DEDUP: COMPARATIVE ANALYSIS OF CLOUD SECURITY AND SPACE CONSERVATION MECHANISMS

Akshay Sharda
Department of Computer Engineering & Technology
Guru Nanak Dev University
Amritsar, Punjab, India

Amit Chhabra
Department of Computer Engineering & Technology
Guru Nanak Dev University
Amritsar, Punjab, India

Abstract:- **Cloud computing provides the resources to the clients on pay per use basis. Cloud computing resources including storage is heavily used by clients. The service level agreement is maintained for ensuring the service availability to the users. As more and more users interact with the cloud computing resources, risk of security arises. In order to tackle the issue, security mechanisms including encryption is used within cloud computing. This paper presents comparative study of techniques used to preserve space and provide security as well. Primary stress is paid towards the deduplication mechanism used to ensure conservation of space and also secure transmission of data at server end. Techniques include block level deduplication, message level deduplication and bit level deduplication. The performance analysis indicates better results of bit level deduplication in terms of space conservation and key size.**

Keywords: Cloud security, Encryption, space conservation, key size.

1. INTRODUCTION

Cloud usage is increasing day by day [9]. Users of the cloud may have uncertain intentions causing threat to cloud resources. To overcome the issue, security mechanisms are considered for evaluation. The security mechanisms of cloud primarily considered in this literature include encryption and storage conservation Space conservation in deduplication mechanism decrease the size causing cost of storage to go down significantly and encryption mechanism applied to the space conserved message increases complexity of key associated with message to be transmission. The cloud computing deduplication mechanism works on two key aspects to increases performance and provides user confidence in utility of cloud.

a. Space Conservation
b. Encryption

a. Space Conservation

Cloud provides resource to client on pay per use basis [8]. Space conservation preserve space and hence reduce cost of cloud usage. Space conservation mechanism used in cloud could be many. This section provides in-depth study of mechanisms of space conservation in cloud.

- **Message Level Redundancy tackling mechanisms**

In this approach, Message terminated by the paragraph is checked for redundancy [1]. This means that in case paragraph repeats then that paragraph is eliminated from the file uploaded on the cloud. This mechanism compress the size of uploaded file but compression mechanism used in this case is inconsistent in nature. The inconsistency is primary due to loss of critical data. The mechanism ensuring redundancy tackling is described through example 1.

Example 1: The file named abc.txt is meant to be stored on the cloud. This file contains text as

First Paragraph
" Cloud computing provides resources on shared basis. Cost is on the basis of pay per use. The use of cloud significantly reduces the expense of purchasing individual resources by the clients."

Second Paragraph
" Cloud provides layered architecture to reduce the complexity of operation. These layers are categorized as IAAS, PAAS and SAAS".

Third Paragraph
" Cloud computing provides resources on shared basis. Cost is on the basis of pay per use. The use of cloud significantly reduces the expense of purchasing individual resources by the clients."

The message level redundancy handling mechanism allows third paragraph to be discarded since third paragraph and first paragraph is similar. Thus size of the file to be transmitted is significantly reduces. The main problem of this approach is critical data is also going to be lost as entire words from the paragraph is lost if number of matched words exceeded threshold level.

- **Byte Level Redundancy handling mechanisms**

The byte level mechanism compares individual words within the paragraph rather than entire paragraph at a time [4]. This will cause only redundant words to be eliminated rather than entire paragraph. Hence this approach is more reliable as compared to paragraph redundancy tackling mechanism. The example 2 provides the working of this approach.

Example 2: The file named abc.txt is meant to be stored on the cloud. This file contains text as

First Paragraph

" Cloud computing provides resources on shared basis. Cost is on the basis of pay per use. The use of cloud significantly reduces the expense of purchasing individual resources by the clients."

Second Paragraph

"Cloud provides layered architecture to reduce the complexity of operation. These layers are categorized as IAAS, PAAS and SAAS".

Third Paragraph

" Cloud computing provides resources on shared basis. Cost is on the basis of pay per use. The use of cloud significantly reduces the expense of purchasing individual resources by the clients."

The third paragraph is similar to the first paragraph. In first approach , entire words from the file is eliminated but in byte level deduplication mechanism individual words are checked and replaced with the white space characters to handle redundancy. After applying Byte level deduplication mechanism, result of example 2 is given as

Result after Byte level deduplication for example 2

" Cloud computing provides resources on shared basis. Cost is _ the basis of pay per use. _ _ of _ significantly reduces _ expense _ purchasing individual resources by _ clients."

The above listed result indicates that similar words are replaced with the white space characters indicated through "_" character. This mechanism reduces the size of file to be reduced before uploaded on cloud.

- **Bit level Redundancy handling**

In this approach, the file to be uploaded is checked character by character and eliminates the redundant characters [7]. The bit level mechanism uses the concept of indexing. The indexing mechanism substitute index number in place of similar text. Thus each byte is replaced with individual bits. The main advantage of this approach is non lossless compression. The mechanism is described through the example 3.

Example 3: The file named abc.txt is meant to be stored on the cloud. This file contains text as

" Cloud computing provides resources on shared basis. Cost is on the basis of pay per use. The use of cloud significantly reduces the expense of purchasing individual resources by the clients."

The maintained index for the same is given in table 1

Table 1: Indexing in bit level deduplication

Word	Index
Cloud	1
Computing	2
Provides	3
Resources	4
On	5
Shared	6
Basis	7
Cost	8
Is	9
On	10
The	11
Of	12
Pay	13

Per	**14**
Use	**15**
Significantly	**16**
Reduces	**17**
Expense	**18**
Purchasing	**19**
Individual	**20**
By	**21**
Clients	**22**

The Result obtained after the redundancy handling mechanism is complete is reduced in size and is of the following form

" Cloud computing provides resources on shared basis. Cost is on the 7 of pay per use. 11 use 12 1 significantly reduces 11 expense 12 purchasing individual 4 21 11 clients."

The size of modified file is reduced significantly using bit level deduplication mechanism.

b. Encryption mechanism used in deduplication

Encryption mechanism used in deduplication to provide security of file being uploaded in cloud [2]. The deduplication procedure first of all reduces the size of the file and then encryption if applied to enhance security of file being uploaded. Distinct security procedure available to be used within cloud computing is discussed in this section.

- **RSA**

This mechanism is used to provide security to data being uploaded to the client [6]. This is one of the simplest and oldest methods of encryption. This technique is also known as public key encryption mechanism since key is shared among source and destination ends. The procedure for RSA encryption initialize by setting up two largest prime numbers 'p' and 'q'. The distributed product n=p.q is obtained where 'n' is the remainder used for public and private key at the source and destination end. Next step is to obtain quotient that is achieved using Q=(p-1)(q-1). Greatest common divisor is applied to obtain the public and private key for encryption and decryption purpose.

- **DES**

This is block level encryption mechanism that is 64 bit in nature [3]. In other words 64 bit block is formed using data encryption standard. In this mechanism, complementation procedure is employed. This mechanism symbolic representation is given in equation 1

$$E_k(P) = C \text{ or } E_k(P') = C'$$

Equation 1: Complementary property exhibited by DES.

The mechanism of parity bits is used within data encryption standard. In parity bit mechanism number of '1' within data is given a count. In case number of '1' is even then the data is said to be even parity. In case data is not in even parity then it is converted into even parity by assigning '1' equal to the requirements of making it even.

- **Bit Level Encryption mechanism**

This mechanism is based on ASCII codes and position of words within given string of the file [5]. The indexing is used in order to convert the file into normalized form. The normalized file is then encrypted by replacing the characters by equivalent ASCII code. The working of this mechanism is described by considering example 4

Example 4: Consider the file abc.txt having the following text

"Cloud computing become need of the hour to provide resources as and when needed by the clients. The need of the cloud is prolonged for reducing the cost metric associated with resource usage"

The mechanism of encryption is given as under

According to index tables maintained, the conversion to nominal form is the first step in the conversion process.

After first phase is complete file abc.txt is reduced to following form
"1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 3 16 17 5 18 19 20 21 22 23 24 5 25 26 27 28 29 30"

The nominal conversion reduces the size of the file being uploaded on the cloud. The ASCII conversion of abc.txt converts the file into transmission form. This will be the second step used for encryption.
File after ASCII conversion process if given as under

"12343745645646726872893478734687365873 4"
Next section gives the characteristics and metric possessed by various space conservation mechanisms along with encryption strategies. In addition the comparative study also disclose which strategy is most suitable for deduplication mechanism.

2. COMPARATIVE STUDY OF SPACE CONSERVATION AND ENCRYPTION STRATEGIES

The Comparative study performs the comparison of 3 distinct mechanisms used for space conservation. The space conservation mechanism employed is critical since overall cost is dependent upon this phase within deduplication.

Table 2: Parametric Comparison of space conservation mechanisms

Metric	Message level Space conservation	Byte level space conservation	Bit level deduplication
Cost	Message level conservation in cost	Cost conservation is achieved but is expensive as compared to message level deduplication.	Cost conservation is good but is on the higher side as compared to other approaches
Space	Space is highly conserved	Space is conserved but words can be redundant depending upon context	Space conservation is high
Indexing	No indexing mechanism is employed	No indexing mechanism is employed	Indexing mechanism is employed
Execution time	Execution time is high in this case	Execution time is low as compared to message level space conservation	Execution time is least in this case
Compression	High compression but data is lost	Least compression and data is not lost	High Compression and data is not lost
Redundancy	Low	Medium	Low

Parametric comparison of existing literature indicates that best possible approach that could be used in future for enhancement is bit level space conservation mechanism. Although cost is almost similar in message and bit level space conservation but due to indexing mechanism of bit level space conservation could be considered for future enhancements.
The comparison of encryption mechanism used in this literature is given in table 3.

Table 3: Comparison of Encryption standard examined in existing literature.

Metric	RSA	DES	Bit level Encryption
Overhead	Overhead is high	Overhead is medium and is better than RSA	Overhead is high since index table has to be maintained
Space	Space is highly conserved	Space is conserved but words can be redundant depending upon context	Space conservation is high
Indexing	No indexing mechanism is employed	No indexing mechanism is employed	Indexing mechanism is employed
Execution time	Execution time is high in this case	Execution time is low as compared to message level space conservation	Execution time is least in this case
Random	Random at initialization	Random at initialization	Random and static
Key Type	Public	Public and Private	Public and Private
File Upload Redundancy Check	Not checked during upload	Not checked during upload process	Not checked during upload process

From the comparative evaluation it is clear that bit level encryption mechanism is best in its class and can be used for future enhancement. The bit level encryption mechanism can be merged along with bit level space conservation to achieve better performance in terms of overhead and cost.

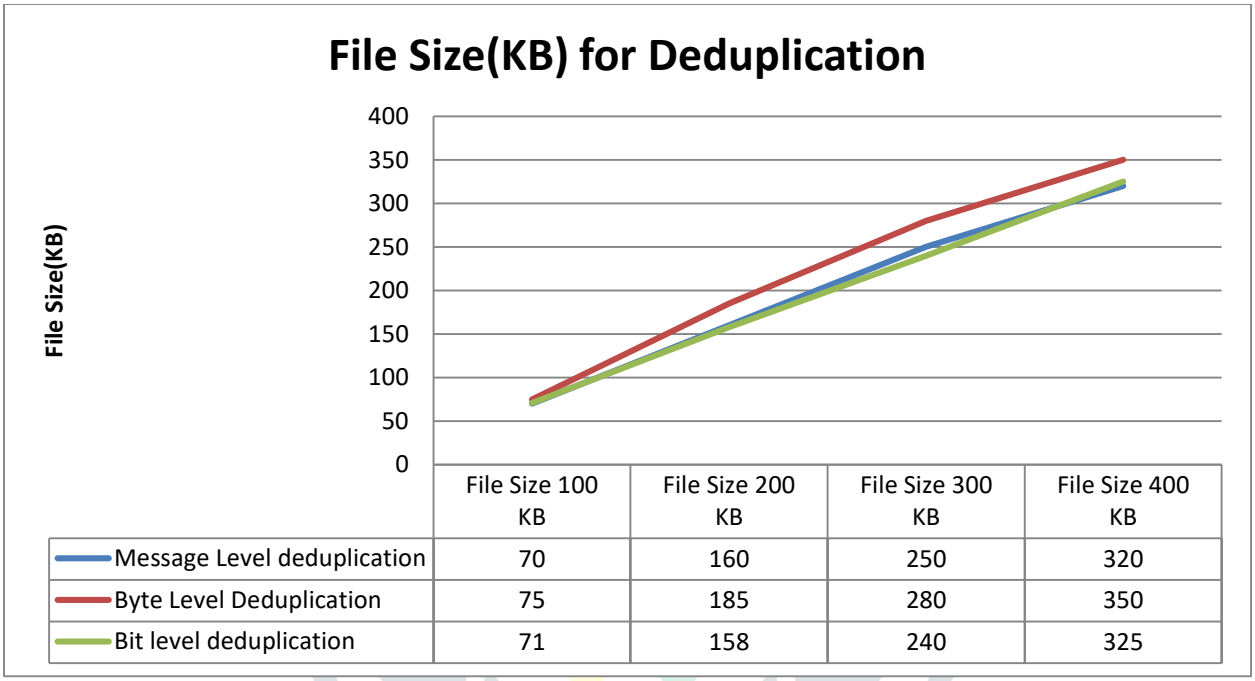
3. PROBLEM DEFINATION

The main problem discovered from the comparative analysis is less use of random number generator and use of static indexing mechanism. The indexing mechanism reduces the size of the file but it is not collaborated with the encryption strategy. In addition, compression ratio is high but data is lost. In other words during decryption procedure, it may not be possible to obtain the same cipher-text again from the plain text. During the file upload operation, same file can be uploaded again and again on the cloud causing space to be used in excess. The key parameters that could be enhanced by collaborating bit level encryption with space conservation procedure includes cost, execution time and file size.

Next section gives result of simulation done within NETBEANS with cloudsim for proving that bit level deduplication is best among all the available simulation and can be used for better deduplication in cloud security systems.

4. PERFORMANCE ANALYSIS AND RESULTS

This section presents comparative study of three best existing deduplication procedures using message level, byte level and bit level deduplication procedures [1][3][7]. The result in terms of file size being uploaded and final file being transmitted is given in figure 1.



	File Size 100 KB	File Size 200 KB	File Size 300 KB	File Size 400 KB
Message Level deduplication	70	160	250	320
Byte Level Deduplication	75	185	280	350
Bit level deduplication	71	158	240	325

Figure 1: File size after deduplication

The size conservation is achieved in all the three cases but message level encryption is better as compared to all other deduplication procedure. The problem however is loss of data and hence bit level deduplication could be considered better as compared to other strategies.

The Execution time is the next parameter that is observed to be better for bit level deduplication. The index is maintained that takes extra space but space complexity leads to reduced execution time since both the parameters are inversely proportional to each other. The execution time comparison is given in figure 2.

Figure 2: Execution time comparison of message, Byte and bit level deduplication

The execution time of bit level deduplication is close to 2.5 ms but rest of the technique execution time is on the higher side. The comparison of reliability in terms of number cloudlet processing is given in figure 3.

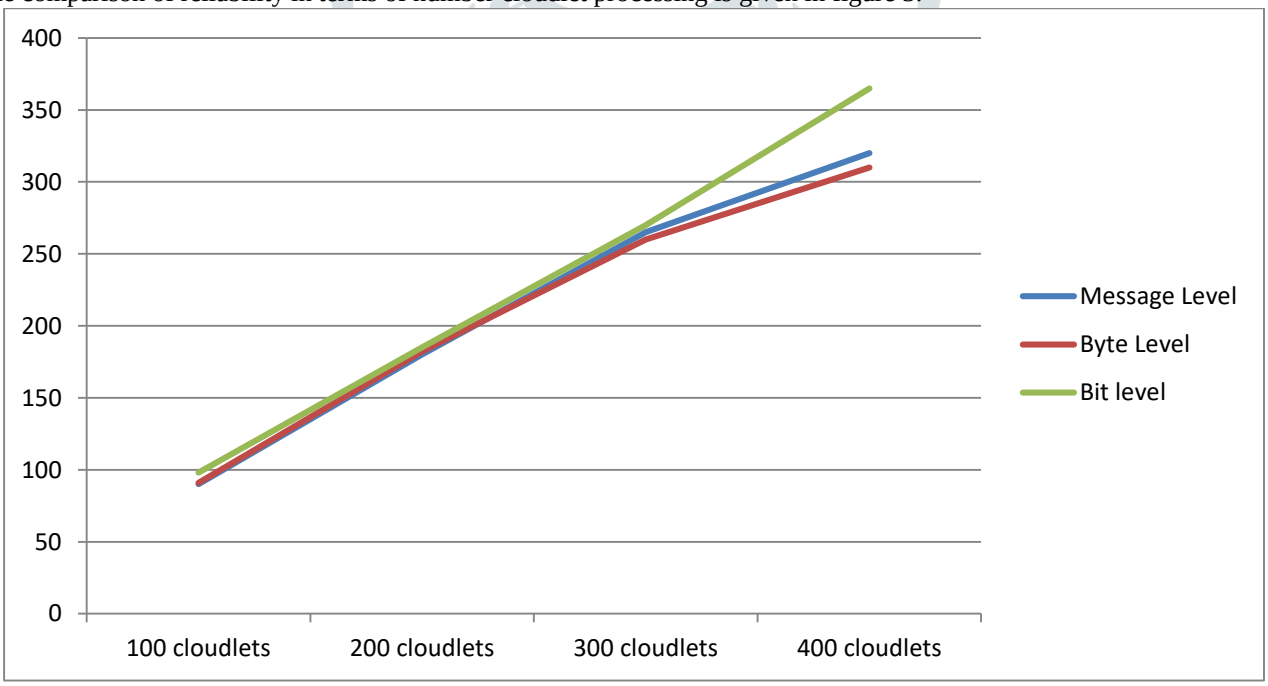


Figure 3: Reliability of cloudlet execution for message, byte and bit level deduplication

The reliability of bit level deduplication procedure is observed to be around 90%. This mechanism is better in all aspects and hence can be used for deduplication for cloud security and space conservation.

5. CONCLUSION

This literature presents the comparative study of existing deduplication strategies including message, byte and bit level deduplication procedures. The message level deduplication although provides highest space conservation but data loss is present. Byte level deduplication procedure provides compression but is static and hence same file can be uploaded on cloud again and again causing cost factor to increase significantly. Bit level deduplication on the other provides dynamic encryption procedure that does not allow the same file to be uploaded on cloud and thus provides space conservation along with reduction in cost.

Message deduplication and bit level deduplication are considered for best for cloud usage but due to high data loss, bit level is considered to be better in terms of reliability, space and execution time.

REFERENCES

[1] Chen, R. Mu, Y. Yang, G. and Guo, F. December 2015. "BL-MLE: Block-Level Message-Locked Encryption for Secure Large File Deduplication," IEEE Trans. Inf. Forensics Secur., 10(12) :2643–2652.

[2] Gupta, S. Jain, S. and Agarwal, M. 2018. "Ensuring Data Security in Databases Using Format Preserving Encryption," Proc. 8th Int. Conf. Conflu. 2018 Cloud Comput. Data Sci. Eng. Conflu. 2018, : 214–218.

[3] Hammami, H. Brahmi, H. and Brahmi, I. 2017. "A Security Approach for Data Migration in Cloud Computing Based on Human Genetics," IEEE Access,: 384–396.

[4] Hua, Y. Member, S. Liu, X. and Feng, D. 2016. "Cost-Efficient Remote Backup Services for Enterprise Clouds," IEEE Access, 3203(c): 1–8.

[5] Nakano, K. Kawakami, K. and Shigemoto, K. February 2009. "RSA encryption and decryption using the redundant number system on the FPGA," IPDPS 2009 - Proc. 2009 IEEE Int. Parallel Distrib. Process. Symp.

[6] Pandey, P. Dhasal, P. and Pandit, R. 2015. "Implementation of RSA RC5 Algorithm in Cloud," Int. J. Comput. Sci. Inf. Technol., 6(1) : 224–227.

[7] Variyar, V. V. S. Haridas, N. Aswathy, C. and Soman, K. P. 2016. "STUDY OF CHUNKING ALGORITHM:BIT LEVEL DEDUPLICATION," Adv. Intell. Syst. Comput., 397: 909–917.

[8] Yan, Z. Zhang, L. Ding, W. and Zheng,Q. 2017. "Heterogeneous Data Storage Management with Deduplication in Cloud Computing," IEEE Trans. Big Data, (2): 1-1.

[9] Zhou, F. Y. A. Wang, S. Zheng, Z. Hsu, C. Lyu, M. 2014. "On cloud service reliability enhancement with optimal resource usage," IEEE Trans. Cloud Comput., 4(4) : 452-466.