Asian Journal of Information Technology 15 (23): 4916-4925, 2016
ISSN: 1682-3915

Enhancing Data Security Through Block Level Data Integrity Assurance using Diagonal Based Rotational Technique on Cloud Storage

[1]P. Premkumar and [2]D. Shanthi
[1]Department of CSE, K. L. N. College of Engineering, 630611 Pottapalayam,
Tamil Nadu, India
[2]Department of CSE, PSNA College of Engineering and Technology, 624622 Dindigul,
Tamil Nadu, India

Abstract: Cloud computing has been envisioned as the next generation computing which enable the users to shareon-demand resources over the internet for cost effective IT services.While outsourcing the data using cloud storage several security issues arises such as confidentiality, integrity and authentication. Each of them plays an important role in the successful achievement of the other. Specifically no security strategy is achieved without assuring data integrity. In cloud computing data integrity assurance is one of the major challenges because the user has no control over the security mechanism that is used to protect the data. Data security refers to the protection of data against unauthorized access, modification or corruption and also it is necessary to ensure data integrity. Data integrityinsures that data receivedis same as data stored.It is a result of data security but the term data integrity refers to the validity and accuracy of data rather than protect the data.It is an important aspect of storage security and reliability of data. In this study, a new technique using the diagonal based rotational technique is used to enhance the performance of both data integrity violation checking in terms of storage bit reduction and time consuming and security of cloud storage. In particular, avoiding the burden of Third Party Auditor (TPA) is considered. The new model provides a mechanism to enhance data security through improving the performance of detection of data integrity violations as well as hiding the information from unauthorized users. In this approach, the data is partitioned into number of blocks and then converted into a square matrix. Determinant factor of each matrix is generated dynamically to ensure data integrity. This model also implements a combination of SHA-256 algorithm for message digest generation and Two fish algorithm for encryption are known as digital signature. Data coloring on digital signature is applied to enhance data security. This combination provides an efficient mechanism as three ways checking to enhance data integrity and data security between the client and the cloud provider. Also block level implementation of the proposed method makes computation faster. The proposed method utilizes attributes such as encryption time and decryption time, memory utilization and execution time to improve the efficiency for verification of untrusted and outsourced storage. The performance analysis using cloud simulator shows that the proposed scheme is highly efficient and secure as it overcomes the limitations of previous approaches of data security using encryption and decryption algorithms and data integrity assurance methods using TPA. Thus, the proposed scheme reduces the server computation time, memory utilization and also achieves maximum level of accuracy while compared with the previous.

Key words: Cloud computing, data integrity, data security, Sha-256, digital signature, two fish, encryption/ decryption

INTRODUCTION

Cloud computing is a modern computing paradigm in which scalable resources are shared dynamically a service over the internet (Akshay, 2014). Cloud storage services enable the user to enjoy with high storage with less overhead but it has many potential threats like data integrity, availability, data privacy and so on. The two issuesare mainly occur while outsourcing the data using cloud storage is data integrity and data securitydue to unfaithful cloud service provider (Diffie and Hellman, 1976). Data integrity is the form of protection of data

Corresponding Author: P. Premkumar, Department of CSE, K. L. N. College of Engineering, 630611 Pottapalayam, Tamil Nadu, India

against loss and damage caused by hardware, software and network failure (Kahate, 2013; Kumar, 2014). Normally data inaccuracy can occur either accidently through programming errors or maliciously through breaches or hacks. It is one of the important aspect among the other cloud storage issues because data integrity ensured that data is of quality, correctness, consistency, accuracy, security, confidentiality, reliability and accessibility but assurance of data integrity in the cloud is a major challenge that is faced by today's cloud users (Wang *et al*., 2010). It refers to assurance by the user that the data is not modified or corrupted by the service provider or other users. The performance of data integrity is measured by using the parameters like execution time,encryption and decryption time, CPU utilization, memory utilization.While outsourcing their data using cloud storage donot maintain a local copy hence cryptographic measures cannot be used directly to monitor the integrity of data and also downloading the data for monitoring integrity is not a viable solution. Therefore, an external Third Party Auditor (TPA) is required (Govinda *et al*., 2012). The TPA is an independent authority that has capabilities to monitor the integrity of outsourced data by the client and also inform on data corruption or loss, if any. But it requires separate memory and also takes more time for verification of data to ensure integrity of data,hence the overall performance is degraded. Now a days, software professionals employ number of practices to ensure data integrity which includes data encryption, data backup, access controls, input validation,data checking,error detection and correction while transmitting and storing the data. The performance of data violation checking methods are affected due to communication overhead, memory overhead, key size and execution time.The scope of the data integrity assurance mechanism can be classified into two levels: first is to prevent data corruption and second is to detect and correct data violation. In this study only focus on detection of data violation.The algorithms and methods to ensure data integrity are discussed in (Premkumar and Shanthi, 2014).

Literature review: This study discusses the performance of the various methods and algorithms used to ensure data integrity and data security in cloud environment. In study (Ghaeb *et al*., 2011), certain degree of integrity assurance is provided by RAID technique but it operates only on binary data,takes more computation time and alsothe value of determinant factor isthree bits long hence need large memory for storage. In study (Wang *et al*., 2010), to evaluate the performance of the encryption algorithm for text files, it uses AES, DES (Coppersmith, 1994) and RSA algorithm and the parameters such as computation time, memory usage and output bytes are considered (Nie and Zhang, 2009). The time taken to convert the plain text into cipher text is known as encryption time. The decryption time is the time that a decryption algorithm takes to reproduce a plaintext from a cipher text. Comparing these three algorithms, RSA takes more time for computation. The memory usage of each algorithm is byte level. RSA requires more memory than AES and DES. In study (Stallings, 2006), various algorithms such as AES, 3DES, Blowfish and DES are discussed. Throughput is equal to total encrypted plaintext in bytes divided by the encryption time. Higher the throughput, higher will be the performance. Asymmetric encryption techniques are slower than symmetric techniques, because they require morecomputational processingpower. Also, Blowfish algorithm gives better performance than all other algorithms in terms of throughput (Schneier, 1993). In study and study Blowfish algorithm, the performance evaluation of AES and Blowfish algorithms are discussed (Schneier, 1994, 1993). The parameters such as time consumption of packet size for 64 bit encodings and hexadecimal encodings, performance for encryption of text filesand the throughput are considered. The result shows that Blowfish has better performance than AES in almost all the test cases.

MATERIALS AND METHODS

The proposed technique is based on the Determinant Factor (DF) approach to ensure both data integrity and security which involves the following steps:

Before transmitting the series of data, it is divided into N-matrices where N is given by:

$$N = \frac{\text{Total number of data}}{d \times d}$$

Where (d × d) is the number of elements per matrix.The determinant factor of each matrix is computed and appended with the data. At retrieving stage, it is compared with the determinant factor of the sender's data for data integrity assurance. But, it is observed thatthere is one defect with this method. The DF is zero, if any one of the rows is proportional to another row; the same is true for columns. Also, the DF does not change, if some of the rows or some of the columns are interchanged. In addition, the DF is zero, if any single row or column has zero values only. In order to alleviate this problem, a new technique is performed as given as: For each element of the matrix is reconstructed using the property of diagonal matrix rotation to formalize the original data

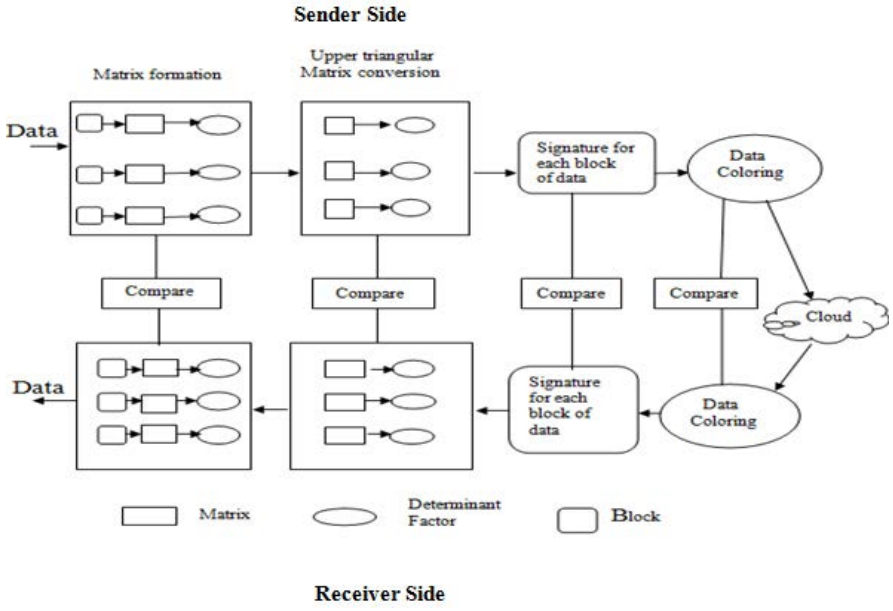

Fig. 1: Architecture of the proposed system

matrix into a new matrix. The determinants of both original and diagonalmatrix rotation is computed and appended with each matrix. For example, DF value for the following matrix is zero. After applyingthis new technique, DF value of the resultant matrix is not zero.

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} = \begin{pmatrix} 4 & 1 & 2 \\ 7 & 5 & 3 \\ 8 & 9 & 6 \end{pmatrix}$$

Then digital signature for each determinant factor is generated using SHA-256 and Twofish algorithms. Finally, data coloring is applied on each digital signature before transmission or storing the data to enhance data security. At the receiver side, both determinants are recomputed againand also decrypting the signature thencompared the sender's values. If there is a match, it ensures that there is no modification in the data during the transmission otherwise particular block of data is violated. The results of the proposed system shows that block based determinant approach outperforms than other data integrity checking methods and also provides data privacy for securing the data from unauthorized users. Figure 1 shows the architecture of the proposed system. The steps which involved in block based determinant approach is given as.

Sender's end:

- Data is taken as a string format. Each string is converted as bytes and the number of bytes that constitute a block is decided. Next bytes will be added and divided into number of blocks
- Convert each block of data into square matrix
- Find Determinant Factor (DF) for each matrix
- Construct a new matrix using Diagonal based rotational techniqueto ensure DF is not zero
- Find DF for the matrix constructed in s.tep 4
- Generate message digest using SHA-256 for each DF calculated in Step5
- Generate Digital Signature ie., Encrypt this Message Digest using Twofish algorithm
- Apply data coloring for each of the digital signature generated using step7
- Store the colored data into cloud storage

Receiver's End:

- Regenerate the colors from the colored data
- Decrypt the Message digest
- Reconstruct the new matrix
- Calculate DF for the above matrix constructed in step3
- Construct the new matrix and calculate DF
- Compare the results obtained in steps 1,2,4,5 respectively of Receiver's End with 8, 6, 5 and 3 of Sender's End
- If the results are same in all the steps mentioned in step6, then this ensures data integrity otherwiseintegrity of data is not attained i.e., a particular block of data has been violated Steps 6, 7 of sender side and also Step 2 of receiver side is explained as below:

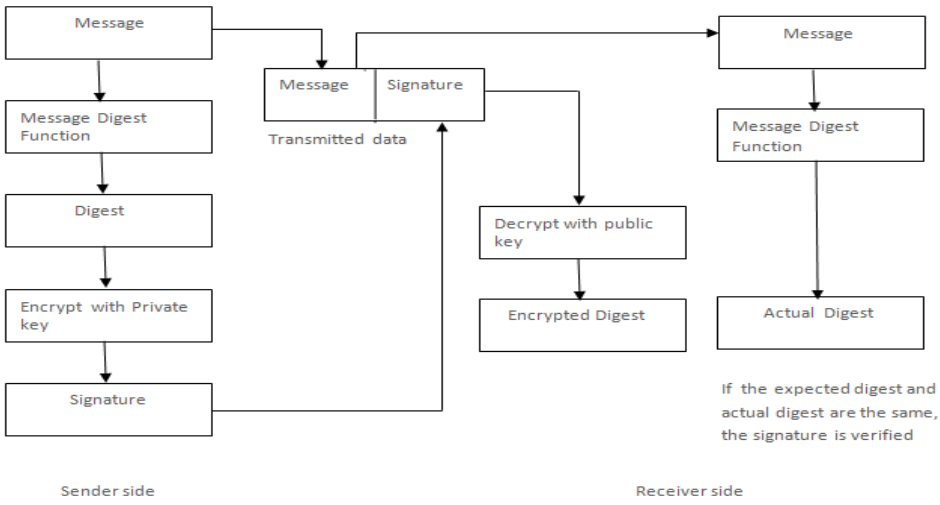

Fig. 2: Block diagram for generation of digital signature

Signed and Encryption:

Sender sends a message as DF
Calculate Digest

Digest=$[Message]_{hash}$

Sign the Digest

Message+$[Digest]k_{pri}+k_{pub}$

Encrpt with Symmetric key

[Message +$[Digest]k_{pri}+k_{pub}+k_{sym}$

Send signed and encrypted message to Recipient
Here steps 1, 2 and 3 are for Signature generation and Step4 for encryption (Two fish algorithm)

Decrypt and verifying message:

- [Message +$[Digest]k_{pri}+k_{pub}]+ [k_{sym}]$
- Decrypt Ksym with receivers private key [Message +$[Digest]k_{pri}+k_{pub}]+k_{sym}$
- Decrypt Digest using Public key and also evaluate the DigestDigest = $[Message]_{Hash}$
- Compare these two Digests (Actual Vs Expected digest)

If two digests are equal, message is decrypted and signature is verified. Here Steps 1-3 are for Decryption and step 4 for Verification. It can be described in Fig. 2.The steps involved to generate encrypted digital signature are as follows:

Step1: The document will be crunched into fixed few lines by using SHA-256 algorithm to generate Message digest.

Step 2: At Sender side encrypt the message digest using its public key to generate Digital signature.
Step 3: At Receiver side decrypt the message using their own private key.

Step 4: Regenerate the message digest

Step 5: Finally the Signature is verified using Sender's public key.

SHA-256 algorithm is used to implement integrity of the message which produce message digest of size 128 bits. These are mathematical functions that process data to produce different message digest for each unique message. It processes the message and generates 128 bits message digest. The SHA-256 algorithm consists of the following steps:

Step 1: Add Padding to the end of the genuine message length is 64 bits and multiple of 512

Step 2: Appending length. In this step the excluding length is calculated

Step 3: Divide the input into 512 bit blocks. In this step the input is divided into 512 bit blocks

Step 4: Initialize chaining variables. In this step chaining variables are initialized. In the proposed method 5 chaining variables are initializedeach of size 32 bits giving a total of 160 bits

Step 5: Process Blocks, i.e., Copy the chaining variables, Divide the 512 into 16 sub blocks,Process 4 rounds of 20 steps each.

Step 6: Output Generation: The algorithm is divided into 5 steps: Key Generation, Digital Signing, Encryption,

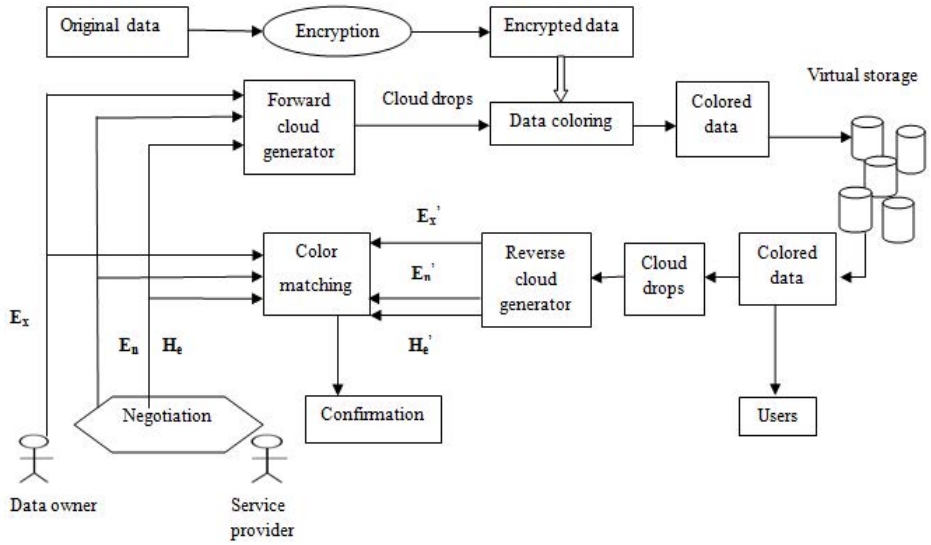

Fig. 3: Block diagram for color coding generation

Decryption and signature verification are discussed as below:

Step 1; key generation: Different combinations of key size such as 128, 192 or 256 bits are used. To perform the Two fish algorithm, round keys must be generated from the userprovided key. The key schedule of this algorithm provides 33 128-bit keys to be mixed with the text blocks during the round function of the algorithm. First create eight 32-bit prekeysusing the key provided by the user. The user's key is split every 32 bits to do this and then generate 132 intermediate keys using the following reoccurrence: for i from 0 to 131.The 33 round keys are generatedfrom these intermediate keys by running through the S-Boxes and combining them into 128-bit blocks.

Step 2: Digital signing: Generate message digest of the document to be send by using SHA-256 algorithm. The digest is represented as an integer m. Digital signature S is generated using the private key(n, d).

S= md mod n.

Sender sends this signature S to the recipient.

Step 3: Encryption: Sender represents the plain text message as a positive integer m. It converts the message into encrypted form using the receiver's public key (e, n). C= me mod n. Sender sends this encrypted message to the recipient. Here, n is the modulus and e is the encryption exponent.

Step 4: Decryption: Recipient does the following operation: Using his private key (n, d), it converts the cipher text to plain text 'm'.

m= Cd mod n; Where d is the secret exponent or decryption exponent.

Step 5: Signature verification: Receiver does the followings to verify the signature:

An integer V is generated using the sender's public key (n, e) and signature S

V= Se mod n

It extracts the message digest M1, from the integer V using the same SHA-256 algorithm. It computes the message digest M2 from the signature S. If both the message digests are identical, i.e., M1= M2, then signature is valid. The block diagram for generating color coding is shown in Fig. 3. Each user is specified by a color that helps to protect and also avoids the manipulation of original data. Figure 3 shows the details involved in the data coloring process which aims to associate a colored data with its own, whose user identification is also colored with the same Expected value (Ex), Entropy (En) and Hyperentropy (He) identification characteristics. The cloud drops are added into the input and remove color to restore the original. The process uses three data characteristics to generate the color: the Expected value (Ex) depends on the data content known only to the data owner. Where as Etropy (En) and Hyperentropy (He) add randomness or uncertainty which are independent of the data content and these three functions generate a collection of cloud drops to form a unique color that the providers or other cloud users cannot detect. This technique can also be applied to protect documentsin the

cloud.The overall property of colored drops can be represented by three numerical characters. On one hand, construct forward cloud generator to produce a lot of drops as illustrated in Algorithm 1. On the other hand, construct reverse cloud generator to cope with the colored drops and revert Ex, En and He, as illustrated in Algorithm 2.

Algorithm 1: Forward cloud generator

Step 1: Generate a normally distributed random number En_i = NORM (EN, He^2)

Step 2: Generate a normally distributed random number x_i = NORM (Ex, En_i)

Step 3:

$$\mu_i = \exp\left[-\frac{(xi - Ex)^2}{2(En_i')^2}\right]$$

Step 4: x_i with certainty degree of μ_i is a cloud drop in the domain.

Step 5: Repeat Steps 1-4 and generate drops.

Algorithm 2: Reverse cloud generator:

Step 1: Calculate mean

$$\bar{X} = \frac{1}{2}\sum_{i=1}^{n} x_i$$

and;

variance

$$S^2 = \frac{1}{2}\sum_{i=1}^{n}\left(x_i \bar{x}\right)$$

Step 2: $En' = \bar{X}$

Step 3:

$$En' = \sqrt{\frac{\pi}{2}} \times \frac{1}{n}\sum_{i=1}^{n}|x_i - Ex|$$

Step 4:

$$He' = \sqrt{S^2 - En'^2}$$

Ex is provided by data owner; En and He are produced by negotiation of data owner and service provider. Each cloud user is provided with a value called expected value which is known only to the user.The negotiated values with the CSPs are Entropy which is unique for all users in the particular group sharing the data in the cloud.Hyperentropy is the value which is common to all the group users of the data.Then, a lot of cloud drops will be formed by forward cloud generatorand are used to color the user data. When the data are used, the cloud drops are extracted from colored data Ex0, En0 and He0 will be produced by reverse cloud generator. The color matching which indicates data is not modified by others. Data owner and storage service provider negotiate together to select En and He, just like the key. Ex, En and He are three mathematical characters. En and He can be used to transform a certain print to uncertain print drops. Figure 3 shows different color drops produced according to different En. Also compute the entropy of each cloud drop (En0) and compare the difference between En and En0. To provide the continuous authentication within the group, an automated validation of data can be made at regular intervals of time.The experiment result is illustrated in the concerned table and the curve of case is shown in different Fig. 1-9 .

RESULTS AND DISCUSSION

The performance of the proposed system is evaluated using cloud simulator based on the parameters viz., execution time, encryption time and decryption time, memory utilization, CPU time, key size. The performance results have been summarized in various tables regarding with various parameters and also a conclusion has been presented. Based on the results, it has been concluded that the Two fish is the best performing algorithm among the variousalgorithms chosen for implementation. Figure 4 describes Encryption time for various block size of data given in Table 1. It can be seen that as the block size increases the encryption time also increases gradually.

Figure 5 describes decryption time for various block size of data given in Table 2. It can be seen from the figure that the decryption time is linearly proportional to the block size. Both Fig. 6 and 7 describes Resource utilization with respect to CPU time, Memory utilization related with the data given in Table 3 and 4 respectively. Figure 8 describes time taken for digital

Table 1: Encryption Time for different text sizes

Data size (kb)	Time (ms)
30	40
45	59
60	72
80	90
95	110
120	130

Table 2:Decryption time for different text sizes

Data size (kb)	Time (ms)
30	38
45	56
60	68
80	85
95	103
120	127

Table 3:CPU time for different text sizes

Data size (kb)	CPU time (sec)
30	10
45	19
60	23
80	37
95	43
120	51

Table 4: Memory utilization for various text sizes

Time (ms)	RAM (mb)
30	12
45	15
60	20
80	24
95	29
120	32

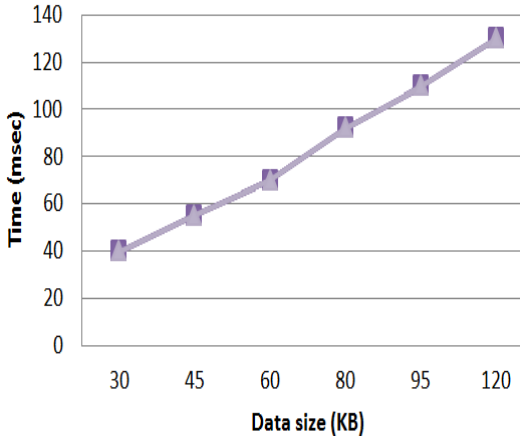

Fig. 4: Encryption time

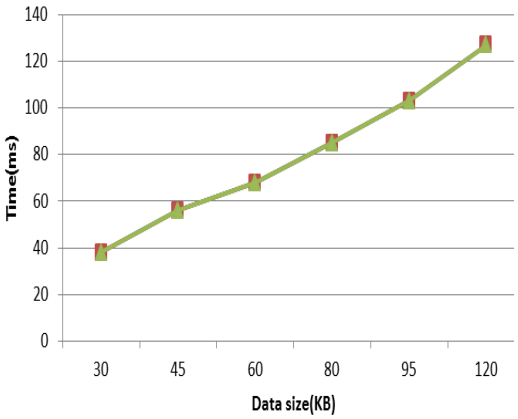

Fig. 5: Decryption time

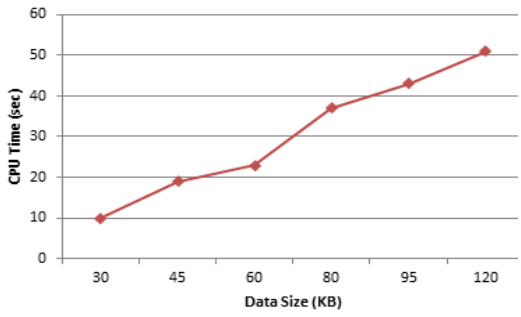

Fig. 6: CPU utilization

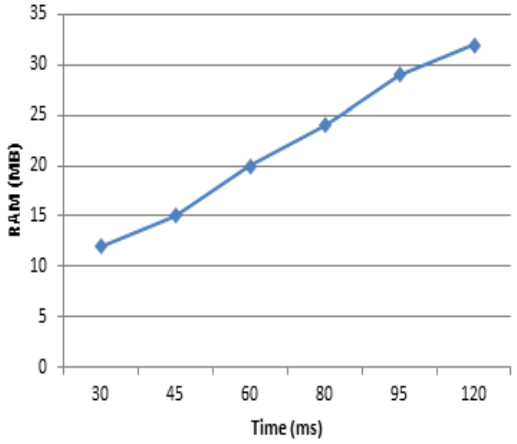

Fig. 7: RAM Utilization

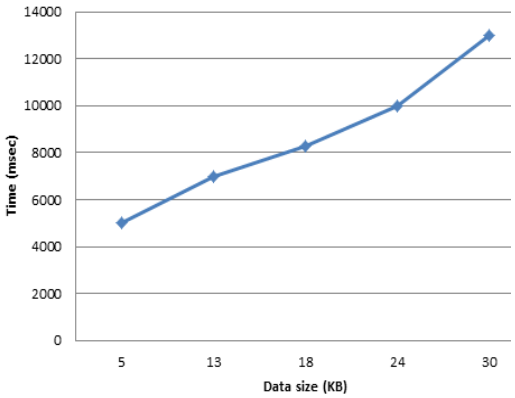

Fig. 8: Generation of Digital signature

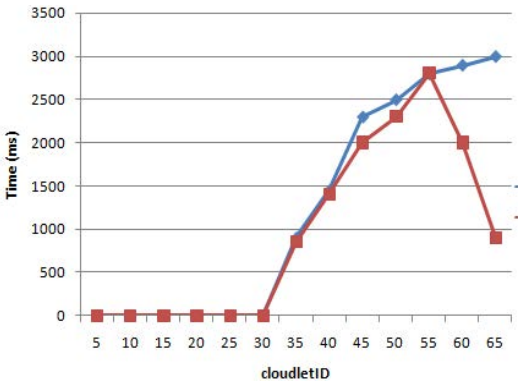

Fig. 9: Execution time

signature generation with various block size of data given in the Table 5. It can be seen from the figure that the digital signature generation time is linearly proportional to the block size. Figure 9 describes time taken for executing various block size of data given in the Table 6. Here, the average finishing time is constant proportional to the block size. Encryption and decryption throughput is shown in Fig. 10 with various size of data is given in

Table 5: Digital signature creation time for various text sizes

Data size (kb)	Time (msec)
5	5000
13	7000
18	8300
24	10000
30	13000

Table 6: Start time, finish time forvarious text sizes

CloudletID	Start time (ms)	Finish time (ms)
5	0	0
10	0	0
15	0	0
20	0	0
25	0	0
30	0	0
35	850	900
40	1400	1450
45	2000	2300
50	2300	2500
55	2800	2800
60	2000	2900
65	900	3000

Table 7: Enhanced Two fish encryption/decryption throughput

Throughput (Mbps)	Two fish	Rijndael
Min throughput	2300	1500
Max throughput	5000	4000

Table 8: Performance comparison with Two fish and RC6

Algorithm name	Encryption performance	Memory usage
Twofish	60	50
RC6	50	50

Table 9: Encryption time using diagonal based rotation compared with RC6

Block size (kb)	RC6	Two fish
5	2	1
10	3.5	2.2
15	4.5	3.5
20	5	4.2
25	6.1	5.3
30	6.8	6
35	7.3	6.7
40	8.2	7.3
45	9.2	8.4
50	10.2	9

Table 7. There are two main characteristics are considered of a good encryption algorithmie., security and speed. In this study also be compared the security performance analysis of two algorithms two fish and AES. First, we will discuss security issues of both algorithms by considering their safety factor. Then we study encryption speed of both algorithms by encrypting text data and analyze their performance in terms of throughput of Rijndael algorithm on different size of memory.The results are the relationship betweenperformances of algorithms as shown in Fig. 11 with the data is given in the Table 8. Figure 12 and 13 gives the comparison between two fish with RC6 algorithm with respect to the parameters

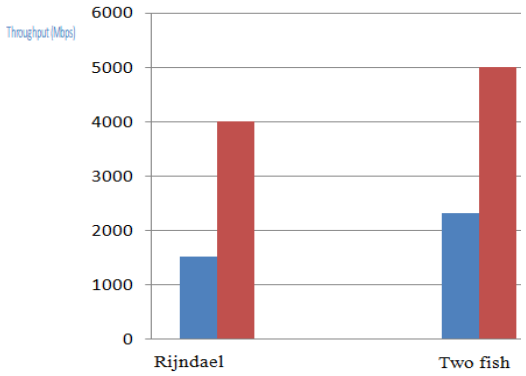

Fig. 10: Enhanced two fish encryption/decryption throughput

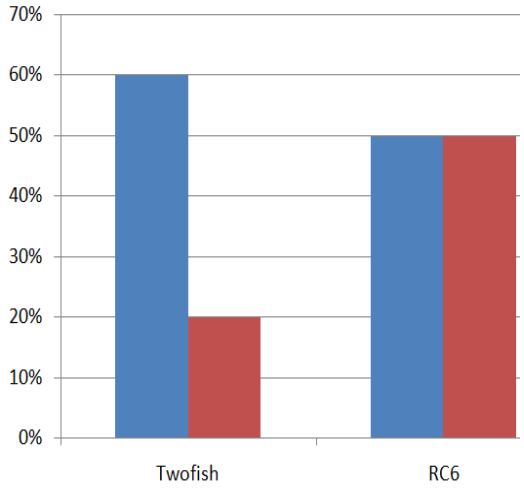

Fig. 11: Performance comparison with two fish and rc6

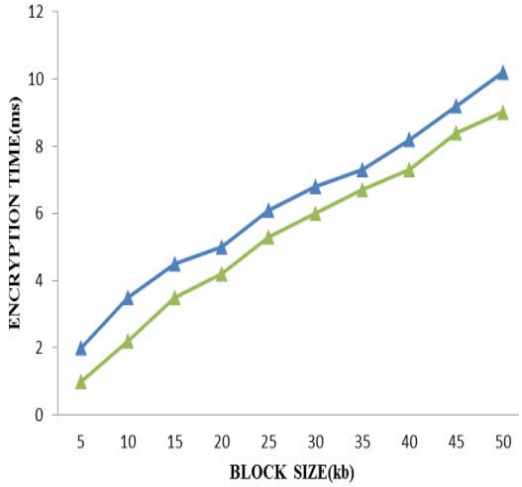

Fig. 12: Encryption time

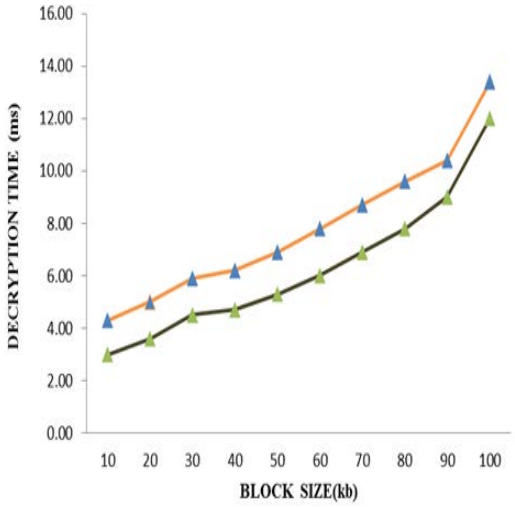

Fig. 13: Decryption time

Table 10: Decryption time using diagonal based rotation compared with RC6

Block size (kb)	RC6	Two fish
10	4.3	3.0
20	5.0	3.6
30	5.9	4.5
40	6.2	4.7
50	6.9	5.3
60	7.8	6.0
70	8.7	6.9
80	9.6	7.8
90	10.4	9.0
100	13.4	12.0

ie.,encryption and decryption time for each block of datais given in the Table 9 and 10. Based on the results two fish algorithm provides better performance with respect to encryption and decryption time.

CONCLUSION

This study presents a new technique for enhancing data security through improving data integrity violation checking over the cloud storage without using TPA. In the proposed technique, the data are divided into blocks; where each block is arranged into square matrix. An element in this matrix is arranged into a new form using diagonal based rotational techniqueleads to memory saving through reduction of bits. Also digital signature is applied on each determinant factor to enhance data integrity assurance. This model also uses data coloring on encrypted digital signature to achieve the data security enhancement which helps the user to verify and examine the data from unauthorized people who manipulate the data in the cloud storage. In this method accuracy is maintained at satisfied level by rearranging the data two times via original matrix and its corresponding Diagonalbased rotation.Though it requires more computation time, it provides good level of accuracy and security of data. Thus, here it tries to provide a new insight to improve the cloud storage security through detection of data integrity violations during storing or transmission. The simulation results show that the new method gives better results compared to the RC6 algorithm and has resolved all of their deficiencies that go along with data integrity assurance methods towards security. The performance measures viz., better encryption/decryption time and also CPU Time,memory utilization and quicker detection of violation is considered. In future, more experiments will be conductedusing various algorithms and methods in cloud computing and test the performance of the proposed approach. Also, the proposed model can be implemented with other types of data like image,sound and multimedia data.

REFERENCES

Akshay, C., 2014. Cloud computing. Asian J. Manage. Sci., 2014: 01-06.

Coppersmith, D., 1994. The data encryption standard (DES) and its strength against attacks. IBM. J. Res. Dev., 38: 243-250.

Diffie, W. and M.E. Hellman, 1976. New directions in cryptography. IEEE Trans. Inform. Theory, 22: 644-654.

Ghaeb, J.A., M.A. Smadi and J. Chebil, 2011. A high performance data integrity assurance based on the determinant technique. Future Gener. Comput. Syst., 27: 614-619.

Govinda, K., V. Gurunathaprasad and H. Sathishkumar, 2012. Third party auditing for secure data storage in cloud through digital signature using RSA. Int. J. Adv. Sci. Tech. Res., 4: 525-530.

Kahate, A., 2013. Cryptography and Network Security. 3rd Edn., Tata McGraw-Hill Education, New Dehli, India, ISBN: 978--25-902988-2, Pages: 492.

Kumar, M.A., 2014. Integrity check mechanism in cloud using SHA-512 algorithm. Int. J. Eng. Comput. Sci., 3: 6033-6037.

Nie, T. and T. Zhang, 2009. A study of DES and Blowfish encryption algorithm. Proceedings of the IEEE Region 10 Conference TENCON, January 23-26, 2009, Singapore, pp: 1 4-.

Premkumar, P. and D. Shanthi, 2014. An efficient dynamic data violation checking technique for data integrity assurance in cloud computing. Int. J. Innovative Res. Sci.Eng. Technol., 3: 2649-2654.

Schneier, B., 1993. Description of a New Variable-Length Key, 64-Bit Block Cipher (Blowfish). In: Fast Software Encryption. Anderson, R. (Ed.). Springer Berlin Heidelberg, Berlin, Germany, ISBN: 978-3-540-58108-6, pp: 191-204.

Schneier, B., 1994. The blowfish encryption algorithm. Dr. Dobbs J. Software Tools Prof. Programmer, 19: 38-43.

Stallings, W., 2006. Cryptography and Network Security: Principles and Practices. 3rd Edn., Pearson Education India, New Dehli, India, ISBN: 978-1-25-902988-2, Pages: 492.

Wang, C., Q. Wang, K. Ren and W. Lou, 2010. Privacy-preserving public auditing for data storage security in cloud computing. Proceedings of the IEEE INFOCOM, March 14-19, 2010, San Diego, CA., USA., pp: 1-9.