www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 3 Issue 4 April, 2014 Page No. 5215-5223

Data Encryption and Decryption Algorithms using Key Rotations for Data Security in Cloud System

Prakash G L[1] ,Dr. Manish Prateek[2] and Dr. Inder Singh[3]
[1]Research Scholar, Department of Computer Science and Engineering, UPES, Dehradun, Email:glprakash78@gmail.com
[2]Associate Dean, Centre for Information Technology, UPES, Dehradun
[3]Assistant Professor, Centre for Information Technology, UPES, Dehradun

Abstract:-Outsourcing the data in cloud computing is exponentially generating to scale up the hardware and software resources. How to protect the outsourced sensitive data as a service is becomes a major data security challenge in cloud computing. To address these data security challenges, we propose an efficient data encryption to encrypt sensitive data before sending to the cloud server. This exploits the block level data encryption using 256 bit symmetric key with rotation. In addition, data users can reconstruct the requested data from cloud server using shared secret key. We analyse the privacy protection of outsourced data using experiment is carried out on the repository of text files with variable size. The security and performance analysis shows that the proposed method is highly efficient than existing methods performance.

Keywords: Data Block, Security, Outsource, Encryption, Decryption, Key Rotation.

1. INTRODUCTION

Cloud computing provides on-demand resource access from a shared pool of computing resources such as; hardware and software for efficient manage. By outsourcing the user data to the public cloud environment, this decreases the control of data for data owner. To maintain the control of data in rest or data in motion within networks, offers more advantages for data security.

Protecting data in the cloud, authentication and integrity, access control, encryption, integrity checking and data masking are some of the data protection techniques. Cryptography is the one of the efficient method for data security in cloud computing. This includes the design and implementation of an efficient encryption and decryption algorithms. In symmetric cryptography, before outsourcing data to cloud server
is encrypted into cypher text using secret key and later user decrypted using same shared secret key.

Encryption is the one of the way to protect data at rest in cloud server. There are four ways to encrypt the data at rest, such as; full disk level, directory level, file level and application level. The most critical part for implementation of any of these methodsis key management for data encryption and decryption. The common way to protect data in motion is to utilize encryption with authentication, which safely passdata
to or from the cloud server [1].

From the perspective of protecting data privacy, the data owners rely on TTP for the storage security of their data. Moreover, there are legal regulations, such as the U S Health Insurance Portability and Accountability Act (HIPAA) [2], further demanding the outsourced data not to be leaked to external parties. Exploiting data encryption before outsourcing is one way to the privacy preserving public auditing scheme.

In cloud computing, data owners become increasingly outsource their sensitive data in encrypted form from local system to public cloud for more flexibility and economic savings [3]. To protect data in transit to and from the cloud as well as data stored in the cloud, efficient data encryption and decryption algorithms are used for security. The block diagram of symmetric key encryption and decryption data storage as shown in the Figure 1. It involves the use of single secrete key for both encryption and decryption. Data owner split the file into smaller blocks and encrypts all the blocks using symmetric secrete key before sending in to the cloud service provider. Then the cloud service provider stores all the encrypted blocks of source file in cloud server[4].

When the authorized user request a file from the cloud server, cloud service provider gets the encrypted file blocks from the cloud server and send to the user. After receiving all the requested blocks from the cloud server, user decrypt all the encrypted blocks using same secrete key before accessing it.

The rest of the paper is organized as follows; The existing cloud data security methods and its performance are presented in the Section II. Section III and IV, introduces the system model and mathematical model for the proposed system respectively. The detailed data encryption, decryption and access algorithms are presented in the section V and section VI, gives the result and performance analysis. Finally, the overall proposed method and future enhancement concludes in section VII.

2. RELATED W ORK

Jing-Jang Hwang et al. [5], has proposed a business model for cloud computing for data security using data encryption and decryption algorithms. In this method cloud service provider has responsible for data storage and data encryption/decryption tasks, which takes more

computational overhead for process of data in cloud server. The main disadvantage of this method is, there is no control of data for data owner i. e, data owner has completely trusted with cloud service provider and he has more computational overhead.

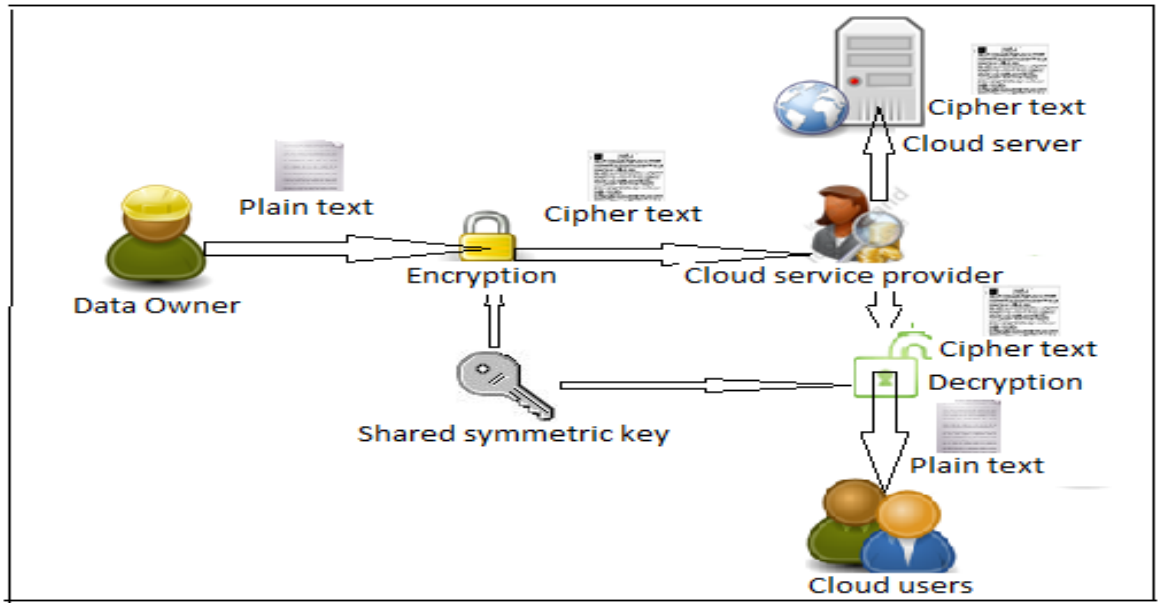

Figure 1: Block Diagram of Data Encryption and Decryption in Cloud System

Junzuo et al. [6], proposed an Attribute Based Encryption (ABE) and verifiable data decryption method to provide data security in cloud based system. They have been designed the data decryption algorithm based on the user requested attributes of the out sourced encrypted data. One of the main efficiency drawbacks of this method is, cloud service provider has more computational and storage overhead forverification of user attributes with the outsourced encrypted data. While introducing third party auditor we can reduces the storage, computation, and communication overheads of the cloud server, which improves the efficiency of the cloud data storage.

FatemiMoghaddam et al. in [7], discussed the performance of six different symmetric key RSA data encryption algorithmsin cloud computing environment. They have proposed two separate cloud servers; one for data server and other for keycloud server and the data encryption and decryption process at the client side. The main drawback of this method is tomaintaining two separate servers for data security in cloud, which creates a more storage and computation overheads.

3. PROBLEM FORMULATION

3.1 Block Status Table(BST)

The Block Status Table(BST) is a small data structure used to access the outsourced encrypted file from the cloud serviceprovider. It consists of two column such as *SNj* and *BNj* , where *SNj* is the sequence number of physical storage of data block j in the file and *BNj* is the data block number. Initially the data owner stores table entries as *SNj = BNj = j*. For insertion of data blocks, the BST is implemented using linked list. The structure of BST for insertion of data blocks as shown in the Table 1.

Table 1 : BST

Sequence number	Block number
1	1
2	2
3	3
4	4

3.2. System Model

The cloud data storage system model for secure data access sequences are explained in the Figure 2. The following sequence numbers are represented for data storage and access operations in cloud server.

The data owner splits the source file in to blocks of 128 characters and encrypt all the blocks using efficient encryption algorithm and prepare the Block Status Table(BST) for encrypted blocks, then send the encrypted file, key, BST to the Trusted Third Party(TTP) auditor.

1. The TTP calculate the combined hash values for BST(TH) and encrypted file (FH), then send only encrypted file and BST to the cloud server for storage.

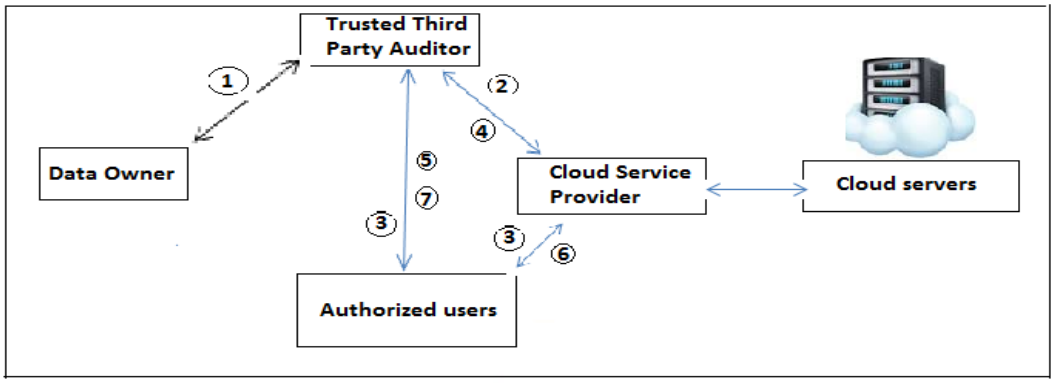

Figure 2: Block Diagram of Encrypted Data Storage

2. The authorized user sends the data access request to both TTP and cloud server.

3. TTP verifies the authorized user, if the user is verified then, it send the authorization signal to the cloud server.

4. TTP send the hash values of BST (TH) and encrypted file (FH) to the requested user.

5. Cloud server send the BST and encrypted file to the user.

6. User calculate the hash values of BST and encrypted file received from the cloud server then verifies with hash values received from the TTP.

7. If both values are verified then user gets a data decryption key and decrypt the data blocks.

3.3. Objectives

Data security for outsourcing and accessing data fromcloud server, our proposed security model achieves the following objectives.

i. Lightweight overhead: design a lightweight computation, storage and communication overhead for verification of authorized cloud users and access the cloud date.
ii. Block level data operation: design an efficient block level encrypted data operations
iii. Confidentiality and integrity: design an efficient dataencryption before outsourcing to cloud server and decryption algorithms at user side.

4. MATHEMATICAL MODEL

4.1. Notations

The various symbols are used in this paper for the encryptionand decryption algorithms is listed in the following Table 2.

4.2. Definitions

File Chunk Size: The security is provided at the blocklevel. The file is divided into blocks and confidentiality isensured on every block and finally on file. The block size isfixed for experimental purpose.

Data Owner File: The file or the content that data owneris looking for confidentiality. And file is a set of blocks andfile size depends on the block size and defined as below inequation.

Table 2: Notations used in the paper

Symbol	Meaning
τ_b	File Chunk Size /Block size of block b
χ	Encryption Key
F	Data Owners file targeted for Encryption
b	File chunk/block.
E_m	Encoding Map for every Character
b_c	Binary Equivalent of Chanater c
$\overline{CA}$	Circular Vector of Characters.
C_{ch}	Cipher Text for character for ch.
φ_F	Size of a file F

$$F = \{ b_1, b_2, b_3, \ldots b_m \}$$

Similarly every block and an Encryption key is a set ofcharacters as defined below,

$$b = \{c_1, c_2, c_3, \ldots |b| \} and \chi = \{k_1, k_2, k_3 \ldots |\chi|\}$$

The file size is defined as the summation of its componentblock size.

$$\Phi_F = \sum_{i=0}^{|F|} |\tau_{b_i}|$$

|Encoding Map(Em): The encoding Map/Encoding tableis a map between a character to every other random characterin ASCII range. The ASCII value of a character is splitted intodigits and the characters from every digit position is summedup in ASCII range to find new character as follows, Let rc bea random character for c and ac is a set of digits forming aASCII value of Character c,

$$\overline{ac} = \{ d_i \mid \forall d_i \in Z^+, 0 \le i \le |\overline{ac}| \}$$

The random character ASCII value is defined as ,

$$ar_c = \sum_{i=0}^{|F|} b_i \ \% 256$$

Circular Array(CA): The circular array is used in bothencryption and decryption process. The circular array is usedfor shift operations on both character and on a key character.

The binary equivalent of a character is stored on array andhence the values are either 0 or 1. The circular array has thevalue obtained as a result from signed right shift operation.The shift operation is performed to disguise the informationby changing its bits position and defined as below in equation,

$$CA = \{v_i | 0 \le i \le |CA|, v_i \in \{0,1\}\}$$

Key Chooser(KC): The key chooser is a vital componentwhich defines the criteria for selecting key character fordisguising the block character of a file. The key character isselected in such a way that, if 1st chunk character is selectedfor encryption then first character of key is considered forencryption, if a selected block character comes outside therange of key size, modulus of block character position tokey size is performed to fetch a key. Two Key characters areselected for every block character if ith character of a block i.e.ci is chosen for encryption then its corresponding key characterat position i is selected as below,

$$\chi_i = \chi_{i\%|\chi|}$$

Similarly second key character χj is selected from the thirdposition away from χi

$$\chi_j = \chi_{i+2\%|\chi|}$$

CA Inverter (CAI): The CA Inverter inverts/complementsthe circular array for high degree of security. The criteriaon which the complements happens based on the resultantnumber obtained after processing the adjacent key characters.

The ASCII values of adjacent key characters are added, if theresultant is even then CA in inverted. Let CA be the circulararray having binary equivalent of block character ci and let χibe the chosen key character, then Inverted CA is defined asbelow

$$ICA = \neg CA, \ iff \ \chi_i + \chi_{i-1} \% 2 = 0$$

CA Shifter (CAS): The CA shifter shifts/rotates the circular array. The stepper movement for circular array is basedon the summation of two key characters. If the summation isa factor of 5 then circular array is moved by 2 else it is movedby the remainder obtained from the division of summation by5 as below,

$$SCA = CA >>> (2|(\chi_i + \chi_j)\%5)$$

Where χi , χj are the chosen key characters, SCA is theshifted Circular for ith block character ci and CA is a CircularArray storing binary value of ci.

Encryption Engine/Cipher Engine: Encryption Engine isa black box which takes block character to produce ciphercharacter. The Encryption Engine is composed of above threecomponents (Key Chooser, CA Inverter and CA Shifter) in thatorder as below, Let ci be the ith block character and CE bethe cipher engine

$$CE = KC \bigcup CAI \bigcup CAS$$

Decryption Engine/Decipher Engine(DE): The Decryption Engine is composed of same components as EncryptionEngine but these components are applied in reverse order asbelow in equation .

$$DE = CAS \bigcup CAI \bigcup KC$$

5. ALGORITHMS

The data owner encrypts the file before sending it to theCloud Service Provider(CSP)[9],[10]. The encryption algorithm has several steps and composed of key Chooser, CircularArray Inverter and Circular Array shifter.

The encryption algorithm is designed, the information athighest factor by applying series of rotations on every blockcharacter and the key is rotated for every character. Fromthis it is ensured that same key is not used for encryptingevery character and hence this algorithm is called as key motorencryption algorithm.

The file is divided into blocks and confidentiality is emphasized on every character level of a block. The binary equivalentof block character is stored in circular array and number ofmoves the circular array is rotated is decided by the CAShifter. Where every rotation divides the data by 2 and thiswill optimize the data to its least value and hence the privacyof data is ensured. Since stepper movement of CA is differentfor different character it's hard/impossible to determine theactual value of CA as explained in Table 2.

The key portion of algorithm is the CA inverter and CA shifter which is performed on every block character and finally on entire block. If File has *N* blocks and if every block has n characters then CAI and CAS is performed by $N*n$ operations. And therefore this algorithm has complexity of $O(N * n)$.

When the user wants to access data from cloud server, the user authorization and data verification procedure is explained in Table 4. The decryption process happens exactly opposite to encryption which finds a block character from cipher text as per equation (13). The algorithm in Table 5, suggests that CA shifter is performed first then Key chooser component is used to select two keys and they are added before inverting CA. Since CA already contains complemented value and complement of CA now yields original encoded value. The Encoding Map (*Em)* is searched to get its original character. The Algorithm has same complexity as Encryption.

Table 3: Data Encryption Algorithm

Algorithm Data_Encryption(Source_file(F), χ, Decrypted_file)

Input: F -File for encryption, χ - Encryption Key
Output: CF - Ciphertext of a file F

1. Split the characters of the file/string into blocks.
2. Get the mapped value from E_m.
3. Get the binary equivalent of the Current Character.
4. Remove the first character from the binary value and store it into First Character and consider rest of binary value for shift operations (this is done to avoid having case the resultant value 00001 or 01111 after, shift operations this would result into forgetting a bit as 0001 can also represented as 1)
5. Store binary value into a CA for bit operations.
6. The key chooser component selects a key character from ith position, such a way that if chunk character is selected for encryption then select the first character of the key, so i is selected chunk character > size of key ($|\chi|$) get the modulus of chunk character position.
7. Add the selected ith key character int value and its previous $(i-1)^{th}$ character int value instance if 1st key character is selected then the previous character would the 16th character (key stored into circular array doubleor way linked list for this operations)
8. The CA Inverter component complements Circular array (ICA) if the summed result is even per the equation (10)
9. Add the selected ith key character and $(i+2)^{th}$ key character.
10. The CA shifter shifts the Circular to find SCA as per the equation (11).
11. Add the stripped off first Character to the resultant binary value of Circular Array obtained previous step.
12. Get the character equivalent of the binary value which is the cipher character.
13. Repeat steps 2 through 12 for all the chunks with different Key (by shifting the key character using Circular Array a double way linked list).

6. EXPERIMENTAL RESULTS

6.1. Performance Analysis

The experiment is carried out on the repository of text files with varying size. For testing purpose the text file is composed of alphanumeric characters. The Encoding Map is restricted to have mapping values for lower case alphabets and numerical values. The key used for experimental purpose is "***doitdueletscshec"*** which is of 256 bits in size. The key size is fixed for experimental purpose. The file is divided into blocks of 256 characters i.e 4096 bits in size. The algorithms are implemented in JAVA. The eclipse IDE and Linux OS forms the complete execution environment. The part of the source file data is encrypted before storing in cloud server as shown in Table 6.

Table 4: Data Access Procedure

Algorithm Data Access Procedure

1. An authorized user sends a request to both the CSP and TTP auditor.
2. CSP sends the encrypted file and BST to the requested user.
3. TTP sends the FHttp, THttp and key to the user.
4. user computes the THuser using THcsp and FHuser using data blocks and compared with THttp and FHttp respectively for integrity check.
5. Then the user verifies the file by comparing FHuser and FHttp
6. If both BST and file computed hash values are maches, then user decrypt the file using shared secret key

Table 5: Data Decryption Algorithm

Algorithm Data_Decryption(Ciphertext of a file(F), χ, Source file

Input: CF - Ciphertext of a file F, χ - Encryption Key
Output: Source file (F)

1. Split the characters of the file/string into blocks.
2. Get the binary equivalent of the current cipher character.
3. Remove the first character from the binary value and store it into first Character variable and consider the rest of binary value for shift operations.
4. Store binary value into a CA.
5. Select a key character from ith position, such a way that if 1st chunk character is selected for encryption then select the first character of the key, so $i = 1$. if selected chunk character $>$ size of key $(|\chi|)$ get the modulus of chunk character position.
6. Add the stripped off first Character to the resultant binary value of CA after bit operations.
7. Add the selected $|\chi|_i$ and $|\chi|_{i+2}$ key character.
8. The CA shifter shifts the Circular to find SCA as per the equation (11).)
9. Add the selected ith key character int value and its previous $(i-1)^{th}$ character int value, for instance if 1st key character is selected then the previous character would the 16^{th} character (key is stored into circular Array or double way linked list for this operations)
10. Get the character equivalent of the binary value.
11. Get the mapped value from E_m, which is the decrypted value of the character.
12. Repeat steps 2 through 11 for all the chunks with different Key (by shifting the key character right using Circular Array a double way linked list)

The vital or key operations in both processes are CAshifter and CA inverter. The time for encryption/decryptionis directly depends on these two operations. The numberof movements of CA and its inversion process decides theaccuracy of encryption/decryption. Along with CA shifter thekey is also rotated for every block character. This ensures thatsame key is not used for multiple characters. The analysisis performed on different files and number of

movementsused for shifting CA and key remains same. From Figure3, for the file with size 313 characters the number ofmovements performed was 646, similarly for file with size3139 it is 6479 movements which is almost double the file size.In other words every movement considers 0.5 characteri.e half the character which is 8 bits. This infers that shiftoperation is performed on every byte and hence the data isdisguised at fine level(byte level).

The other parameter for analysis is the CA inverter operation. As shown in figure 1, 118 times the complement isperformed for the file size of 313 characters similarly forthe file size of 3139 characters the number of complementsperformed was 1178 which is approximately 3 characters. Thisimplies that complement operation is performed for every 3characters and hence the data is disguised at coarse level(byteslevel). As CA shift is performed for every character comparedto complement operation it has more impact on the encryption/decryption process and therefore it can be concluded thatencryption is happening at finer level i.e byte level. In figure4, the execution time is plotted on the graph. With increase infile size the number of movements and complements are highand hence the execution time is directly proportional to filesize. It is observed that decryption is taking more time thanencryption process.

6.2. Advantages

To retain control over data in cloud environment, theencryption and strong key management is more important tothe organization to meet the security challenges. The benefitsof the encryption in cloud environment are;

- Encryption ensuring the privacy of the organizationdata, while encrypted data is in the transmission, inuse and at storage location.
- Encryption Helps Achieve Secure Multi-Tenancy inthe CloudEncrypting data in the cloud and holding encryptionkey data owner can avoid the cloud service providerto access the data.
- Encryption Provides Safe Harbor from Breach Notification, If a data breach occurs and personally identifiable information is lost, the breached party mustnotify all individuals who are impacted.
- Encryption Provides Confidence of data backups aresafe in cloud environment from the breached party.
- Encryption can expand revenue potential to customerswith sensitive or regulated data by maintaining the keyby cloud data owner and gives cloud service providersa competitive edge.

Table 6: File text and Cipher text

File Text	Ciphertext
a b c d e f g h i j k l m n o p q r s t u v w x y z 1 2 3 4 5 6 7 8 9 letasdfsadf sdf dkfjlsfdjsad sadflkjasdlfasdfasdfsaldfjalsjfasdfalsdjflasdjflsa lsajfdlasjdflasjdflsajdflsajdflasjdflasjdflajsdfljas asdlfjasdfldsfasdflksajflasjflksadflksajdflasjdflk dfljasdlfkjaslkdfjajfldsajflaksjflkasjdflkajs	|@b@M@Q@Y@KG@p[@r@'@W@|@4N @Wz@#@m@s@*@iH@;m@F@%@9@\@ 5j@<J@;@K@B@QlLamWVv|nt@mnt@],K YndVQMICWi@vO]Kh9e^RQhKCdbKORn Km^QQtM|WdeKCR]K@and]rKhCdbMtJR|n YiTRCMKQTFIrkVh|[eWVWI|rWihvOZnKkF dZQKhCdentJyMIhiTZCIntTe|vkQKMamWV WnIt^mntJ8I|YiTyvMKW&m|kVh,[Ff]VW|IrV h3OZInZnfyvnht&e|vJ8nKYFfVWnICfihC#RN Kk'^RrnKW&FMv

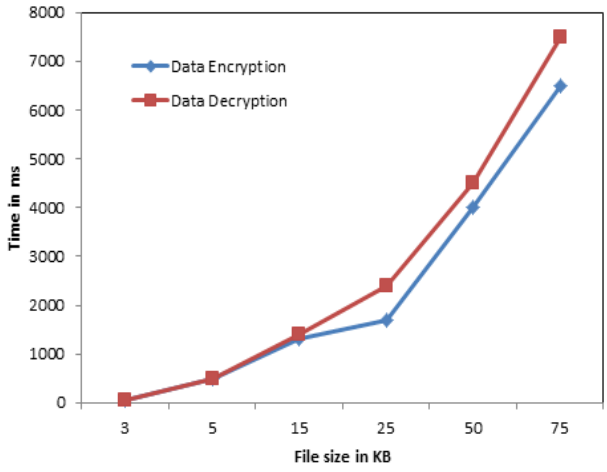

Figure 3: Data Encryption and Decryption time comparison.

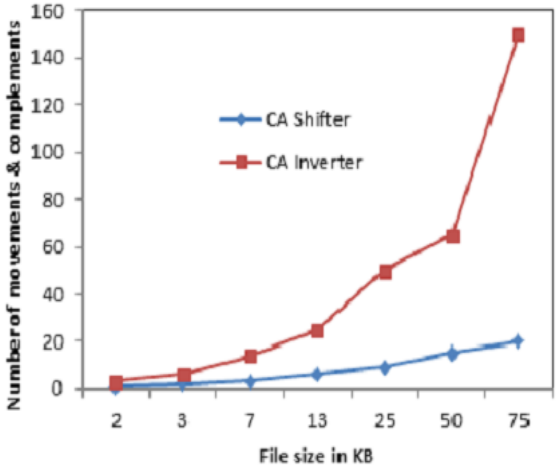

Figure 4: CA Shifter and CA Invertor comparison

7. CONCLUSION AND FUTURE ENHANCEMENT

In this paper, we have proposed an efficient data encryption and data decryption algorithm to protect the outsourcedsensitive data in cloud computing environment.

With dataencryption, data owner can utilize the benefits of file splittingto reduce storage and computational overheads. On the otherhand, to reduce the burden of data owner, trusted third partyis introduced for verification of authorized users to access thedata from cloud server. We demonstrate the performance ofencryption and decryption algorithms in terms of data privacy,computational efficiency and effectiveness of the cloud storagesystem. On top of this architecture, we can also demonstratefor dynamic block level operations on encrypted data blocksfor insertion, deletion and update, which we consider is ourimprovement future work.

REFERENCES

[1]. J R Winkler, Securing the Cloud: Cloud Computing Security Techniquesand Tactics, *Elsevier Inc.*,USA, 2011.

[2]. http://aspe.hhs.gov/admnsimp/pl104191.htm, *104th United StatesCongress, Health Insurance Portability and Accountability Act of1996*.

[3]. Tim Mather, SubraKumaraswamy, and ShahedLatif, Cloud Securityand Privacy,*Published by O Reilly Media, Inc.,* 2009.

[4]. http://security.setecs.com, Security Architecture for Cloud ComputingEnvironments, *White paper,* 2011.

[5]. Jing-Jang Hwang, Taoyuan, Taiwan,Yi-Chang Hsu, Chien-Hsing Wu, ABusiness Model for Cloud Computing Based on a Separate Encryptionand Decryption Service, *in International Conference on InformationScience and Applications (ICISA),* pages 1-7, 2011.

[6]. Junzuo Lai, Deng R H, Chaowen Guan, JianWeng, Attribute-BasedEncryption With Verifiable Outsourced Decryption, *in IEEE Transactions on Information Forensics and Security, vol. 8(8), pages 1343-1354,*2013.

[7]. FatemiMoghaddamF,KarimiO,Alrashdan M T, A Comparative Studyof Applying Real-Time EncryptioninCloud Computing Environments, *inIEEE 2nd International Conference onCloudNetworking (CloudNet),pages 185-189*, 2013.

[8]. Lan Zhou, Varadharajan V, Hitchens M, Integrating Trust with Cryptographic Role-Based Access Control for SecureCloud Data StorageTrust, *in 12th IEEE International Conference on SecurityandPrivacy inComputingand Communications (TrustCom),* pages 560-569, 2013.

[9]. Qin Liu, Tan CC, Jie Wu, Guojun Wang, Reliable Re-Encryption in Unreliable Clouds, *in IEEE International Conference on Global Telecommunications(GLOBECOM),*pages 1-5, 2011.

[10]. Miwen, Rongxinglu, Kuanz hang, Jing Shenglei, Xiaohuiliang andXueminshen, PaRQ:A Privacy-Preserving Range Query Scheme OverEncrypted Metering Data for Smart Grid, *in IEEE International Journalof Computer Networks*, pages 178-191, 2013.