Indian Journal of Science and Technology, *Vol 8(24), DOI: 10.17485/ijst/2015/v8i24/80032, September 2015*

ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Confidentiality Technique to Enhance Security of Data in Public Cloud Storage using Data Obfuscation

S. Monikandan[1]* and L. Arockiam[2]

[1]Department of MCA, Christhu Raj College, Panjappur, Tiruchirappalli – 620012, Tamil Nadu, India; moni.tamil@gmail.com

[2]Departmentof Computer Science, St. Joseph's College (Autonomous), Tiruchirappalli – 620102, Tamil Nadu, India; larockiam@yahoo.co.in

Abstract

Security of data stored in the cloud is most important challenge in public cloud environment. Users are outsourced their data to cloud for efficient, reliable and flexible storage. Due the security issues, data are disclosed by Cloud Service Providers (CSP) and others users of cloud. To protect the data from unauthorized disclosure, this paper proposes a confidentiality technique as a Security Service Algorithm (SSA), named MONcryptto secure the data in cloud storage. This proposed confidentiality technique is based on data obfuscation technique. The MONcrypt SSA is availed from Security as a Service (SEaaS). Users could use this security service from SEaaS to secure their data at any time. Simulations were conducted in cloud environment (Amazon EC2) for analysis the security of proposed MONcrypt SSA. A security analysis tool is used for measure the security of proposed and existing obfuscation techniques. MONcrypt compares with existing obfuscation techniques like Base32, Base64 and Hexadecimal Encoding. The proposed technique provides better performance and good security when compared with existing obfuscation techniques. Unlike the existing technique, MONcrypt reduces the size of data being uploaded to the cloud storage.

Keywords: Cloud Service Providers, Cloud Storage, Confidentiality, Obfuscation, Security, Security Service Algorithm

1. Introduction

Cloud computing is an internet based computing. It is evolved from grid computing, utility computing, parallel computing, distributed computing and virtualization[1]. It has more powerful computing infrastructure with a pool of thousands of computers and servers[2]. It provides computational resources like server, storage, software, memory, network etc., as on-demand services[3]. It helps to reduce the computational infrastructure investment and maintenance cost of IT requisite for Small and Medium scale Enterprises (SME)[4]. It provides Everything (X) as a Service (XaaS) where 'X' denotes software, OS, server, hardware, storage, etc[5]. Cloud services are scaling up and down based on the users' demand[6]. Cloud has multiple datacentres placed in different geographical locations in the world to provide reliable services to the users[7]. It provides unlimited service provisioning without any human intervention. Cloud automates the service provisioning by way of running a number of Application Programming Interface (API) in the cloud storage environment.

Data protection is a crucial security issue for most of the enterprises[8]. The main issue focused in cloud computing is data security. However, users are more concerned about the security ofthe data in the cloud. Obfuscation is a process of converting original data into unintelligible data. It is similar to encryption but it uses mathematical calculations or programming logics[9]. Encryption is the procedure of transforming the readable data into unreadable data using an algorithm and a key[10]. The major difference between encryption and obfuscation is that encrypted data cannot be processed until theyare decrypted, whereas obfuscated data can be processed without de-obfuscation. This technique has recently become popular for data storage security in cloud storage[11].

Author for correspondence

Data storage and monitoring are the most important tasks for all enterprises. Small and Medium scale Enterprises (SME) are not having enough infrastructure to do so. Cloud computing solves this problem by providing enormous virtual storage[12]. SMEs outsource the data to cloud storage. The basic definition of outsourcing is to move the in-house activities to a third party (cloud provider) who has the required competence to perform data maintenance activities in an effective and efficient way. Other reasons for outsourcing data are to gain access to the knowledge and to minimize the risks associated. This phenomenon has helped SMEs concentrate more on its core competencies and thereby to gain competitive advantage in the markets.

Data outsourcing creates many security issues on the data stored in the cloud. Researchers start using obfuscation technique to provide security to the data in cloud. To maintaining security for the outsourced data in cloud is an imperative task. To enhance the security of data,the MONcrypt SSA is proposed for securing numerical data. As a part this research, there is an SSA called AROcrypt[13] which is used for Non-numerical data. Users can use these two SSA based on their sensitive data. MONcrypt is used for Numerical data and AROcrypt is used for Non-numerical. This paper only discusses about the MONcrypt SSA.

2. Related Work

There are many more obfuscation techniques available in the literature. The approach for each of the obfuscation techniques can range from very simple to highly mathematical. Recently, researchers have started to use obfuscation techniques for security protection in cloud environment.

Siani Pearson et al.[14] described a privacy manager for cloud computing, which reduces the risk to the cloud computing user of their private data being stolen or misused. As a first line of defense, the privacy manager uses a feature called obfuscation. The idea is that instead of being present unencrypted data in the cloud, the users' sensitive data are sent to the cloud in an obfuscated form. The obfuscation method uses a key which is chosen by the users and known by the privacy manager, but which is not communicated to the CSPs. Thus the CSP is not able to de-obfuscate the users' data and the data are not present on the CSPs' machines. It reduces the risks of theft of the data from the cloud and unauthorized uses of the data. Moreover, the obfuscated data are not personally identifiable information, and so the CSPs are not subject to the legal restrictions that apply to the processing of the de-obfuscated data. However, it is not practical for all cloud applications to work with obfuscated data.

Maheshwari et al.[15] introduced a scheme which allows a user to store data in a cloud and perform database style query on the stored data without using standard cryptography schemes while maintaining data confidentiality. Data obfuscation is used for security which makes it hard for the cloud to reconstruct the plaintext. This scheme represents each character by its glyph image. Instead of storing the normal glyph in a given font, which can be readily understood by a machine, add noise and split the glyph image into small portions. Different portions belonging to a glyph image are stored in servers belonging to different cloud providers. To reconstruct the data, the glyph portions from different servers need to be gathered. The overall concept of this scheme is to convert the users' data into an image and each part of the image is sent to different independent clouds.

The proposed MONcrypt SSA in SEaaS is an obfuscation technique and it is compared with existing obfuscation techniques such as Base64, Base32 and Hexadecimal Encoding with respect to security, time.

3. Existing Data Obfuscation Techniques

Data obfuscation is a method of data hiding where data is purposely scrambled to prevent from unauthorized access[14]. The result of obfuscation is an unintelligible or confusing data. Data obfuscation techniques are used to prevent the intrusion of sensitive data stored in the cloud. However, issues have stemmed from an inability to vigorously prevent security attacks[16]. Several data obfuscation techniques are discussed in the following subsections[17].

3.1 Base64 Encoding

Base64 encoding schemes are commonly used when there is a need to encode binary data that need to be stored and transferred over media that are designed to deal with textual data. This is to ensure that the data remain intact without modification during transport. The Base64 encodesa word "Man" with "TWFu". The Base64 encodes the word "Man", by determining the related ASCII 8-bit values for each letter.

M = 77 a = 97 n = 110

Those values can then be represented as 8-bit binary

M = 01001101 a = 01100001 n = 01101110

Since base64 breaks binary into 6-bit groups, the easiest way is to connect them linearly and then redistribute into 6-bit groups.

010011010110000101101110
is thus represented as:
010011 010110 000101 101110

Each of these values is then recalculated in decimal as, 010011 = (0 x 2^5) + (1 x 2^4) + (0 x 2^3) + (0 x 2^2) + (1 x 2^1) + (1 x 2^0) = 19, then the alphabet corresponding to 19 is taken from the base64 index[18] tableto produce T. In the same way remaining decimals are calculated to produce WFu.

3.2 Base32 Encoding

The Base32 encoding is designed to represent arbitrary sequences of octets in a form that needs to be case insensitive but that need not to be human readable. A 33-character subset of US-ASCII is used, enabling 5-bits to be represented per printable character.

The encoding process represents 40-bit groups of input bits as output strings of 8 encoded characters. Proceeding from left to right, a 40-bit input group is formed by concatenating five 8-bit input groups. These 40-bits are then treated as 8 concatenated 5-bit groups, each of which is translated into a single character in the Base32 alphabet. When a bit stream is encoded via the Base32 encoding, the bit stream must be presumed to be ordered with the most-significant-bit first. That is, the first bit in the stream is the high-order bit in the first 8-bit; the eighth bit will be the low-order bit in the first 8-bit, and so on.

Each 5-bit group is used as an index into an array of 32 printable characters. The character referenced by the index is placed in the output string. These characters, identified from the Base32 index table in[19] are selected from US-ASCII digits and uppercase letters.

3.3 ASCII Encoding

ASCII is a character encoding scheme originally based on the English alphabet. ASCII encoding is a method of representing characters with Base2 (binary) strings. These strings can then be further converted to other formats as needed like Hexadecimal, Base64, etc.

3.4 Hexadecimal Encoding

Hexadecimal is a positional notation system with a base of 16. It uses sixteen distinct symbols. Most often the symbols 0–9 represent the values zero to nine, and A, B, C, D, E, F represent values ten to fifteen. For example, the number 2AF3 is equal, in decimal, to (2 x 16^3) + (10 x 16^2) + (15 x 16^1) + (3 x 16^0), or 10995. Each hexadecimal digit represents four binary digits (bits), and the primary use of hexadecimal notation is a human friendly representation of binary coded values in computing.

Recently, data obfuscation is focused for the security of data in the cloud storage. Normally, obfuscation technique is not using keys to obfuscate the data unlike encryption. Simple obfuscation technique without using key is easy to break by brute force attack. The proposed MONcrypt SSA is using a key to obfuscate the data in cloud. The key is kept by the user to de-obfuscate the data.

4. Motivation

Cloud storage is widely popular and it is used by millions of people in the world. The users are eager to adopt the cloud storage by outsourcing their IT requisites. Cloud computing has numerous advantages such as easy to use and maintain, low power consumption for operation and reductions in the overhead for storing the data. Despite several advantages, cloud also suffers from different security threats and risks. Protecting from security threats and attacks is the primary concern for the enhancement of a more secured cloud infrastructure. Motivated by this fact, this paper aims at enhancing the security of the outsourced data by achieving the following goals.

- To ensure that the data stored in the cloud are accessed only by the data owners.
- To propose security service algorithms suitable for cloud environment using data obfuscation technique.
- To reduce the size of data being stored in the cloud storage.

5. Problem Definition

Data security is a critical area in cloud computing environment. Cloud has no limits, and the data can be physically placed at any data centers which are geographically distributed. The users are forced to use the platform and infrastructure provided by the same CSP. Hence the

CSP knows where the data are located and have full access to the data. This scenario of cloud raises several issues regarding confidentiality of data.

Users' data sent to the cloud are controlled and monitored by CSPs. CSPs as privileged administrators have the rights to look into the users' data. So, there is a possibility that data are hacked from CSPs. Users do not have any control over the data in cloud storage. Moreover, cloud is a public environment. Hence, data may have the chance to be mingled with data of other users.

Users do not know whether the data are encrypted in the cloud storage or not. Maintaining keys for each user is more difficult for CSPs, and the same key is used for all users' data[20]. Users' data have to be in a fixed format specified by the CSP, and hence the CSP knows all the information required for understanding users' data. Here are the issue raised up for data protection.

6. Objective

The objective of this paper is as follows:

- To propose a security service algorithm to enhancing the security of numerical data in cloud storage.
- To reduce the size of users' data are uploaded to cloud.

7. Methodology

Cloud storage has huge amount of virtual storage space. It is used by millions of users every day. Outsourced data in cloud storage needs to be protected from unauthorized access. MONcrypt is one of the SSAs provided by SEaaS. Figure 1 shows the methodology diagram of MONcrypt SSA. There are three cloud services used for Security, Key Generation and Management and Cloud Storage. These services are provided by three different independent CSPs.

The data which are to be outsourced need to be obfuscated before they are uploaded to the cloud storage. It is executed when users choose this SSA for data security. It uses mathematical functions to obfuscate the data such as pow(), reverse(), module() and ascii(). Key denotes the number of rotation during obfuscation process. Key used for MONcrypt is generated from Key Generation and Management as a Service (KGMaaS). Users use the keys and MONcrypt SSA to obfuscate the data. The keys used for obfuscation are kept by the users. It is not communicated to the CSP. MONcrypt SSA is used to obfuscate numerical data only. If users' data contains numerical and non-numerical values, then MONcrypt analyses data for numerical values. Non-numerical values in the users' data are not considered for obfuscation by the MONcrypt SSA. It not only obfuscates the data but also reduces the size of data being uploaded to the cloud storage. There are no existing techniques to reduce the size of the data after the obfuscation process.

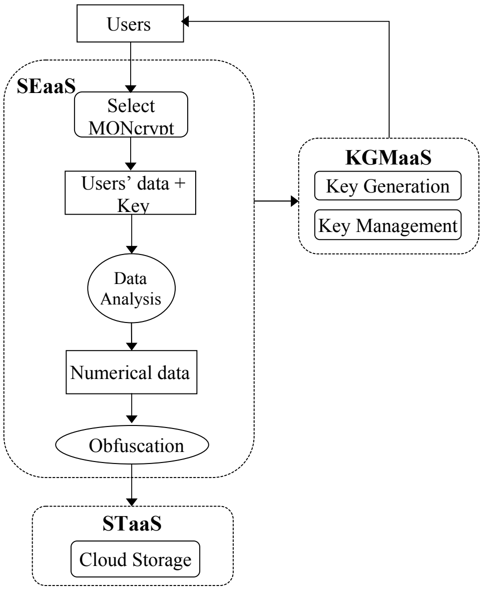

Figure 1. Methodological diagram of MONcrypt SSA.

8. Proposed Technique

This section provides the detailed description of proposed obfuscation technique called MONcrypt SSA.Data obfuscation is recently popular in the field of security in the cloud. The proposed technique is used for the data protection without loss of information content. Steps involved in the proposed MONcrypt SSA are as follows:

- Users submit the PlainText (PT) and key (K) to MONcrypt SSA.
- Determine the numerical values in the PT.
  $\text{MONcrypt} = O_K^{(Num)}$, (O denotes obfuscation)
- Find the number of values in the N=sizeof(PT)
- Calculate the square (ST) for each value in the PT, ST=square (PT).
- Rotate ST at K number of times, K is incremented by 1 for each consecutive value in the PT.
  $\text{Rotation\_ST (RT)} = R_{K+j}^{(ST)} j = 0,1,2,\ldots<N$
  (R denotes Rotation)

- Calculate module (MT) for RT by 256, MT = RT%256.
- Convert the MT into ASCII code to produce ciphertext (CT).

MONcrypt SSA uses mathematical functions pow(), rotate(), module() and ascii(). Thekey Kis used for rotate() function. Users choose MONcrypt to obfuscate the numerical value in their data. SEaaS instructs KGMaaS to generate a key for MONcrypt. KGMaaS generates a key Kand forwards it to the users. Key K is an integer value. It is used at the time of rotating the digits. Rotation is done by K+j number of times. Pseudo code of MONcrypt SSA is given below.

Pseudo Code for MONcrypt SSA

```
MONcrypt_digits(PT)
start
PT ← plaintext
N ← sizeof(PT)
for i ← 1 to N
    ST(i) ← pow(PT(i),2) i=0,1,2…<N
    RT(i) ← rotate(ST(i),K+j) j=0,1,2…<N
    MT(i) ← RT(i)%256
    count(i) ← MT(i)/256
    CT(i) ← ascii(MT(i))
endfor
CT ← cipher Text
end
```

Initially, the Plain Text values are arranged as an array of numerical values PT=PT(1), PT(2), PT(3),…PT(N). The mathematical function pow() is applied on the PT. The pow() function is used to find square ST value of PT. ST is rotated (RT) at K+j, j=1 to N number of times. The key K is incremented by one for each consecutive digit in the ST. RTvalue is divided by 256 to find the count value and also to get the reminder value in MT. The count represents the number of times the RT is divided by 256. Finally, the obfuscated text is attainedby converting the MT into ASCII character code.

The key Kis kept by the users. This value is used for de-obfuscation to get the original PT. In the proposed MONcrypt SSA, each PT value is obfuscated as an 8-bit character value. MONcrypt SSA compresses the size of plaintext.

8.1 Obfuscation Procedure of MONcrypt SSA

The detailed description of procedure involved in obfuscation process of the MONcrypt SSA to secure the data is given below.

Consider the following marks of a student in five subjects for obfuscation,

PlainText(PT)= **66 78 81 66 76**

1. Count the number of values in PT is N
2. The PT values arranged in an array PT(N),

PT(i)	Value
PT(0)	66
PT(1)	78
PT(2)	81
PT(3)	66
PT(4)	76

3. Calculate square (ST) for each value in the PT(i), ST(i) ← pow(PT(i),2)

ST(i)	Value
ST(0)	4356
ST(1)	6084
ST(2)	6561
ST(3)	4356
ST(4)	5776

4. Submit key value K to MONcrypt. Key K ← 3 generated from KGMaaS.
5. Rotate (Rotate from last value, last to first) the ST(i) at key K number of times, K value is incremented by 1 for each consecutive values in ST(i). K+j, j←0,1,2…<N

ST(i)	Value	Key
ST(0)	4356	K←3
ST(1)	6084	K←4
ST(2)	6561	K←5
ST(3)	4356	K←6
ST(4)	5776	K←7

Rotation ST(i) is, RT(i) ← rotate(ST(i), K)

RT(i)	Value
RT(0)	3564
RT(1)	6084
RT(2)	1656
RT(3)	5643
RT(4)	7765

6. Calculate module (MT) for RT(i) by 256, MT(i)←RT(i)%256.

MT(i)	Value
MT(0)	236
MT(1)	196
MT(2)	120
MT(3)	11
MT(4)	85

7. Convert the *MT(i)* into ASCII code to produce ciphertext*(CT)*. CT(i)←ascii(MT(i))

CT(i)	Value
CT(0)	ý
CT(1)	
CT(2)	x
CT(3)	\v
CT(4)	U

8. Ciphertext produced from the obfuscation of student's marks is,

   Ciphertext/Obfuscated text = **ýx\vU**

MONcrypt SSA producing different ciphertext for same PT value which is occurred more than oncein the plaintext. In the above plaintext, value '66' appears two times at 2nd and 4th place. Result of ciphertext for '66' at 2nd and 4th place is 'y' and '\v' respectively. This confuses the cryptanalysis to hack the data using dictionary or brute force attack.

MONcrypt also reduces the size of the ciphertext produced from obfuscation. The size of the plaintext is 14 bytes and after the MONcrypt obfuscation, the size of the cipher text is 10 bytes. MONcrypt SSA converts the plaintexts into ASCII character codes. It reduces the size of data uploaded to the cloud. Existing obfuscation techniques like Base64, Base32 and Hexadecimal Encoding are not reducing the size of data after the obfuscation.

9. Simulation Results

Simulation is performed in the cloud environment. The cloud users' machine connected to the cloud server has the configuration of Windows Operating System with core i3 Intel processor and 4GB RAM. The users' data are obfuscated before they are uploaded and are de-obfuscated when retrieved from the cloud. Thus, the obfuscation is done in the users' machine connected to the cloud server. Time taken for the execution of obfuscation is calculated in the users' machine.

Cloud server is used for cloud storage. Key generation and SSAs are developed as web service and hosted in the cloud server. These services are used for securing the data in cloud storage. The cloud server, Amazon micro instance has the following configuration as Microsoft Windows Server 2008 Base 32-bit operating system, 2.5 GHz Intel Xeon processor, 613 MB RAM, 30GB of Elastic Block Storage (EBS). The users upload the data via user interface. The data submit to SSA in SEaaS are obfuscated and then they are uploaded to cloud server. Security levels of the proposed SSA and existing techniques are measured in cloud server using a Security analysis tool.

Performance of the proposed MONcrypt SSA is calculated by the time taken for obfuscation and de-obfuscation in the users' machine.

Table 1and Figure 2 present the performance comparison of obfuscation with existing obfuscation techniques such as Base 64, Base 32 and Hexadecimal Encoding. The time taken by the existing and proposed obfuscation technique is calculated for different sizes of plaintext. The results show that the proposed technique has taken lower time duration than existing technique for obfuscating different sizes of plaintext.

Table 2 and Figure 3 present the performance comparison of de-obfuscation with existing techniques based on the time. The time taken by the existing and proposed

Table 1. Performance comparison by obfuscation time

Size (MB)	Obfuscation Techniques			
	Base64	Base32	Hexadecimal Encoding	MONcrypt
	(Milliseconds)			
1	62	47	78	31
2	93	78	109	62
3	140	124	156	93
4	156	141	171	125
5	203	187	218	156
10	374	343	390	328
15	515	499	561	484

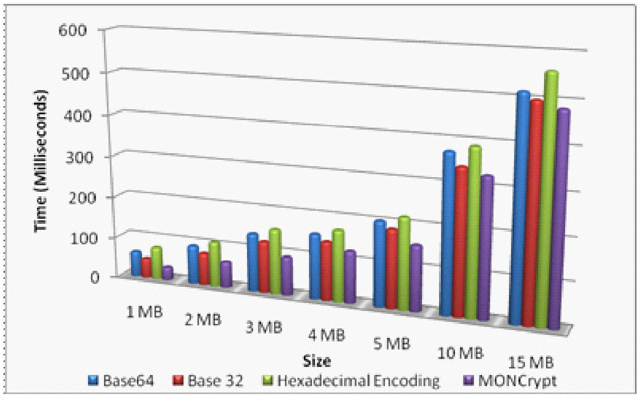

Figure 2. Performance comparison by obfuscation time.

Table 2. Performance comparison by de-obfuscation time

Size (MB)	Obfuscation Techniques			
	Base64	Base32	Hexadecimal Encoding	MONcrypt
	(Milliseconds)			
1	31	93	94	16
2	62	171	125	31
3	78	234	187	47
4	94	312	249	51
5	124	390	327	63
10	250	702	593	109
15	405	1045	876	124

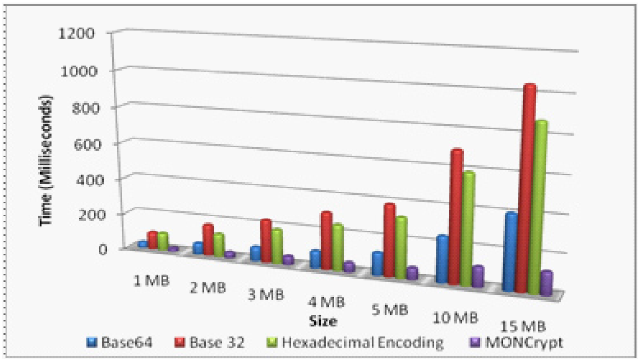

Figure 3. Performance Comparison by De-obfuscation Time.

de-obfuscation technique is calculated for different sizes of plaintext. The result shows that the proposed technique has taken lower time duration than existing technique for de-obfuscation.

Table 3 denotes the size of the plaintext and obfuscated text. Plain_text_size denotes the different size of plaintext before obfuscation. Obfuscated_text_size denotes the different size of the output data generated after the obfuscation.

The executed results show the data size after and before obfuscation. From the results, it is noted that the size of data are reduced after obfuscation. MONcrypt saves lot storage space in the cloud storage. Table 4 shows the comparison of the input and output data size of existing and proposed obfuscation techniques.

In Table 4, no existing technique reduces the size of data after processing. MONcrypt reduces the size of data once the obfuscation is completed.

Table 3. Size of plain text and moncrypt obfuscated text

Plain_Text_Size	Obfuscated_Text_Size
1 MB	160 KB
2 MB	321 KB
3 MB	481 KB
4 MB	642 KB
5 MB	802 KB
10 MB	1.56 MB
15 MB	2.35 MB

Table 4. Size of Input and Output Data Taken for Simulation

Input_ Data Size	Output_Data_Size			
	Base64	Base32	Hexadecimal	MONcrypt
(Mega Bytes)				
1	1	1	1	0.16
2	2	2	2	0.31
3	3	3	3	0.47
4	4	4	4	0.63
5	5	5	5	0.78
10	10	10	10	1.56
15	15	15	15	2.35

9.1 Measure of Security Level of Proposed SSA

Security level is analyzed by using a security analysis tool called All Block Ciphers (ABC) Hackman. The ABC Universal Hackman tool reads the obfuscated data from cloud server and uses dictionary and brute force attack for measuring the security level of each SSA in SEaaS. The Hackman tool analyses the security level of proposed SSA with corresponding existing techniques. This tool is installed in cloud server.

Hackman tool hacks the obfuscated data in Amazon server. It attacks the data in the cloud storage in different ways like dictionary and brute force attack to retrieve the original data from obfuscated data. Once the retrieval is done, Hackman compares the retrieved data with original data to find number of original data retrieved from obfuscated data. Based on the retrieval of original data, the percentage of security level is measured for the proposed SSAs. The procedure for measuring the security level of proposed SSAs is as follows, Let N be total number of cipher text stored in the cloud storage, X denotes the number of original text retrieved from cipher text by ABC Hackman.

To measure the security level of SSA,

$$K = N - X \tag{1}$$

Where, K denotes data not matched with original data.

Therefore, the percentage of security level is measured by,

$$Z = \frac{K}{N} * 100 \tag{2}$$

Where, Z denotes percentage of security level of proposed SSA in SEaaS, and the percentage of insecurity level is measured by,

$$Y = \frac{X}{N} * 100 \tag{3}$$

Where, Y denotes the percentage of insecurity level of the proposed SSA, each SSA produces different security level based on its working procedure.

Table 5and Figure 4 represent the comparison of security level with existing techniques. The results show that the security level of MONcrypt is 90% and security level of existing obfuscation techniques is 64%, 57% and 55% for Base64, Base32 and Hexadecimal Encoding respectively.

Table 5. Comparison of security level of existing and proposed obfuscation techniques

Sl.No.	Obfuscation Technique(s)	Security Level (%)
1.	Base64	64
2.	Base32	57
3.	Hexadecimal Encoding	55
4.	MONcrypt	90

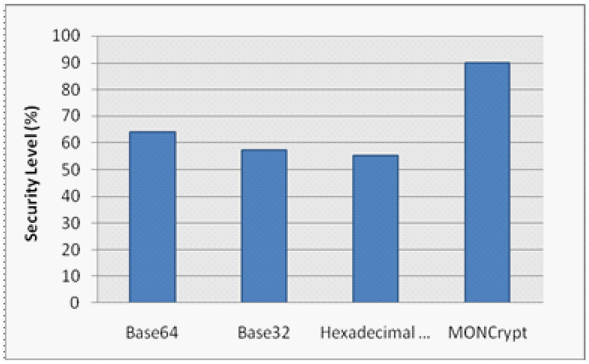

Figure 4. Comparison of security level of existing and proposed obfuscation techniques.

MONcrypt shows maximum level of security when compared with existing obfuscation techniques. It shows the enhancement of security for numerical data stored in the public cloud environment.

10. Result Analysis

The efficiency of MONcrypt SSA is analyzed by considering the space and time complexity. MONcrypt SSA is used for numerical data. It converts the numerical data into ASCII character code. It compresses the total size of data after obfuscation. Table 4 shows the input and output data size of existing and proposed obfuscation techniques. The data presented in the Table 4 shows that there is a linear regression is occurred for space complexity.

The result of MONcrypt SSA shows that it reduces size of input data linearly from 1 to 15 MB. Hence the linear equation for the Space Complexity (SC) is defined as,

$$y = Kx \tag{4}$$

Where x represents plaintext size and y represents ciphertext size and the scalar factor K is represented in Equation (5).

Space complexity of MONcrypt SSA is,

$$K = \frac{y}{x} \tag{5}$$

The average value of K is,

$$\text{SC of MONcrypt SSA} = \frac{\sum_{i=1}^{n} Ki}{n} \tag{6}$$

Where, n denotes number of sample input data used for simulation.

For example, (1) y=0.16 MB, x=1 MB then,

$$K = \frac{0.16}{1} = 0.16$$

(2) y=2.35 MB, x=15 MB then

$$K = \frac{2.35}{15} = 0.16$$

$$SC = \frac{\sum_{i=1}^{7} Ki}{7} = 0.16$$

Space complexity of MONcrypt SSA is 0.16, which is derived from the values in Table 4 MONcrypt SSA requires only 16% of memory for storing the data in cloud. It reduces the memory into 84% than existing obfuscation techniques. Space complexity of existing obfuscation techniques are K= 1. Existing techniques are not reducing the size of data after obfuscation. They did not concentrate on memory usage of the data stored in the cloud.

Time Complexity (TC) is also calculated from the linear equation. Table 1 shows the obfuscation time taken by MONcrypt SSA.

Time Complexity of MONcrypt SSA is defined as y = Kx,

$$K = \frac{y}{x} \tag{7}$$

Where, X denotes size of data, Y denotes time taken for the obfuscation technique, K denotes the time complexity of obfuscation techniques.

For example, (1) x = 1024, y = 31

$$K = \frac{31}{1024} = 0.03$$

(2) x = 15360, y = 484

$$K = \frac{484}{15360} = 0.03$$

$$TC = \frac{\sum_{i=1}^{7} Ki}{7} = 0.03$$

Time Complexity of MONcrypt SSA is 0.03. MONcrypt SSA minimizes the space complexity and it also reduces the time complexity. Time complexity of existing techniques are Base64 is 0.06, Base32 is 0.04 and Hexadecimal encoding is 0.07.Hence, the proposed MONcrypt SSA is more efficient than the existing obfuscation techniques. It is more suitable for public cloud environment.

11. Advantages of MONcrypt SSA

- The main advantage of MONcrypt is to reduce the size of data which are uploaded to the cloud storage.
- It converts the numerical data onto ASCII character code. The existing obfuscation techniques in the literature are not reducing the size of the data.
- Hackers are not able recover the original data.
- MONcrypt takes low time to process the data for obfuscation and de-obfuscation.
- Latency in data upload and download is reduced.
- Low bandwidth usage.

12. Conclusion

Cloud storage is more efficient to store the user's data. It provides cost effective usage to the SMEs. SMEs do not possess full infrastructure to maintain their data with their premises. Data outsourcing helps the SMEs to effectively maintain their data with cloud storage. Data outsourcing has data security issues with cloud storage. The SSA is proposed, namely, MONcrypt. MONcrypt SSAis based on obfuscation technique. According to the proposed technique, the data are obfuscated before they are forwarded to the cloud storage. This technique obfuscates numerical values. MONcrypt SSA obfuscates the plaintext into ASCII character code, so the size of obfuscated text is reduced. The proposed technique not only ensures the confidentiality of the data, but also reduces the size of the plaintext. Existing techniques are not reducing the size of data after obfuscation. The simulation is conducted with different sizes of data and performance is calculated based on the time taken for obfuscation. From the results obtained, it is evident that the proposed MONcrypt provides maximum security to outsourced data in minimum time.

13. References

1. Buyya R, Yeo CS, Venugopal S, Broberg J, Brandic I. Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. Elsevier Science Publishers; 2009; 25(6):599–616.
2. Furht B. Cloud computing fundamentals. Handbook of Cloud Computing. Springer Science, Business Media, LLC.; 2010;1–17.
3. Mell P, Grance T. The NIST definition of cloud computing. Technical Report-800-145. Version 15. National Institute of Standards and Technology. Gaithersburg, MD, United States. 2011.
4. Khajeh-Hosseini A, Sommerville I, Sriram I. Research challenges for enterprise cloud computing. Proceedings of ACM Symposium on Cloud Computing; 2010. p. 1–11.
5. Sun D, Chang G, Sun L, Wang X. Surveying and analyzing security, privacy and trust issues in cloud computing environments. Elsevier Journal of Advanced in Control Engineering and Information Science Procedia Engineering; 2011. p. 2852–6.

6. Buyya R, Yeo CS, Venugopal S. Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities. Proceedings of IEEE International Conference on High Performance Computing and Communications; 2008. p.5–13.
7. Sudha M, Monica M. Enhanced security framework to ensure data security in cloud computing using cryptography. Advances in Computer Science and its Applications. 2012; 1(1):32–7.
8. Chawla R, Nagpal K. Data security issues and requirements in cloud computing. International Journal of Computing Science and Communication Technologies. 2013; 5(2):883–6.
9. Mowbray M, Pearson S, Shen Y. Enhancing privacy in cloud computing via policy-based obfuscation. The Journal of Super Computing. 2012; 61(2):267–91.
10. Yau SS, An HG. Confidentiality protection in cloud computing systems. International Journal of Software Informatics. 2010; 4(4):351-65.
11. Arockiam L, Monikandan S, Sheba PD, Malarchelvi K. Obfuscrypt: A novel confidentiality technique for cloud storage. International Journal of Computer Applications. 2014 Feb; 88(1):17–21.
12. Ristenpart T, Tromer E, Shacham H,Savage S.Hey, you, get off my cloud: Exploring information leakage in third-party compute clouds.Proceedings of ACM Conference on Computer and Communications Security; 2009. p. 199–212.
13. ArockiamL, Monikandan S.AROcrypt: A confidentiality technique for securing enterprise's data in cloud. International Journal of Engineering and Technology. 2015 Feb-Mar; 7(1):245–53.
14. Pearson S, Shen Y,Mowbray M.A privacy manager for cloud computing.Proceedings of International Conference on Cloud Computing; 2009; 5931:90–106.
15. Maheshwari V, Nourian A,Maheswaran M.Character-based search with data confidentiality in the clouds.Proceedings of IEEE International Conference on Cloud Computing Technology and Science; 2012. p. 895–9.
16. Data Obfuscation 2013.Available from: http://www.techopedia.com/definition/25015/ dataobfuscation-do
17. Robertson C. PDF obfuscation - A primer. 2012. Available from: ttps://www.sans.org/reading-room/whitepapers/engineering/pdf-obfuscation-primer-34005
18. Base64 Table. 2013. Available from: http://en.wikipedia.org/wiki/Base64
19. Josefsson. The Base16, Base32, and Base64 data encodings. The Internet Society. 2013 Jan. Available from:http://tools.ietf.org/pdf/rfc4648
20. Mather T, Kumaraswamy S, Latif S. Cloud security and privacy. O'Reilly Media, Inc.; 2009.
21. Arockiam L, Monikandan S. AROMO security framework to enhance security of data in public cloud. International Journal of Applied Engineering Research. 2015; 10(9):6740–6.