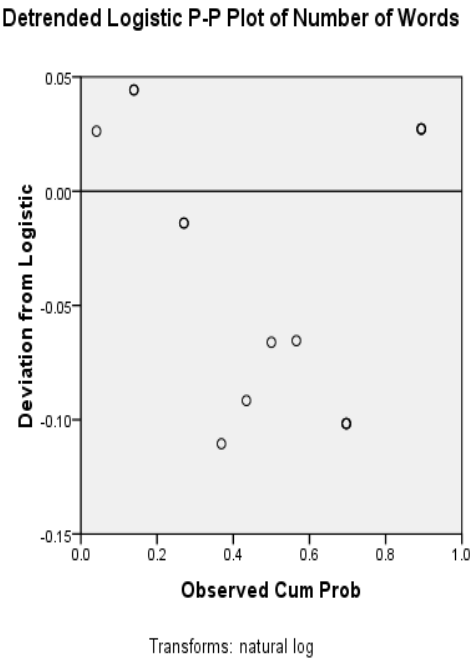

`````

## Number of Words






ISSN: 2233-7857 IJFGCN
`````



The advantages of this scheme are:

- There is no such scheme exist yet in the current literature survey we conducted which can perform verification and protection at once. Here in our scheme the verification and protection will be performed and by a single novel scheme.
- Other schemes will first verify the data separately and then protect the data separately by mean of two different schemes as discussed in literature. While we will do both steps in one scheme.
- Our scheme will reduce the computation time as compare to other schemes which are using two schemes separately for verification and modification

4. Conclusion and future work

The big data in this scheme is divide into different parts and stored in different cloud storage centre so there may be a chance that the data may lost. In this scheme the big data stored in different cloud data centre with different provide there may have chance that someone cloud data storage provide involved in fraud. The maintenance effort is reduce because this is combination of two schemes. The maintenance cost is reducing because this scheme provides two feature in one time so the client do not need to pay for different cloud storage server.

5. References

[1] I. A. T. Hashem, I. Yaqoob, N. B. Anuar, S. Mokhtar, A. Gani, and S. Ullah Khan, "The rise of 'big data' on cloud computing: Review and open research issues," *Inf. Syst.*, vol. 47, pp. 98–115, 2015.
[2] C. Liu, C. Yang, X. Zhang, and J. Chen, "External integrity verification for outsourced big data in cloud and IoT: A big picture," *Futur. Gener. Comput. Syst.*, vol. 49, pp. 58–67, 2015.
[3] P. R. Kumar, P. H. Raj, and P. Jelciana, "ScienceDirect Procedia Computer Science Exploring Data Security Issues and Solutions in Cloud Computing," *Procedia Comput. Sci.*, vol. 125, no. 2009, pp. 691–697, 2018.

ISSN: 2233-7857 IJFGCN

[4] N. Manikandan, U. Senthilkumaran, and S. Prasnna, "Data integrity proofs in cloud storage," *Int. J. Appl. Eng. Res.*, vol. 9, no. 21, pp. 11193–11200, 2014.
[5] E. Mykletun, M. Narasimha, and G. Tsudik, "Authentication and integrity in outsourced databases," *ACM Trans. Storage*, vol. 2, no. 2, pp. 107–138, 2006.
[6] I. Song and Y. Zhu, "A rticle," vol. 33, no. 4, 2018.
[7] M. Cox and D. Ellsworth, "Managing Big Data for Scientific Visualization," *ACM Siggraph*, vol. 97, no. January, pp. 1–17, 1997.
[8] J. Manyika, M. Chui, B. Brown, and J. Bughin, "References," no. May, p. 2011, 2011.
[9] W. Zhang *et al.*, "Building Intelligent Transportation Cloud Data Center Based on SOA," vol. 8, no. 2, 2017.
[10] A. Jula, E. Sundararajan, and Z. Othman, "Cloud computing service composition: A systematic literature review," *Expert Syst. Appl.*, vol. 41, no. 8, pp. 3809–3824, 2014.
[11] M. Sookhak *et al.*, "Remote Data Auditing in Cloud Computing Environments," *ACM Comput. Surv.*, vol. 47, no. 4, pp. 1–34, 2015.
[12] G. Ateniese *et al.*, "Provable data possession at untrusted stores," *Proc. 14th ACM Conf. Comput. Commun. Secur. - CCS '07*, p. 598, 2007.
[13] G. Ateniese *et al.*, "Remote data checking using provable data possession," *ACM Trans. Inf. Syst. Secur.*, vol. 14, no. 1, pp. 1–34, 2011.
[14] C. Hanser and D. Slamanig, "Efficient Simultaneous Privately and Publicly Verifiable Robust Provable Data Possession from Elliptic Curves," *10th Int. Conf. Secur. Cryptogr. (SECRYPT 2013), Reykjavik, Iceland, 29-31 July 2013. Note This is full version which is available as Cryptol. ePrint Arch. Rep. 2013/392*, pp. 15–26, 2013.
[15] T. Icart, "How to hash into elliptic curves," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5677 LNCS, pp. 303–316, 2009.
[16] W. Shen, G. Yang, J. Yu, H. Zhang, F. Kong, and R. Hao, "Remote Data Possession Checking with Privacy-Preserving Authenticators for Cloud Storage," *Futur. Gener. Comput. Syst.*, 2017.
[17] Y. Yu, Y. Zhang, J. Ni, M. H. Au, L. Chen, and H. Liu, "Remote data possession checking with enhanced security for cloud," *Futur. Gener. Comput. Syst.*, vol. 52, pp. 77–85, 2015.
[18] H. Wang, "Proxy provable data possession in public clouds," *IEEE Trans. Serv. Comput.*, vol. 6, no. 4, pp. 551–559, 2013.
[19] T. Wu and Y. Tseng, "Non-Repudiable Provable Data Possession Scheme With Designated Verifier in Cloud Storage Systems," pp. 19333–19341, 2017.
[20] D. Cash, A. Küpçü, and D. Wichs, "Dynamic Proofs of Retrievability Via Oblivious RAM," *J. Cryptol.*, vol. 30, no. 1, pp. 22–57, 2017.
[21] E. Es, "UPCommons," 2015.
[22] A. Juels and B. S. Kaliski Jr, "PORs: Proofs of retrievability for large files," *Proc. Comput. Commun. Secur. 14th ACM Conf.* , pp. 584–597, 2007.
[23] C. Tan, M. Hana, Y. Lim, and A. Gani, "Journal of Network and Computer Applications A survey on Proof of Retrievability for cloud data integrity and availability : Cloud storage state-of-the-art , issues , solutions and future trends," vol. 110, no. August 2017, pp. 75–86, 2018.
[24] H. Shacham and B. Waters, "Compact proofs of retrievability," *J. Cryptol.*, vol. 26, no. 3, pp. 442–483, 2013.
[25] J. Yuan and S. Yu, "Proofs of retrievability with public verifiability and constant communication cost in cloud," *Proc. 2013 Int. Work. Secur. cloud Comput. - Cloud Comput. '13*, p. 19, 2013.
[26] A. Kate, G. M. Zaverucha, and I. Goldberg, "Constant-size commitments to polynomials and their applications," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6477 LNCS, no. Dl, pp. 177–194, 2010.
[27] Y. Dodis, S. Vadhan, and D. Wichs, "Theory of Cryptography," vol. 9562, pp. 109–127, 2016.
[28] O. Goldreich, "A sample of samplers: A computational perspective on sampling," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6650 LNCS, pp. 302–332, 2011.

ISSN: 2233-7857 IJFGCN

[29] A. Mufson, "Th e ra c e t o ru l e th e s u n," vol. d, no. September, pp. 2–5, 2012.
[30] W. Becker and T. Botzkowski, "Auswirkungen der Digitalisierung auf das Geschäftsmodell mittelständischer Unternehmen: Eine unternehmensgrößen-, branchen- und geschäftsmodelltypabhängige Analyse," *Geschäftsmodelle der Digit. Welt*, 2019.
[31] S. Halevi, D. Harnik, B. Pinkas, and E. Haim, "Proof of Ownership in Remote Storage Systems PoW Solution : A General Protocol Security-Efficiency Tradeoff," 2013.
[32] D. Harnik, B. Pinkas, and A. Shulman-peleg, "Side Channels in Cloud Services : Deduplication in Cloud Storage Side channels in cloud services , the case of deduplication in cloud storage," no. May, 2014.
[33] J. Li, J. Li, D. Xie, and Z. Cai, "Secure Auditing and Deduplicating Data in Cloud," vol. 9340, no. c, pp. 1–11, 2015.
[34] Q. Zheng and S. Xu, "Secure and efficient proof of storage with deduplication," *Proc. Second ACM Conf. Data Appl. Secur. Priv. - CODASKY '12*, p. 1, 2012.
[35] Y. Shin, D. Koo, J. Hur, and J. Yun, "Secure proof of storage with deduplication for cloud storage systems," *Multimed. Tools Appl.*, vol. 76, no. 19, pp. 19363–19378, 2017.
[36] J. Yuan and S. Yu, "Secure and constant cost public cloud storage auditing with deduplication," *2013 IEEE Conf. Commun. Netw. Secur. CNS 2013*, pp. 145–153, 2013.
[37] X. Li, J. Li, and F. Huang, "A secure cloud storage system supporting privacy-preserving fuzzy deduplication," 2015.
[38] X. Zhang, H. T. Du, J. Q. Chen, Y. Lin, and L. J. Zeng, "Ensure data security in cloud storage," *Proc. - 2011 Int. Conf. Netw. Comput. Inf. Secur. NCIS 2011*, vol. 1, pp. 284–287, 2011.
[39] Organisation II, "Autor : Unger, A.," *Management*, vol. 2, no. 1, pp. 174–179, 2010.
[40] S. F. Roca and R. Cited, "(12) United States Patent," vol. 2, no. 12, 2006.
[41] E. Shmueli, R. Vaisenberg, Y. Elovici, C. Glezer, and C. Glezer., "Database encryption: An overview of contemporary challenges and design considerations," *ACM SIGMOD Rec.*, vol. 38, no. 3, pp. 29–34, 2012.
[42] F. Sabahi, "Virtualization-Level Security in Cloud Computing," pp. 250–254, 2011.
[43] R. A. Popa *et al.*, "Building Web Applications on Top of Encrypted Data Using Mylar This paper is included in the Proceedings of the Building web applications on top of encrypted data using Mylar," *Usenix Nsdi*, 2014.

ISSN: 2233-7857 IJFGCN