International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-4, February 2020

Secure Distributed Data using Multi-Cloud

Worud Mahdi Saleh, Ziyad Tariq Mustafa Al-Ta'i, Ghassan Sabeeh Mahmood

Abstract*: *Abstract—Implementing cloud computing provides many paths for web-based service. But, data security and privacy requirement become an important problem that limits several cloud applications. One of the key security and privacy concerns is the fact that cloud service suppliers have access to data. This concern greatly reduces the usability of cloud computing in many areas, such as financial business and government agencies. This paper focuses on this important issue and suggests a new approach, so cloud providers cannot directly access data. The proposed approach is divided into two sides: upload side and download side. In upload side, there is three stages, at the first stage; the transmitted file is splitted and then encrypted in order to achieve the data security requirement. At the second stage, the splitted data are integrity checked by MD5 algorithm, in order to achieve integrity requirement. At the third stage, the checked splitted data are stored separately in three -clouds, in order to achieve distribution requirement. In download side, also there is three stages. At the first stage, the data is retrieved from the three-clouds. At the second stage, data integrity is performed using MD5. At the third stage, data decryption and merging are done. The proposed approach is successfully implemented on (25 KB) image. The proposed model is successfully implemented in uploading side dependent on shares3 because provide high security with total time of (8.144 sec), and in downloading side with total side of (9.42).

Keywords: Multi-cloud, data security, distribution, integrity.

I. INTRODUCTION

A model that has the capability for convenient, on-claim network access to a shared pool of resources for configurable computing is cloud computing for instancenets, servers, storage, services, and applications that can be quickly by using less effort of management or interaction of service provider be provisioned and released. The central multi-clouds criteria is the protection of data security, the security of data and reliance difficult have continually been the core and puzzling issues in cloud computing. In the case that multi-user use the same resources lead to risk that the data been damaged. In order to avoid this problem, security must put on the counting storage of data and also access or process. Three main requests of the security of the cloud are confidentiality, integrity, and availability.

Revised Manuscript Received on February 28, 2020.
***** Correspondence Author
Worud Mahdi Saleh*, Student, Department of Computer Science, University of Diayla.
Ziyad Tariq Mustafa Al-Ta'i, Scholar, Department of Computer Science, University of Diayla.
Ghassan Sabeeh Mahmood, Scholar, Department of Computer Science, University of Diayla.

The main weaknesses are confidentiality and great care should be taken to make sure that data is not subjected to any attacks[1].

II. RELATED WORK

Gutte, Vitthal S., Amol N. Jadhav, and Pramod Mundhe [5], they used certificate less encryption-based schemes by relying on a public cloud server. they applied it as underlying techniques to design the data users and data access control scheme in the process. they came to know that for them certificated less encryption methods can efficiently provide not only forward security but also backward security in a more efficient way. El-Booz, et.al.[6], they provided the organization protection from the cloud provider by making systemof cloud storage secure, the auditor of third party, and precise users who have the ability to use their old account to get access the data that kept in the cloud . Their proposed system enhanced the authentication, to ensure the security two techinques of authentication used; time-based one-time password (TOTP) for verification of cloud users and reflex blocker protocol (ABP) to completely guard and unsure that unauthorized third-party auditor would not attack the system. Bala, Yogesh, and Amita Malik[7],they used (BIHEA) algorithm which had the aim of securing data/files extent overthe location of hybrid cloud. The intended algorithm had the job of user data encryption at run-time by given that the user whos is authorized biometric-feature-based one-time password. Every time a user is authenticated by a totally different one time password.

. Dijk, and A. Juels [8], they proved that privacy execution hyperd model is needed for security in cloud storage. In order that the users will have improved and rational probabilities to advantage capable of providing services of security for their cloud storage at inexpensive costs, a model was suggested that distributes the units of data between more than one service provider (SP) in a way that SPs can recover any significant data from the data portions kept on its servers, without achievement some more data pieces from other providers of service. Oliveira, et. al. [9], they calculated distributed data over several clouds or networks in a way that if one attacker as an opponenthas the capability to attack in one network, he will not have the ability to recover any significant data, as the pieces of balancing are kept in the different network. This is the reason that the proposed model; suggestusing a distributed method, in that all the data pieces are needed out of the whole range of distribution, for operational recovery. Athens, and Giuseppe[10], they suggested an application named atomic proxy re-encryption, in that a semi-trusted proxy changes a Alice cipher text into a Bob cipher textsubtracting seeing the original plaintext.

The calculation that was re-encryption would be quick and safe for encrypted files. Shivanna, K., S. Prabhu Deva, and M. Santoshkumar[11], they approached for double encryption that rises privacy for keeping and resources accessing on platforms of the cloud. The probable technique offers structures of authentication and privacy to the owner of the data, cloud service providers and cloud users.

Yesilyurt, Murat, and Yildiray Yemen[12], they generated the architecture of the cloud and models of organization are saw, and the key properties in the offering of requirements security of all those models as well as topics to be busy into matter are termed in detail. In addition, the approaches and tools seeing how information security, integrity and confidentiality that approaches the basis of recent knowledge are applied in cloud computin.

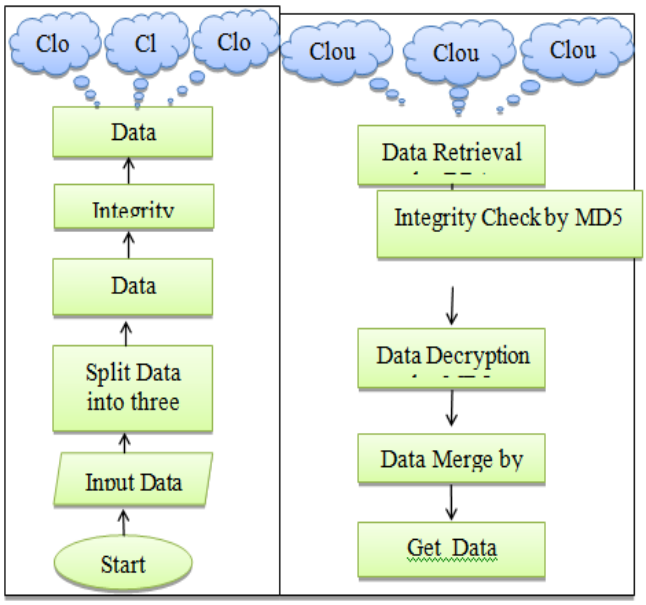
Pereira Sancha, et al.[13], they presented Storekeeper, a privacy preserving cloud aggregation service that enables file sharing on multi-user multi-cloud storage platforms while preserving data confidentiality from cloud providers and from the cloud aggregator service. To provided those property, Storekeeper decentralizes most of the cloud aggregation logic to the client side enabling security sensitive functions to be performed only on the trusted client endpoints.

Alsolami, Fahad, and Terrance E. Boult[14], they proposed the CloudStash scheme, a system that applied the secret-sharing scheme directly on the file to store multi-shares of a file into multi-clouds. CloudStash utilizes secret-sharing, low cost cloud storages and multi-threading to improve confidentiality, availability, performance and fault tolerance. CloudStash achieves this improvement by splitting a file into multi-shares of secret and distributing these multishares into multi-clouds simultaneously where threshold shares are required to reconstruct the file..

The rest of the paper is structured as follows: Section 2 includes the related works. Section 3 contains a suggested approach. Section 4 demonstrates the experimental analysis. Section 5 is the inferred conclusions from this work.

III. THE PROPOSED APPROACH

The proposed approach is divided into two sides: upload side and download side. In upload side, there is three stages, at the first stage; the transmitted file is splitted and then encrypted in order to achieve the data security requirement. At the second stage, the splitted data are integrity checked by MD5 algorithm, in order to achieve integrity requirement. At the third stage, the checked splitted data are stored separately in three -clouds, in order to achieve distribution requirement. In download side, also there is three stages. At the first stage, the data is retrieved from the three-clouds. At the second stage, data integrity is performed using MD5. At the third stage, data decryption and merging are done. The proposed model is shown in figure (1) (a) and (b).



(a) UploadSide (b) Download Side

Figure (1) The Proposed Model (a) Upload Side (b) Download Side

A- Upload Side

Upload stage consists of: data security, data integrity, and data distribution stages.

1- Data Security Stage

Data security stage in the proposed model consists of data splitting, renaming, and encrypting.

A- Data Split and Rename

The splitting of data is a type for data security over a network of computers. The data is splitted into three data shares by using data split and rename (DSRA) technique. This technique is described in algorithm (1) and algorithm

Algorithm(1): Spilt Shares Image Algorithm

Input: User Image (Colored Image)
Output: Spilt Shares image *Ni (N1, N2, N3)*
Step1: Read user image as VAR IM and number of block as VAR *BlockNumber*
Step2: Calculate horizontal block size from *numberOFColume* in image divided by *BlockNumber* as VAR *blockSize*
Step 3: SET start Cut_Index value to 1
Step 4: FOR index =1 to *BlockNumber*
Step5: SET startIndex value to 1 for each image block
Step 6: FOR i =1 to *number OF Row in N*
Step 7: FOR j = Cut_Index to *blockSize in N*
Step 8: Put TempImageBlock (i, startIndex, index) value to IM (I,j) value.
Step 9: Increase *startIndex* value by 1
Step 10: ENDFOR
Step 11: ENDFOR
Step 12: Upadate *Cut_Index* by Set it to *blockSize value*
Step 13: Upadate *blockSize* by Set it to *blockSize* value multiply by (index+1)
Step 14: *N* (index) = *TempImageBlock*
Step 15: Upload *N* (index) image to cloud by use Data distibuted Algorithm
Step 16: ENDFOR
Step 17: Set Procedures Output to *Ni*

Algorithm (2): Splitted Data Rename Algorithm

Input: *User colored Image name, Number of Spilt, User Cloud ID*
Output: Spilt Shares image name as Ni *(N1, N2, N_n)*
Step 1: Generate random INTEGER number as image ID merge with image name.
Step 2: FOR i=1 TO *Number of Spilt DO*
Step 3: Generate Encryption Image Spilt name by Merge *User Cloud ID with* image ID and image sequence i.
Step 4: Save Generated Encryption Image Spilt name in S_i
Step 5: INCREASE *i* value by 1.
Step 6: ENDFOR

B- Encryption Algorithm (RC6)

Data encryption is done by using RC6 algorithm. RC6 is an entire encryption algorithm parameterized individual. A kind of RC6 is more precisely definite as RC6-w/r/b where the word size is w bits, encryption includes of a nonnegative number of rounds r, and b means the length of the key of encryption in bytes. For values w = 32 and r = 20. Of the actual consequence of the AES, effort will be the kinds of RC6 with 16-, 24-, and 32-byte keys. For all alternates, RC6-w/r/b operates on parts of four w-bit words using the succeeding six basic operations[15]. Encryption using RC6 is shown in algorithm (3).

Algorithm (3): Image Encryption using RC6 Algorithm

Input: *User colored Image, w-bit round keys S[0, ..., 2r + 3]*
Output: *Cipher colored Image as Ci (C2* $_{RED}$, *C2* $_{GREEDN}$, *and C2* $_{BULE}$*)*
Step1: Spilt colored Image *into* Image $_{RED}$, Image $_{GREEN}$, and Image $_{BULE}$ layers
Step 2: SET *Number Of Byte* to 16 and VAR p =1 as Layer number in colored image.
Step 3: Calculate the number of all byte in Image by multiply number of Row with number of columns and save it as VAR *allPixle*
Step 4: FOR each layer of Colored image (RED.GREEDN, BULE) DO
Step 5: Covert current image layer to ONE dimension Array as VAR *T*.
Step 6: SET VAR i =1 as Start index.
Step 7: WHILE i lass or equal (*allPixle-15)* DO
Step 8: C_p (i to (i+15)) = RC6Encrypt (T_p(i to (i+15)), S), call RC6 Encrypt algorithm
Step 9: INCREASE *i* value by *16*.
Step 10: ENDWHILE
Step 11: Covert C_p to TWO dimension Array as VAR $C2_p$
Step 12: INCREASE *p* value by *1*.
Step 13: ENDFOR
Step 14: Merge Cipher image layers (*C2* $_{RED}$, *C2* $_{GREEDN}$, *C2* $_{BULE}$) in *Cipher* colored Image

C- Data Integrity Stage

Data integrity is the protection of, and the guarantee of the accuracy, and consistency of data. Data integrity in this work had been done using MD5 algorithm which is described in algorithm (4).

Algorithm (4): MD5 [16]

Input – Suppose a b-bit message

Output – Integrity massage

Step 1 – append padded bits:

– The message is padded so that its length is congruent to 448, modulo 512.

-Means extended to just 64 bits shy of being of 512 bits long.

– A single "1" bit is appended to the message, and then "0" bits are appended so that the

length in bits equals 448 modulo 512.

Step 2 – append length:

– A 64 bit representation of b is appended to the result of the previous step.

– The resulting message has a length that is an exact multiple of 512 bits.

Step 3 – Initialize MD BLffer:

A four-word buffer (A,B,C,D) is used to compute the message digest.

- Here each of A,B,C,D, is a 32 bit register.

These registers are initialized to the following values in hexadecimal:

word A: 01 23 45 67

word B: 89 ab cd ef

word C: fe dc ba 98

word D: 76 54 32 10

Step 4 – Process message in 16-word blocks.

– Four auxiliary functions that take as input three 32-bit words and produce as output

one 32-bit word.

$F(B, C, D) = (B \wedge C) \vee (\neg B \wedge D)$

$G(B, C, D) = (B \wedge D) \vee (C \wedge \neg D)$

$H(D, C, D) = (B \oplus C \oplus D)$

$I(B, C, D) = (C \oplus (B \vee \neg D)$

Step 5 – Process message in 16-word blocks cont.

– If the bits of X, Y, and Z are independent and unbiased, the each bit of F(X,Y,Z),

G(X,Y,Z), H(X,Y,Z), and I(X,Y,Z) will be independent and unbiased.

Step 6 – output

– The message digest produced as output is A, B, C, D.

– That is, the output begins with the low-order byte of A, and end with the high-order byte

of D.

END

D- Data Distribution Stage

Data distribution stage in proposed system is intended to protect data shares in clouds. Firstly, the requirements of the consumer to achieve information of metadata, with username, password, cloud address, ID of data chunks and ID of data chunks stored in the multi-cloud. All of these metadata are saved in the metadata table (table 1).

Table 1: Metadata table

Username	Password	Cloud address	The ID of data chunks	The ID of data chunks stored in the multi-cloud

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

The shares that encrypted are saved in the equivalent three-clouds depending on table (1). The data distirbuted algorithm (DDIA) is described in algorithm (5).

Algorithm(5): Data Distibuted Algorithm

Input: *Image path, Number of Spilt, Access Token Code*
Output: Distributed Shares *Ni* stored in multi-cloud storages *Si (S1, S2, S_n)*
Step 1: Read Encryption Image use Image path
Step2: Generates the metadata table
Step 3: SET Cloud request Info as Media Type with value (application/octet-stream), Character Encoding with value (ISO-8859-1), Request Method with value (post), User signature with value (*Access Token Code*)
Step 4: Generate Encryption Image Spilt using Spilt Shares Image Algorithm
Step 5: FOR each Spilt Image Encryption DO
Step 6: Generate Encryption Image Spilt name using Data Split Rename Algorithm
Step 7: Upload Spilt Image to Cloud by *webwrite* function with *(Cloud address, Encryption Image data, Cloud request Info).*
Step 8: ENDFOR

Algorithm(6): Data Retrieval Algorithm

Input: *Image URL, Number of Spilt, Access Token Code*
Output: Retrieval Shares *Ni* stored in multi-cloud storages S_i *(S1, S2, S_n)*
Step 1: ET Encryption Image path in User PC.
Step2: Reads the table of metadata.
Step 3: SET Cloud request Info as Media Type with value (application/octet-stream), Character Encoding with value (ISO-8859-1), Request Method with value (post), User signature with value (*Access Token Code*).
Step 4: FOR i=1 To *Number of Spilt DO*
Step 5: GET Encryption Image Spilt name using Data Split Rename Algorithm as S_i.
Step 6: Sends a request to the quantified cloud by use *User signature.*
Step 7: Download Spilt Image from Cloud by *webread* function with *(Cloud address, Cloud request Info) as save it as rawData.*
Step 8: Save *rawData* in S_i as image file format.
Step 9: INCREASE *i* value by 1.
Step 10: ENDFOR

E- Decryption Stage

Decryption stage is done by using RC6 algorithm as shown in algorithm (7).

Algorithm (7): Image Decryption by RC6 Algorithm

Input: *Cipher colored Image, w-bit round keys S[0,..., 2r + 3]*
Output: *Origin colored Image as M_i ($M2_{RED}$, $M2_{GREEDN}$, and $M2_{BULE}$)*
Step 1: Spilt Cipher colored Image *into* Cipher Image $_{RED}$, Cipher Image $_{GREEN}$, and Cipher Image $_{BULE}$ layers
Step 2: SET *Number Of Byte* to 16 and VAR p =1 as Layer number in Cipher colored image.
Step 3: Calculate the number of all byte in Image by multiply number of Row with number of columns and save it as VAR *allPixle*
Step 4: FOR P=1TO 3, for each layer of Cipher Colored image (RED.GREEDN, BULE) DO
Step 5: Covert current image layer to ONE dimension Array as VAR *T*.
Step 6: SET VAR i =1 as Start index.
Step 7: WHILE i lass or equal (*allPixle-15)* DO
Step 8: M_p (i to (i+15)) = RC6Decrypt (T_p(i to (i+15)), S), call RC6 Decrypt algorithm
Step 9: INCREASE *i* value by *16*.
Step 10: ENDWHILE
Step 11: Covert M_p to TWO dimension Array as VAR $M2_p$
Step 12: INCREASE *p* value by *1*.
Step 13: ENDFOR
Step 14: Merge Origin image layers ($M2_{RED}$, $M2_{GREEDN}$, $M2_{BULE}$) in Origin colored Image

F-Data Merge Stage

The decrypted data are Merged using data merge algorithm as shown in algorithm (8).

Algorithm (8) Data Merge Algorithm

Input: *Image URL address* in server, *Number of Spilt*
Output: Origin Colored *Image*
Step1: DEFINE Origin Image as empty 3 dimensions Array with VAR name IMAGE
Step 2: SET VAR i =1, VAR j =1. (Use those VAR as Origin image indexes)
Step 3: FOR *Spilt Number* =1 to *Number of Spilt in cloud*
Step 4: Download spilt image by use Data Retrieval Algorithm from cloud using *Image URL address and save as VAR spiltImage*
Step 5: Calculate total number of *ROW* from number of row in *spiltImage*
Step 6: WHILE i lass or equal (number of column in *spiltImage multiply by Number of Spilt*) DO
Step 7: FOR index =1 to number of column in *spiltImage*
Step 8: SET *Image (i,j)* = *spiltImage* (i, index)
Step 9: INCREASE *i* value by 1
Step 10: INCREASE *j* value by 1
Step 11: ENDFOR
Step 12: SET VAR j =1 value to 1.
Step 14: ENDWHILE
Step 15: ENDFOR
Step 16: SET Procedures Output to *Image*

IV. EXPERIMENTAL ANALYSIS

The proposed model is implemented on an image (image 25KB) which is shown in figure (2). Note that the used computer is (Processor: core i5 and RAM: 4 GB).

Figure (2) image25KB.bmp

A. *Performance of Data Splitting*

Table 2 shows the execution time of splitting and renaming the (image25KB) into three shares.

Table 2: Execution Time for Data Splitting and Rename of image25KB

Image	*DSRA time*
Share1	*0.004 Seconds*
Share2	*0.001 Seconds*
Share3	*0.008Seconds*

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

B- Performance of Cryptographic Operation

RC6 block Cipher is applied on the color image "image 25KB" using an arbitrarily chosen key K = a s d f g h j k l a s d f g h j .

The results of encryption and decryption execution times are shown in table 3.

Table 3:Execution time for Encryption and decryption of image25KB

Image	*Encryption time*	*Decryption time*
Share1	*4.4 Seconds*	*4.5 Seconds*
Share2	*2.3 Seconds*	*2.4 Seconds*
Share3	*1.3 Seconds*	*1.6 Seconds*

B. *Performance of Data Integrity:*

The execution time for MD5 integrity of the three shares is shown in table (4).

Table 4: Execution time for integrity check by MD5

Image	*MD5 time*
Share1	*0.05 Seconds*
Share2	*0.04 Seconds*
Share3	*0.03Seconds*

C- Performance of Data Distribution and Retrieval:

The Distribution and Retrieval time using:(google drive, one drive, and dropbox) are described in tables (5 and 6) sequentially.

Table 5: Time for Data Distribution

a. Image b.	***Share1 using Dropbox***	***Share2 using Google Drive***	***Share3 using One Drive***
c. ***(25KB).bmp***	*1.4 Seconds*	*1.5 Seconds*	*1.2 Seconds*

Table 6: Time for Data Retrieval

d. Image e.	***Share1 using Dropbox***	***Share2 using Google Drive***	***Share3 using One Drive***
f. ***(25KB).bmp***	*1.5 Seconds*	*2.0 Seconds*	*1.0 Seconds*

D- Performance of Data Upload and Download:

The upload and download time using :(google drive, one drive, and dropbox) are described in tables (7 and 8) sequentially dependent on shares3 because provide high security.

Table 7: Time for Data Upload

Image	Share1 using Dropbox	Share2 using Google Drive	Share3 using One Drive	Proposed system
(25KB).bmp	2.748 Sec	2.848 Sec	2.548 Sec	8.144 Sec

Table 8: Time for Data Download

Image	Share1 using Dropbox	Share2 using Google Drive	Share3 using One Drive	Proposed system
(25KB).bmp	3.14 Sec	3.64 Sec	2.64 Sec	9.42 Sec

E-Performance of Data Merge

The time of data merging for the three shares is described in tables (9).

Table 9: Execution time for data merge

Image	DMA time
Share2	0.03 Seconds
Share3	0.001 Seconds

V. CONCLUSION

Cloud computing is a technology that considers hopeful and promising for the IT applications in the next generation. The proposed model is successfully implemented in uploading side with total time of (8.144 sec), and in downloading side with total side of (9.42). The difficulty in comparison process because of no previuos work had the same steps with the proposed work as shown in table (7). However, the main purpose of this paper is to get secured data using distributed cloud technology, which had been done. Therefore, the important conclusion that has been reached from this work that the security using distributed cloud computing had added a significant time in the proposed model.

REFERENCES

1. Shabir, Muhammad Yasir, et al. "Analysis of classical encryption techniques in cloud computing." Tsinghua Science and Technology 21.1 (2016): 102-113.
2. Muhammad Yasir Shabir, Asif Iqbal, Zahid Mahmood, and AtaUllah Ghafoor, Analysis of Classical Encryption Techniques in Cloud Computing (2016), Vol. 21, pp. 102–113.
3. Shivanna, K., S. Prabhu Deva, and M. Santoshkumar. "Privacy Preservation in Cloud Computing with Double Encryption Method." Computer Communication, Networking and Internet Security. Springer, Singapore, 2017. 125-133.
4. Yesilyurt, Murat, and Yildiray Yalman. "New approach for ensuring cloud computing security: using data hiding methods." Sādhanā 41.11 (2016): 1289-1298.
5. Gutte, Vitthal S., Amol N. Jadhav, and Pramod Mundhe. "Image management and data access control for Multi-authority cloud storage with use of certificate less encryption." 2016 IEEE International Conference on Advances in Electronics, Communication and Computer Technology (ICAECCT). IEEE, 2016.

6. El-Booz, Sheren A., Gamal Attiya, and Nawal El-Fishawy. "A secure cloud storage system combining time-based one-time password and automatic blocker protocol." EURASIP Journal on Information Security 2016.1 (2016): 13.
7. Bala, Yogesh, and Amita Malik. "Biometric inspired homomorphic encryption algorithm for secure cloud computing." Nature Inspired Computing. Springer, Singapore, 2018. 13-21.
8. Van Dijk, Marten, and Ari Juels. "On the impossibility of cryptography alone for privacy-preserving cloud computing." HotSec 10 (2010): 1-8.
9. Oliveira, Paulo F., et al. "Trusted storage over untrusted networks." 2010 IEEE Global Telecommunications Conference GLOBECOM 2010. IEEE, 2010.
10. Ateniese, Giuseppe, et al. "Improved proxy re-encryption schemes with applications to secure distributed storage." ACM Transactions on Information and System Security (TISSEC) 9.1 (2006): 1-30.
11. Mahmood, Ghassan Sabeeh, Dong Jun Huang, and Baidaa Abdulrahman Jaleel. "Data Security Protection in Cloud Using Encryption and Authentication." Journal of Computational and Theoretical Nanoscience 14.4 (2017): 1801-1804.
12. Abdullah, Salma H., Janan A. Mahdi, and Ashwaq T. Hashim. "A proposed 512 bits RC6 encryption algorithm." IRAQI JOURNAL OF COMPUTERS, COMMUNICATION AND CONTROL & SYSTEMS ENGINEERING 10.1 (2010): 11-25.
13. Pereira, Sancha, et al. "Storekeeper: A Security-Enhanced Cloud Storage Aggregation Service." 2016 IEEE 35th Symposium on Reliable Distributed Systems (SRDS). IEEE, 2016.
14. Alsolami, Fahad, and Terrance E. Boult. "CloudStash: using secret-sharing scheme to secure data, not keys, in multi-clouds." 2014 11th International Conference on Information Technology: New Generations. IEEE, 2014.
15. Rivest, R.L., Robshaw, M.J.B., Sidney, R., & Yin, Y.L (1998a). "The RC6 Block Cipher."URL:ftp://ftp.rsasecurity.com/pub/rsalabs/rc6/rc6v11.pdf
16. *Deepakumara, Janaka, Howard M. Heys, and R. Venkatesan. "FPGA implementation of the MD5 hash algorithm." Canadian Conference on Electrical and Computer Engineering 2001. Conference Proceedings (Cat. No. 01TH8555). Vol. 2. IEEE, 2001.*