Preservation of Data Integrity in Public Cloud Using Enhanced Vigenere Cipher Based Obfuscation

A. K. JAITHUNBI
RMD Engineering College
S. SABENA
Anna University Chennai Regional Office Tirunelveli
L. SAIRAMESH (sairamesh.ist@gmail.com)
Anna University Chennai https://orcid.org/0000-0003-0630-2571

Research Article

Keywords: Obfuscation, Vigenere Cipher, Privacy, Data protection, public cloud.

Posted Date: December 28th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-830050/v1

Abstract

Today's internet world is moves to cloud computing to maintain their public data privately in a secure way. In cloud scenario, many security principles are implemented to maintain the secure transmission of data over the internet. And still, the main concern is about maintaining the integrity of our own data in public cloud. Mostly, research works concentrates on cryptographic techniques for secure sharing of data but there is no such mentioned works are available for data integrity. In this paper, a data masking technique called obfuscation is implemented which is used to protect the data from unwanted modification by data breaching attacks. In this work, enhanced Vigenere encryption is used to perform obfuscation that maintains the privacy of the user's data. Enhanced Vigenere encryption algorithm combined with intelligent rules to maintain the dissimilarity between the data masking for perform encryption with different set of rules. This work mainly concentrates on data privacy with reduced time complexity for encryption and decryption.

1. Introduction

Cloud computing is not an emerging technology as previous decade where now it's a necessary needed technology that everyone knows. As per services offered by the cloud, IaaS,PaaS and SaaS are the well known services, in which the additional service is SecaaS (Security as a Service). In all scenarios where we talk about cloud services that one word come into the mind is security. As per the standard, cloud services are mostly offered by the third parties. When storing the information in third party storage or sharing the information through the third party medium which may raise the question on secure sharing of information. For ensuring the secure sharing, cloud service providers maintain service level agreement to prove their integrity on present level of security. But again, most of CSP's needs the third party auditor to prove the integrity level of their organization security.

Mostly, available security processes are maintaining the privacy of the data by ensuring the usage of advanced cryptographic techniques. But, what kind of privacy provided for the data which is stored in the public cloud. Even though, we can say the data is in public cloud but it belongs to some private concerns. So, the question is data integrity in public cloud for private data. Because, the data stored in the cloud in our specific storage space doesn't need any cryptographic algorithm as assumed. And even if the CSP's provides the algorithms which create, unwanted time consuming for encryption and decryption to access my own data. With that, another point of discussion is that the attacker can easily get the data from cloud if they know the algorithm and key what the user contains.

By comprising all these points, user needed cloud environment should be with the following characteristics. First, need to store the data in secure way that no one can access. Second, if any attacker can access the data, they should not get the actual data. Third, the security provided for the data should maintain the integrity which doesn't allow anyone to modify the content even in cipher text mode. Fourth, user doesn't want to spend more time on encryption and decryption for accessing their own data.

The solution for all these given points moves to the point of obfuscation. The main objective of obfuscation is to prevent the act of reverse engineering process from identifying the kind of cryptographic algorithms we used and protect the data by changing its form to different view.

The article is organized as follows. Section 2 describes the related works for obfuscation and data security. Section 3 provides the proposed Enhanced Vigenere Encryption Algorithm (EVEA) with intelligent for data privacy in public cloud from user side. Section 4 shows the results and performance analysis of the proposed system. Finally, Sect. 5 concludes the article with advantages with proposed system and needed enhancement in the proposed work.

2. Related Works

Some research articles which are describe about the encryption algorithm and cloud security is discussed in this related works. Subhashini and Kavitha (2011) provides a survey on security issues in cloud computing. It clearly shows the need of security in cloud computing mainly in data storage. Because security in cloud is implicitly shows the privacy preservation of data in cloud storage. The cloud users accessing the public cloud storage where the service provider has to provides privacy for user's data.

Philip and Gracia (2006) proposed the modified Vigenere cipher algorithm to reduce the cracking of information encrypted by Vigenere cipher. By adding a few bits of random padding to each byte, one can diffuse the statistical retentiveness found within most messages. The exact quantity of pad will be determined by a one way function in an effort to eliminate the distinguishably of the message bits from the padded random bits. This methodology moderately increases the size of the cipher text, but greatly increases the security of the cipher

Gurpreet singh (2013) proposed the modified Vigenere cipher algorithm for provides the data privacy for user's data by integrating Base 64 and AES. It considers the simple key and converts the alphabets into ASCII value and applying that in the Vigenere table with 92 keys including all special characters.

Govinda and Sathiyamoorthy (2012) describes about the agent based security for cloud computing using the obfuscation technique. In this, various algorithms are proposed for different data types. And they achieved the better results in privacy perspective. Even though, the privacy is achieved but the similar algorithm not able to provide for all data types and it require some additional computational complexity in the obfuscation.

Zhang et al (2012) provides the series of methodologies for privacy protection using different techniques in cloud. In this series, one work describes about noise generation in cloud computing for data protection based on association probability method. Another work by Zhang et al (2012) describes the trust based noise injection strategy for privacy protection. This work calculates the trust value of the service provider and avoiding the unwanted service injection into the user's window. This creates obfuscation by injecting noise service requests to confuse the immoral service requests.

Another recent work by Zhang et al (2015) proposed the time series pattern based obfuscation with probability fluctuation occurrence in the service request. It presents the cluster based technique for analyzing the privacy risk and investigates the corresponding probability fluctuations. All the three works by Zhang et

al explains about the noise generation to confuse the immoral service request given by intruders. Forecasting technique needs the continuous manipulation of data in frequent intervals. It also needs the third party authority support for noise generation.

Arockiam and Monigandan (2014) give the system for maintaining the confidentiality for data security. They give the solution for obfuscation which is similar like encryption algorithm but obfuscation is only applied for numeric values in the data and remaining data is encrypted using the prescribed cryptographic algorithm. Yang et al (2013) proposed the method for obfuscation which encrypt the data with different keys for different users. So, by this each user can access only their allocated information and the other user information should not be viewed using the same key allotted for them. This again requires the additional computational and time complexity for generating different keys for different users. And also, if the attacker can understand the algorithm used for key generation then all user's information is easily decrypted.

Tian et al (2011) provides the base view for personal cloud computing. This paper deals to maintain the private place in public cloud for individual user. This work intends to serve as a

technical reference for the development of security requirements methodologies aiming to the personal cloud.

SaiRamesh et al (2016) gives the method for trusted data sharing between the users from the multiple owners in the same public cloud. This work provides the framework for the multiple data providers sharing their data for multiple users in the protected way. Another recent work by Selvakumar et al (2019) multi-authority access control mechanism for maintains the privacy in the public cloud for user's data. It also maintains the privacy of data from unauthorized access in the public cloud environment.

The above mentioned survey through related articles gives the overview of the importance of data privacy in public cloud. And also, it describes the need for technique like obfuscation to avoid computational complexity by using simple encryption algorithm for data masking. And some of the limitations are overcome and some of them are still unsolved like strong key for encryption with less computational complexity model. This proposed system gives the solution by providing efficient data protection with less computational complexity.

3. Proposed System

This proposed work projects mainly on providing the security for user's information in public cloud with less computational cost. In previous work (Mowbray et al (2012)) specified about the privacy manager for defining the policy specification. Here, the policy specification should be maintained by the clients itself for choosing their encryption algorithm and where to apply that algorithm and all.

The method obfuscation discussed here about encrypting the data from the user side itself and user itself to protect user data from the service provider. For that, Extended Vigenere Cipher is used in this work which provides better encryption mechanism with less computational time. Gurpreet singh and Supriya(2013) proposed the Modified Vigenere Encryption Algorithm (MVEA) which differs from standard Vigenere algorithm by including all characters in the keyboard of the computer system. But the current internet world using the mobile as their system and they want everything should be in the mobile environment. In this scenario, using all special characters as a keyword is again becoming the complex task. To avoid these kind of difficulties, we modified and extended the standard Vigenere Encryption algorithm with only alpha-numeric characters. And also, this system could achieve better avalanche effect than MVEA while using alpha numeric characters (totally 62) instead of 95 characters which is discussed in MVEA.

The proposed EVEA doesn't contain any special characters because we are not going to use this for any transfer of information. The main objective of this work is to enhance the privacy for client's data which is stored in public cloud storage. This information need not going to be shared any other client's. If the need arise then the third party auditor introduce the key management policies with the use of any encryption algorithms.

In the proposed EVEA, CT_i is the cipher text obtained from the given plain text PT_i by using the key text KT_i. Some article discussed about modified Vigenere Cipher by using the ASCII value of the given text. But here, the EVEA doesn't consider the ASCII value. It simply uses the same mathematical expression which is used for standard Vigenere Cipher but with 62 characters. If we go with the 26 characters, then 26 possible Caesar cipher are applied and cipher text is generated.

The Extended Vigenere Encryption algorithm (EVEA) character with its values is shown in Table 1.

Table 1
Enhanced Vigenere Cipher Table

	A	B	C	D	E	.	.	.	W	X	Y	Z	0	1	2	3	4	5	6	7	8	9	a	b	c	d	.	.	.	x	y
A	A	B	C	D	E	.	.	.	W	X	Y	Z	0	1	2	3	4	5	6	7	8	9	a	b	c	d	.	.	.	x	y
B	B	C	D	E	F	.	.	.	X	Y	Z	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	.	.	.	y	z
C	C	D	E	F	G	.	.	.	Y	Z	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f	.	.	.	z	A
D	D	E	F	G	H	.	.	.	Z	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f	g	.	.	.	A	B
E	E	F	G	H	I	.	.	.	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f	g	h	.	.	.	B	C
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
W	W	X	Y	Z	0	.	.	.	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z				T	U
X	X	Y	Z	0	1	.	.	.	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	0				U	V
Y	Y	Z	0	1	2	.	.	.	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	0	1				V	W
Z	Z	0	1	2	3	.	.	.	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	0	1	2				W	X
0	0	1	2	3	4	.	.	.	m	n	o	p	q	r	s	t	u	v	w	x	y	z	0	1	2	3				X	Y
1	1	2	3	4	5	.	.	.	n	o	p	q	r	s	t	u	v	w	x	y	z	0	1	2	3	4				Y	Z
2	2	3	4	5	6	.	.	.	o	p	q	r	s	t	u	v	w	x	y	z	0	1	2	3	4	5				Z	0
3	3	4	5	6	7	.	.	.	p	q	r	s	t	u	v	w	x	y	z	0	1	2	3	4	5	6				0	1
4	4	5	6	7	8	.	.	.	q	r	s	t	u	v	w	x	y	z	0	1	2	3	4	5	6	7				1	2
5	5	6	7	8	9	.	.	.	r	s	t	u	v	w	x	y	z	0	1	2	3	4	5	6	7	8				2	3
6	6	7	8	9	a	.	.	.	s	t	u	v	w	x	y	z	0	1	2	3	4	5	6	7	8	9				3	4
7	7	8	9	a	b	.	.	.	t	u	v	w	x	y	z	0	1	2	3	4	5	6	7	8	9	A				4	5
8	8	9	a	b	c				u	v	w	x	y	z	0	1	2	3	4	5	6	7	8	9	A	B				5	6
9	9	a	b	c	d				v	w	x	y	z	0	1	2	3	4	5	6	7	8	9	A	B	C				6	7
a	a	b	c	d	e				w	x	y	z	0	1	2	3	4	5	6	7	8	9	A	B	C	D				7	8
b	b	c	d	e	f				x	y	z	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E				8	9
c	c	d	e	f	g				y	z	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F				9	a
d	d	e	f	g	h				z	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	G				a	b
w	w	x	y	z	0				I	J	K	L	M	N	O	P	Q	R	S	T	U	V								t	u
x	x	y	z	0	1				J	K	L	M	N	O	P	Q	R	S	T	U	V	W								u	v
y	y	z	0	1	2				K	L	M	N	O	P	Q	R	S	T	U	V	W	X								v	w
z	z	0	1	2	3				L	M	N	O	P	Q	R	S	T	U	V	W	X	Y								w	x

Key selection is the major task in all encryption algorithms. In Vigenere Cipher also the key length should be a maximum and if it's minimum then it should be repeatable until it equals the length of the given plain text. If we make the large text as key then it's should be shared with some other or to be saved in some place. Again, it should be the tedious task from the user side to maintain the key in the secure manner.

In this work, algorithm for key maintenance should also be reduced by using the plain text itself as a key. Instead of repeating the same key multiple times for equals the length of the plain text, we can use the same plaintext as key to encrypt the plain text as cipher text. If the plain text will be the key then we have to make a copy of the whole plain text to recall the key when decryption occurs.

In account of all above mentioned difficulties, the proposed work come with the solution to include language processing technique which used to chunk the whole text into characters. The whole plain text is chunked into separate words and ten words considered as a segment. In a segment, first five words will be encrypted using the next five words. Most of the times, characters may not be equal if we go with the words length. In such scenario, the actual Vigenere Cipher technique of repeatable key is to be followed.

For example, the sentence "Cryptography is the important subject in computer science that everyone needs to study"

By applying the chunking the five words total length is 33 characters and next five words is 29 characters length. In this case, again the key starts from the first character of the key word and it consumes the needed key length from the given words. Another one drawback arises that the next 33 characters in cipher text

is same as plain text. To overcome this, simple keyword to be used to encrypt the plain text as cipher text as it is in Vigenere cipher. So, this proposed works follows obfuscation policy with simple encryption techniques without the involvement of cloud service provider and third party auditor.

3.1Process Flow for Obfuscation Technique

The process flow diagram in Fig. 1 explains the flow of obfuscation technique. Here, the input data is given by the user and the data is stored in an array. Then, encryption and data masking are performed on the data. After that, the data is stored in public cloud storage. Whenever the user wants to access the data, the user retrieves the data from the cloud and decrypts it at the user's system.

3.1.1 Data Chunking

In this module, the plain text is given by the user. This plain text is taken as input for this module. Then, chunking process is applied on the plain text. The process results in chunked data. The chunked data is stored in a two dimensional array. During subsequent processes, this two dimensional array is used for encryption and decryption of the data.

3.1.2 Data Encryption

The data stored in two dimensional arrays is taken as input for this module. The encryption process comprises of two phases. During the first phase, the data in even numbered rows of the two dimensional array is added with the data in odd numbered rows of the two dimensional array. If the number of data in the odd rows is less than the number of data in even rows, then the data in odd row is repeated up to the length of the data in even row. Or else, the data is taken as it is. After that, the data in the even numbered rows of the two dimensional array is encrypted.

During the second phase, the data in the odd numbered rows of the two dimensional array is encrypted using a random key, which is given by the user. The key is repeated for the length of the data in the odd numbered rows of the two dimensional array and added to the data. After this, the data in the odd rows is also encrypted. Then, the encrypted text (i.e., cipher text) is stored in the public cloud.

3.1.3 Authentication

Whenever the user wants to retrieve the data from the cloud storage, the authentication process is performed. Authentication provides security for the data in the cloud storage. It protects data from unwanted use. It also provides integrity for the data stored in public cloud storage. A user can retrieve the data only after passing through this authentication process.

3.1.4 Decryption

The data which is retrieved from the cloud is used as an input in this module. This process is converse to that of the encryption process. This module also constitutes two phases. Firstly, the data in the odd rows of the array is decrypted using the random key, which has been used for encryption. After this process, the data in the odd rows will be decrypted.

Subsequently during the second phase, the data in the even rows are decrypted using the data in the odd rows of the array. If the cardinality of data in odd rows is less than the cardinality of the data in even rows of the two dimensional array, then data in the odd rows are repeated up to the length of the data in the even rows. Or else the data in the odd rows of the array is taken as it is. Then, the data in the even rows of the two dimensional array are decrypted using the odd row data. The decrypted data is the original data that the user wants.

3.2 Extended Vigenere Encryption Algorithm

The extended Vigenere encryption algorithm includes key generation, encryption and decryption with key verification. Intelligent rules are applied to extract the key from the given plain text.

The intelligent rules are written to implement the language processing technique in order to make decision regarding the character length with respect to word count. This procedure is explained in an algorithmic form as follows.

Intelligent Rules for Key Manipulation Algorithm (IRKMA)

Input : File with Plain text PT, Number of words in one segment N

Output : Segmented words S_1, S_2, ...S_l - S(PT)

1. Start
2. Read the Plain text PT and chunk into words (w)
3. Count the word in the given file and represent as Number (w)
4. Get the input N
5. Read the words w_1, w_2, ...w_n based on the given length and segment as S_1, S_2, ...S_l

Pseudocode for Extended Vigenere Encryption Algorithm

Key Generation

Input : Segmented Plain text S(PT), keyword KW

Output : Cipher CT

1. Give Segmented PlainText S(PT)
2. Choose the segment length L
3. Once segment length is chosen, divided into two segments $S_1(PT)$ and $S_2(PT)$
4. Count the words in both segments separately wc_1 and wc_2
5. Check $wc_1 = wc_2$,

5a. If yes choose wc_2 as key for wc_1

5b. Else make wc_2 as length equals wc_1 by repeat the character of wc_2 from initial character and updated wc_2 which chosen as key for wc_1

Encryption :

6. Initialize cipherText to null

7. If $S_1(PT)$ is less than length of $S_2(PT)$ then choose subpart of wc_2 which equals the length of wc_1.

8. Now choose the wc_2 as encryption key KT for wc_1 of plain text PT

9. Apply the Encryption process of Vigenere Cipher Algorithm using equation 1

10. Choose the simple keyword some KL=n

11. Apply simple Vigenere Cipher Algorithm for $S_2(PT)$ using the simple key KW

12. Encrypt $S_2(PT)$ using the equation 1and get CT_{kw}

13. Encrypted text CT_{kw} append with CT_i in the segment of $S_2(PT)$

Decryption :

Input : Cipher Text CT, Keyword KW

Ouptut : Plain Text PT

1. Apply IRKMA to segment the Cipher Text CT

2. Decrypt CT_{kw} by keyword KW using equation 2 to get $S_2(PT)$

3. Check $wc_2 = wc_1(CT_i)$, If equals apply the wc_2 as key to decrypt CT_i by applying equation 2

4. Else make wc_2 as length equals wc_1 by repeat the character of wc_2 from initial character and updated wc_2 which chosen as key for wc_1

5. Apply the decryption expression as given equation 2.

6. Receive the plain text PT

As mentioned in algorithm for key generation, the key is count as the words 6 to 10 of the plain text and total characters are counted for the five words. After, manipulate the key length as equal to the plain text to be encrypted apply the encryption algorithm steps. The key is applied to alternate five words and it also reduces the repeatability in keywords in actual Vigenere Cipher algorithm. The word length may be vary based on the user's perspective. This makes the proposed algorithm to supports dynamic key generation and it will not be same as every time while the encryption carried out for plain text.

The encryption are carried out with the same process as followed in Vigenere Cipher. The mathematical expression for encryption process is given in equation 1

$$CT_i = E_{kt}(PT_i) = (PT_i + KT_i) \bmod 62 \quad (1)$$

The decryption is carried out by the equation 2

$$PT_i = D_{kt}(PT_i) = (CT_i - KT_i) \bmod 62 \quad (2)$$

By using this EVEA, the privacy over the information is preserved with less computation complexity. And also, storage space for key management is also neglected. Section 4 provides the experimental results and analysis the performance of EVEA by comparing with Standard Vigenere Cipher Algorithm (SVCA) and MVEA [1].

4. Results And Performance Analysis

The experiments are performed based on the time taken for encryption and decryption based on the size of the plain text. Table 2 and Fig. 2 shows the time analysis for encryption and decryption for SVCA, MVEA and EVEA by varying the size of the plaintext. In this analysis, EVEA requires less time than other two algorithms for larger plaint text.

Table 2
Time consuming for encryption and decryption

File Size (MB)	Time Taken (ms)							
	Encryption				Decryption			
	AES	DES	MVEA[1]	Proposed EVEA	AES	DES	MVEA[1]	Proposed EVEA
1	1	1.05	1	1	0.90	1.10	0.95	0.90
5	2.1	2.25	2.05	1.95	1.70	2.10	1.55	1.50
10	3.2	3.40	3.10	2.85	2.40	2.90	2.25	2.10
25	4.0	4.95	3.6	3.40	3.50	4.50	3.30	2.90
50	4.8	5.60	4.2	3.60	4.35	5.10	3.90	3.45
100	7.3	9.35	5.9	5.10	6.30	8.25	4.75	4.25

Table 3 shows the need of obfuscation method without intervention of service provider for time consumption and data integrity. It shows the time analysis for storing the data in cloud storage after encryption executed by Third party auditor and our proposed EVEA. The result shows that the proposed EVEA requires less time for storing the data in cloud storage after encryption. In decryption also it processed in less time than execution done in third party auditor. The character length is considered for analysing the performance based on time taken with respect to the key length. Figure 3 shows the pictorial representation of this analysis.

Table 3
Time taken for obfuscation in cloud side by third party auditor and from user level

Key Length (Number of Character)	Time Taken (ms)			
	Key Generation		Obfuscation	
	Using TPA	User Level	Using TPA	User Level
5	2.50	2.3	3.20	1.95
10	3.55	3.25	4.50	2.85
15	4.80	3.85	5.25	3.40
25	6.30	4.50	6.50	3.60
50	9.20	5.20	11.50	5.10
100	12.15	6.05	15.70	5.85

Avalanche Effect

Avalanche Effect refers to a desirable property of cryptographic algorithms where, if an input is changed slightly (for example, flipping a single bit) the output changes significantly (e.g., more than half the output bits flip). This is a desired effect in encryption to ensure that a person cannot easily predict a message based on the changes in the hash value through a statistical analysis as shown in Eq. 3.

$$\text{Avalanche Effect} = \frac{\text{Number of flipped bits in ciphered text}}{\text{Number of bits in ciphered text}} \qquad (3)$$

The input for the encryption is ”THIS IS MY NEW TYPE HELLO WORLD PROGRAM IN THE”. The key for encryption given is ”HELLO”. This results the output as ”ALTD PW XM RPH ACAP OIWWC AZCZK ACCNVLX PR EVL”.

Then the input is changed as ”THIS IS MY NEW TYPE HELLO SUPER PROGRAM IN THE”. The key is same as the first encryption.

This results the output as ”ALTD PW XM RPH ACAP OIWWC WFASY ACCNVLX PR EVL”. The number of bits changed is 5. The total number of bits 37. The avalanche effect is,

$$\text{Avalanche Effect} = \frac{5}{37} \times 100 = 13.5135\%$$

The input for the encryption is ”THIS IS MY NEW TYPE HELLO WORLD PROGRAM IN THE”. The key for encryption given is”HELLO”. This results the output as ”ALTD EG BP VRE MFTX OIWWC DSCWR WVZRFHQ PR ALP”.

Then the input is changed as ”THIS IS MY NEW TYPE HELLO SUPER PROGRAM IN THE”. The key is same as the first encryption. This results the output as ”ALTD AM BP VRE MFTX OIWWC ZYAPF WVZRFHQ PR ALP ”. The number of bits changed is 5. The total number of bits 37. The avalanche effect is,

$$Avalanche\ Effect = \frac{7}{37} \times 100 = 18.9189\%$$

Based on the Avalanche effect, the performance of the proposed system is evaluated and compared with existing MVEA technique.

5. Conclusion

This work concentrated on data protection and privacy preservation of user’s data in public cloud using obfuscation technique. For obfuscation, the Enhanced Vigenere Encryption algorithm is used with Intelligent Rules to generate varied key length. The key-keying technique also applied here by encrypting the key using different key by the user. The main objective is to encrypt the information without any third party intervention. This is achieved in this proposed by carried out cryptographic evaluation in the user side and it also reduces the time taken for encryption and decryption. And also, performance based on time and attacks carried out in the user side is lesser in the proposed system when compared with previous techniques using Avalanche effect. The system may enhanced in future by using ASCII values for alphabets and numeric instead of using the serial number of the character. And, future work can also use fuzzy rules for choosing efficient key by varying the character length.

Declarations

Ethical Responsibilities of Authors

The authors declare that they do not have any conflict of interests. This research does not involve any human or animal participation. All authors have checked and agreed the submission.

References

1. Singh, G. "Modified Vigenere Encryption Algorithm and Its Hybrid Implementation with Base64 and AES." In Advanced Computing, Networking and Security (ADCONS), 2013 2nd International Conference on, pp. 232–237. IEEE(2013).
2. Subashini, S., & Kavitha, V. (2011). A survey on security issues in service delivery models of cloud computing. *Journal of network and computer applications*, 34(no. 1), 1–11
3. Govinda, K., & Sathiyamoorthy, E. (2012). *"Agent based security for cloud computing using obfuscation." Procedia Engineering*, 38, 125–129
4. Zhang, G., Zhang, X., Yang, Y., Liu, C., & Chen, J. "An association probability based noise generation strategy for privacy protection in cloud computing." In International Conference on Service-Oriented Computing, pp. 639–647. Springer, Berlin, Heidelberg(2012).
5. Arockiam, L., & Monikandan, S. "Efficient cloud storage confidentiality to ensure data security." In Computer Communication and Informatics (ICCCI), 2014 International Conference on, pp. 1–5. IEEE(2014).
6. Selvakumar, K., SaiRamesh, L., Sabena, S., & Kannayaram, G. (2019). "CLOUD COMPUTING-TMACS: A Robust and Verifiable Threshold Multi-authority Access Control System in Public Cloud Storage.". *In Smart Intelligent Computing and Applications* (pp. 365–373). Singapore: Springer
7. Yang, P., Gui, X., Tian, F., Yao, J., & Lin, J. "A privacy-preserving data obfuscation scheme used in data statistics and data mining." In High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing (HPCC_EUC), 2013 IEEE 10th International Conference on, pp. 881–887. IEEE(2013).
8. Mowbray, M., Pearson, S., & Shen, Y. (2012). Enhancing privacy in cloud computing via policy-based obfuscation. *The Journal of Supercomputing*, 61(no. 2), 267–291
9. Tian, Y., Song, B., & Eui-Nam, H. "Towards the development of personal cloud computing for mobile thin-clients." In Information Science and Applications (ICISA), 2011 International Conference on, pp. 1–5. IEEE(2011).
10. Zhang, G., Liu, X., & Yang, Y. (2015). Time-series pattern based effective noise generation for privacy protection on cloud. *IEEE Transactions on Computers*, 64(no. 5), 1456–1469
11. Zhang, G., Yang, Y., Yuan, D., & Chen, J. (2012). A trust-based noise injection strategy for privacy protection in cloud. *Software: Practice and Experience*, 42(no. 4), 431–445
12. Wilson, P. I., & Garcia, M. (2006). *"A Modified Version of the vigenere Algorithm." IJCSNS*, 6, 140
13. SaiRamesh, L., Sabena, S., Thangaramya, K., & Kulothungan, K. (2016)."Trusted Multi-Owner Data Sharing Among Dynamic Users in Public Cloud."

Figures

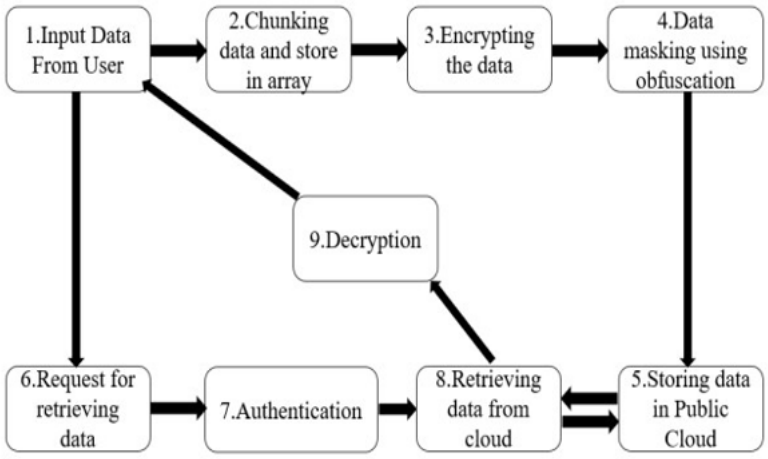

Figure 1

Process Flow for Obfuscation Technique

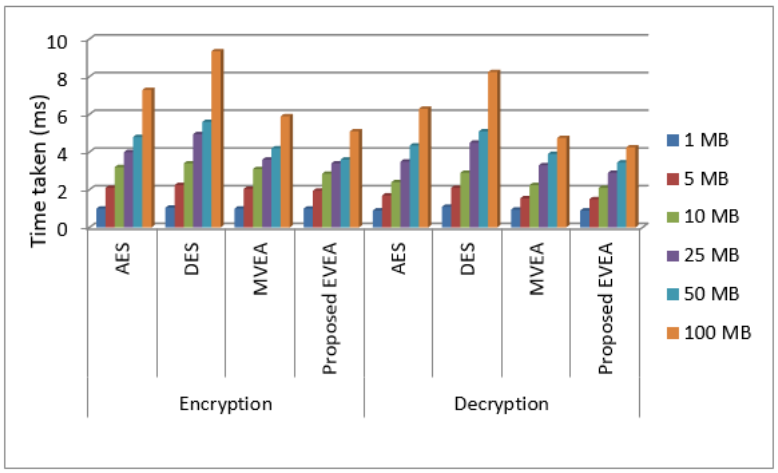

Figure 2

Performance analysis of different Encryption System

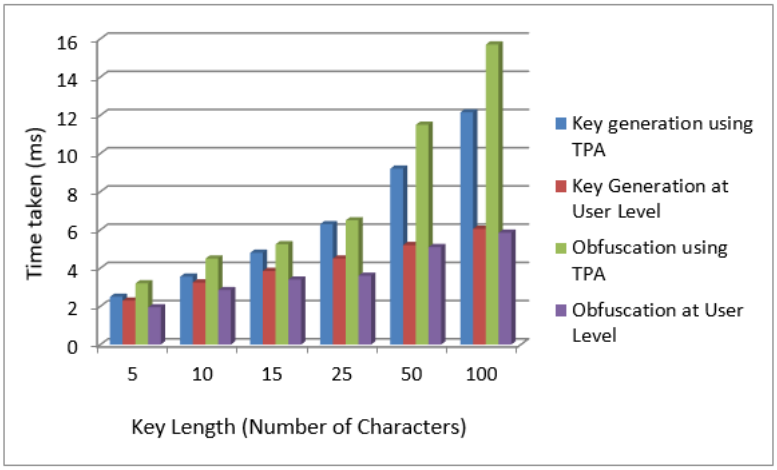

Figure 3

Time taken for obfuscation in cloud side by third party auditor and from user level