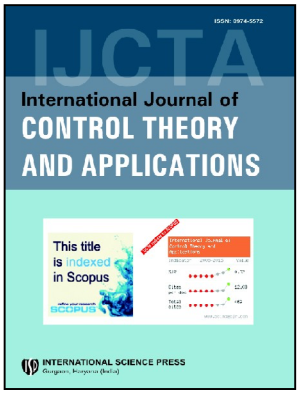


International Journal of Control Theory and Applications

ISSN : 0974-5572

© International Science Press

Volume 10 • Number 29 • 2017

A Secure Confidential Data Storage Using Fast Chaos-based Dna Cryptography (FSB-DC) for Cloud Environment

R. Pragaladan[1*] and S. Sathappan[2]

[1]Ph.D Research Scholar, PG and Research Department of Computer Science, Erode Arts and Science College, Erode, Tamil Nadu, India
[2]Associate Professor, PG and Research Department of Computer Science, Erode Arts and Science College, Erode, Tamil Nadu, India
[1]Corresponding Author, E-mail: pragaladanr@gmail.com*
[2]E-mail: devisathappan@yahoo.co.in

Abstract: Cloud computing is the growing technology in the real world environment which concentrates on providing various types of services to the cloud users as per their requirement on demand basis. There are many cryptographic algorithms has been presented in many existing works which are not suitable for the dynamic cloud platform. In the proposed work, the Fast Chaos-Based DNA Cryptography (FCB-DC) is introduced to support and provide the high level of the security and confidentiality to the dynamic cloud environment. The 3-bit binary encoding system increases the cryptosystem complexity in a considerable way. The FCB-DC mechanism integrated with the pseudo random number which increases the randomness. Thus the possibility of random is avoided. The proposed FCB-DC framework compared with two existing methods called DNA-based Cryptography for the multi-cloud system (DNA-CMCS) and Random Function and Binary Arithmetic Operations (RFBAO). The FCB-DC framework is evaluated using business data transactions in Java code with Cloudsim 3 simulation environment using Amazon EC2 data sets. In the experimental scenario from which it concluded that the FCB-DC structure provides the highly confidential and secure data storage over the cloud environment for the cloud users. The numerical analysis of the proposed methodology compared various parameters like data confidentiality, execution time, communication overhead and space complexity of the existing system where the security level is more in proposed system.

Keywords: fast chaos-based, DNA cryptography – 3 bit – randomness- cloud data storage.

1. INTRODUCTION

Cloud computing has developed significantly in recent years, but at the same time, it includes many security challenges and risks are available. To reduce the possibilities of these risks such as service outage, theft of data, data leakage and the chances of a malicious insider attack, using “single cloud” provider is becoming less favoured. To protect data through the unsecure network of cloud computing environment, using various types of data protection is necessary.

Cryptography is the method of giving security of data transmission via a public network by encrypting the original data or message. The plain text converted to the message which cannot be read by a human. An efficient direction of providing data security can term as DNA-based Cryptography. This paper process the encryption and decryption using DNA sequence with applied to cryptographic algorithms. Also proposed the DNA sequencing method be more secure and reliable for transmitting information effectively through networks.

The Fast Chaos-based DNA Cryptography System (FCB-DC) achieve more security over the cloud environment is concentrated.

2. LITERATURE SURVEY

In cryptography research, the DNA cryptography is a positive direction to improve safety aspects in cloud concept. DNA can use in cryptography for storing and transmitting the data, as well as for computation.

Richa H. Ranalkar *et al.* [1] proposed a DNA-based Cryptography for the multi-cloud system (DNA-CMCS) enable the users to demonstrate powerful security strategy of using DNA cryptography. Siddaramappa V [2] developed Random Function and Binary Arithmetic Operations (RFBAO). In this method used to describe and review the data security by using the random function in DNA sequence.

A study of large-scale data management in cloud environment was provided in Sherif Sakr *et al.* [3]. However, with the increase in scalable cloud users, security remained unsolved. Optimal allocation of sensitive data objects presented in Manghui Tu *et al.* [4] used secret sharing scheme. However, a related problem arises when data that has stored for confidentiality on security.

R. Pragaladan *et al.* [5] discussed a Watson-crick Hoogsteen base Confidential Data Transaction (WHO-CDT) mechanism used in cloud infrastructure. WHO-CDT mechanism employs business transactional information storage in a ultra-compact form using the DNA confidentiality structure.

K. Menaka [6] proposed Message Encryption Using DNA Sequences. She described DNA-based data hiding using DNA sequence. In this work, an algorithm using DNA sequences for data hiding proposed and discussed for secure data transmission and reception. Another method based on the Randomised error correcting DNA codes applied in Dan Tulpan *et al.* [7] method for data confidentiality in cloud infrastructure. However, a trade-off achieved between data privacy and integrity.

A wide variety of security techniques proposed for a cloud service provider. Mohammad A. Alzain *et al.* [8] compares their multi-cloud database model with Amazon cloud. They concluded that data storage and retrieval could be done more efficiently using proposed model. Vincent Gramoli *et al.* [9] demonstrates a comparison of two own multi-tenancy architectures defined at different levels, one at the operating-system kernel level and second at the hypervisor level. Sangdo Lee *et al.* [10] have defined a new stretch called as rain cloud system using libraries to manage different CSPs. According to S. Jaya Prakash *et al.* [11] the service availability risk or loss of data reduced by multiple CSPs with data replication technique.

K. D. Bowers [12] suggested HAIL i.e., a distributed cryptographic system (High-Availability and Integrity Layer) that allows proving client that stored file intact with retrievable using a different set of servers. HAIL handles file integrity and availability across the collection of servers with no cost storage services. Abu-Libdeh [13] suggested RACS that employs RAID5 techniques to implement high-available and storage-efficient data replication on Redundant Array of Cloud Storage system. The RACS system does not solve security problems of cloud storage but instead deals with economic failures and vendor lock-in from DEPSKY model by C. Cachin C. Cachin *et al.* [14].

There are some studies on gaining constancy from untrusted clouds to improves the flexibility of cloud storage, as Mahajan *et al.* [15] believe that cloud storages face many risks. It does not tolerate losses of data and its service availability depends on cloud availability shown by F. Rocha *et al.* [16]. Other works which implement services on top of untrusted clouds are studied in A.J. Feldman *et al.* model called SPORC [17] and A. Shraer *et al.* [18] developed the untrusted verification.

Michael François *et al.* [19] proposed a Chaos-based cryptography widely investigated in the field of random number generators which describes a novel pseudo-random bit generator based on chaotic logistic maps.

In this work, the author considers improving the security challenges such as confidentiality, data integrity and service availability over the business transactions in a cloud environment using random number with strong DNA cryptosystem. The aim is to achieve high confidentiality and minimise execution time, communication overhead and space complexity.

In this paper, a new FSB-DC combining three chaotic logistic maps is presented. It provides a significant improvement on security and performance, of the generator proposed by Patidar *et al.* [20].

The proposed algorithm uses the binary64 floating-point arithmetic and produces at each iteration a block of 32 random bits. The assets of the FSB-DC are: high security, high level of randomness and fast execution time. The paper is structured as follows, in Section 2 the used chaotic logistic map. Section 3, presents a detailed description of Fast chaos based DNA cryptography algorithm. The Experimental analysis is given in Section 4. The paper concluded in chapter 6.

3. PROPOSED METHOD

(A) Fast Chaos-Based DNA Cryptography

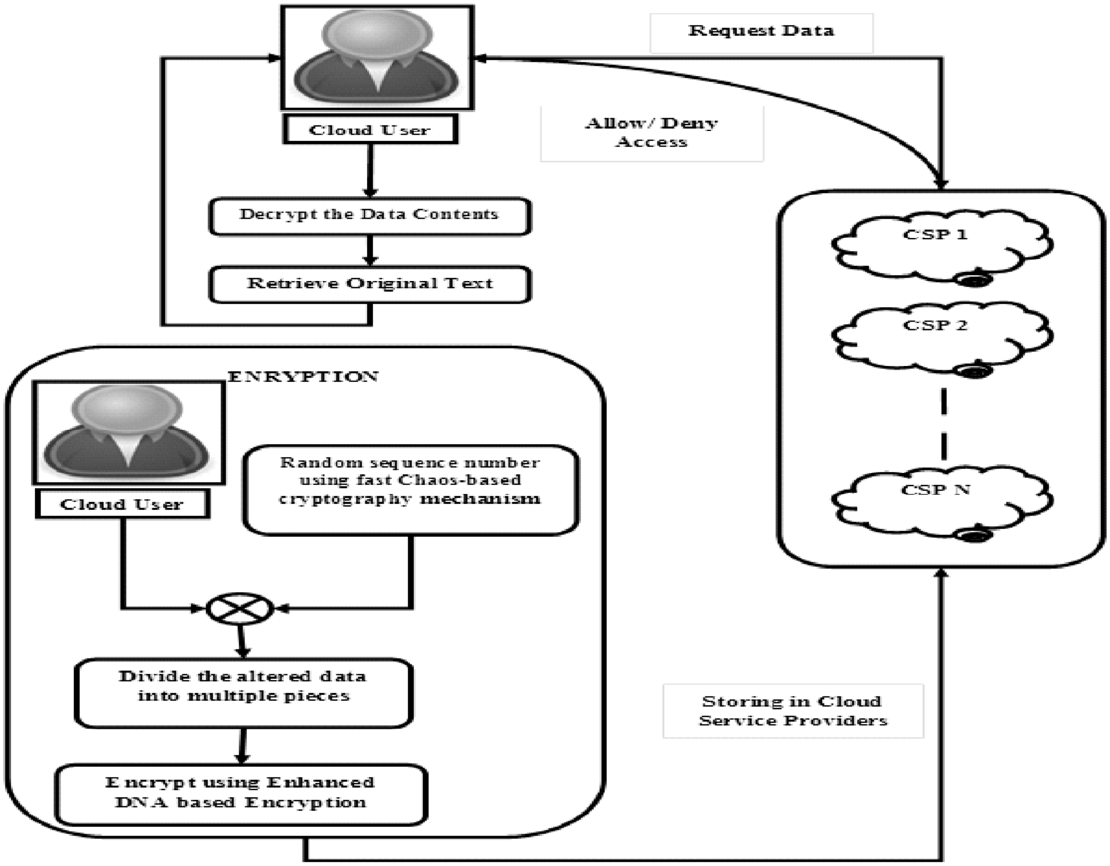
Cloud computing offers significant perspective, but at the same time, it holds many security risks and challenges. To ensure better safety, the data availability achieved by splitting down the user's significant data block into pieces and scatter among the available Cloud Service Providers (CSP). Each divided part of data can protect by utilising some interesting features of DNA sequences and data hiding.

There are many types of research has been conducted towards achieving security in data transmission using DNA-based encryption algorithm. This algorithm with 2-bit binary encoding procedure might tend to security violation in the modern computing technologies where it consists of only 24 possible encoding methods. This problem resolved in the proposed research methodology by introducing the Fast Chaos-based DNA Cryptography (FCB-DC) System in which 3-bit binary encoding is performed to increase the complexity. The proposed algorithm would generate possible encoding methods than the 2-bit binary encoding procedure.

In addition to that, the randomness of the FCB-DC is improved to prevent the security violation that might arise in the real world application. Processing achieves it and altering the plain text using the random sequence number before encryption. The Fast Chaos-based cryptography mechanism is used to generate the random sequence number which improves the randomness with high de-correlation by combining three logistic maps in the algorithmic procedure.

Generating the random number sequence using fast chaos-based cryptography mechanism and altering the plain text to increase the randomness encrypting the altered plain text using the Fast Chaos-based DNA Cryptography (FCB-DC) System with the goal of attaining increased complexity and storing them in clouds. Decrypting the ciphertexts that are present in clouds also execute in a secure way. Figure 1 shows the block diagram of Fast Chaos-based DNA Cryptography (FCB-DC) mechanism.

Figure 1: Block Diagram of Fast Chaos-Based DNA Cryptography Framework Figure 1 provides the overall preview of the working scenario of the FCB-DC methodology in the detailed manner from which it can see that the cloud users can perform their data storage and transaction processes in the more secure way. This framework contains three modules that are


Request Data
Cloud User
Allow/ Deny Access
Decrypt the Data Contents
Retrieve Original Text
CSP 1
CSP 2
ENRYPTION
Cloud User
Random sequence number using fast Chaos-based cryptography mechanism
CSP N
Divide the altered data into multiple pieces
Storing in Cloud Service Providers
Encrypt using Enhanced DNA based Encryption

1) Random sequence number generation.

2) Encryption.

3) Decryption.

This method consider three entities of Cloud User '$CU = CU_1, CU_2, ..., CU_n$' wants to share their transaction data 'TD' with the Cloud Server '$CS = CS_1, CS_2, ..., CS_n'$' respectively.

(B) Random Sequence Number Generation

In the proposed system, initially, random number generation is performed to alter the plain text contents so that the hackers could not guess the original value of plain text even in the case of known secret keys. It strongly achieved by altering the plain text by XORing it with the random number which generated. However the XOR between the plain text which might be text or non-text with the random number which is a combination of binary number. After converting the plain text into binary numbers which are then can be XORed with the random number. For example,

Random Number	0	0	1	0	0	1	0	0
Binary converted Plain Text	0	0	1	0	0	0	0	1
XOR Result	0	0	0	0	0	1	0	1

This modified plain text content will be encrypted using the DNA cryptosystem. Thus the tremendous improvement in the security of the system can be obtained. In cloud environment randomness of the contents that stored in cloud storage plays a more important role which would increase the complexity of the cryptosystem. Thus the random guess possibility of the attackers can be avoided considerably.

One interesting way to design pseudo-random generators could found in chaos theory. Indeed, chaotic systems are characterised by their high sensitivity to initial parameters and some properties like ergodicity, mixing property and great complexity. For a high-security level, it is necessary to combine several logistics maps, to increase the complexity of the cryptosystem. However, this is not always sufficient, because an accurate analysis is more appropriate to evaluate the chance level and the global security of the generator. In this work, a fast Chaos-based cryptography mechanism combining three chaotic logistic maps are presented. It provides a significant improvement in safety and performance of the pseudo random generator.

Chaotic Logistic Map

The logistic map of chaos theory within the chaos-based cryptosystems represented by the equation 1:

$$F(X) = \beta\, X\, (1 - X) \tag{1}$$

With β between 3.57 and 4.0. The chaotic behaviour has widely studied and several generators have already used such logistic map for generating pseudo-random numbers. To avoid non-chaotic behaviour of the value of β should be near 4.0, which correspond to a highly chaotic behaviour. The logistic map used under the iterative form:

$$X_{\mathrm{n}+1} = \beta\, X_n\, (1 - X_n),\ \forall\, n \geq 0 \tag{2}$$

Here, the value of X_0 and X_n are the real numbers between the interval of]0, 1[.

Fast Chaos-Based Cryptography Mechanism

In the proposed system, only binary 64 floating-point formats focused, which is used to attain an excellent simulation accuracy for the study of chaotic regimes. The proposed algorithm uses the same type of chaotic logistic map given by the equation (1). In this case, the value of β fixed to 3.9999 that correspond to a highly chaotic case. Indeed, the Lyapunov exponent measures the chaotic behaviour of a function and the corresponding Lyapunov supporter of the logistic map for β = 3.9999 is 0.69 which is very close to 0.59. The disorganised logistic map used under the iterative form given in equation 3:

$$X_{n+1} = 3.9999 X_n (1 - X_n),\ \forall\, n \geq \tag{3}$$

The starting seed X_0 is a real number that belongs to]0, 1[. All the computed elements X_n are also real numbers in]0, 1[. The proposed algorithm takes into account the various weaknesses of the algorithm proposed by Patidar *et al.* Thus, to have a large space of output sequences, three logistic maps are used during the generation process. The same value of â used for each one and the corresponding equations presented in the equations 4,5 and 6.

$$X_{n+1} = 3.9999\, X_n (1 - X_n),\ \forall\, n \geq 0 \tag{4}$$

$$Y_{n+1} = 3.9999\, Y_n (1 - Y_n),\ \forall\, n \geq 0 \tag{5}$$

$$Z_{n+1} = 3.9999 Z_\mathrm{n} (1 - Z_n),\ \forall\, n \geq 0 \tag{6}$$

For each computed value X_n, Y_n and Z_n, a binary 64 floating-point representation is used. The algorithmic technique is simple and consists at each iteration, to apply a XOR operation on the 32 bits of the mantissa of the three output elements X_n, Y_n and Z_n. Thus, the algorithm allows producing 32 random bits per iteration and therefore increasing the throughput of the generator.

The seed elements X_n, Y_n and Z_n are almost same for initial rounds. The generation process starts from seeds which are X_k, Y_k and Z_k. Using to start the kth iteration is eliminate the pseudo-random sequences at the beginning of the selection. The preliminary rounds k and the way to choose the first seeds presented in the following subsection.

Initial Seed Selection

We improve the randomness quality, the choice of generated sequences of the initial seed values should not neglect. The coefficient values of the elements X_n, Y_n and Z_n, belong to] 0, 1[. Due to symmetric structure of the logistic map, it is necessary to choose the starting seeds in one of the two half-intervals (here] 0, 2^{-1}[) to avoid similar trajectories. In binary64 floating-point format, the calculate term (1 - X) is equal to 1 or 0 for any X in] 0, 2^{-53}[, then for a seed value in] 0, 2^{-53}[, the computed value of Eq. 2 is equivalent to âX_n. So that, the initial seed values must choose in] 2^{-53}, 2^{-1}[.

Selection of Preliminary Rounds

In the case where the values of initial seeds (X_0, Y_0 and Z_0) are very close, the beginnings of chaotic course are almost similar.

It is necessary to apply some preliminary rounds before starting to produce the random bits to avoid initial seed problem.

Thus, it is a need to see at which some iterations, the difference $\delta_{[2]}$ begin to propagate.

It considers the initial seed is $X_0 = \delta_{[2]} = 2^{-49.8289}$, and the obtained trajectory with the Equation 3. It expected that the trajectory would start to oscillate almost from the 30th iteration. Thus, the generation of random bits will begin from the iteration 30. That allows de-correlating the outputs of the fast Chaos-based cryptography mechanism and then increasing the sensitivity related to the first seeds.

Finally, the output of random sequence number generated from the fast chaos-based cryptography mechanism would be XORed with the plain text value to alter them which shown in Figure 2. By doing so, random guess of the plaintext would avoid considerably. After processing the plain text, Fast Chaos-based DNA Cryptography (FCB-DC) System done on the altered plain text before sending it to the cloud storage.

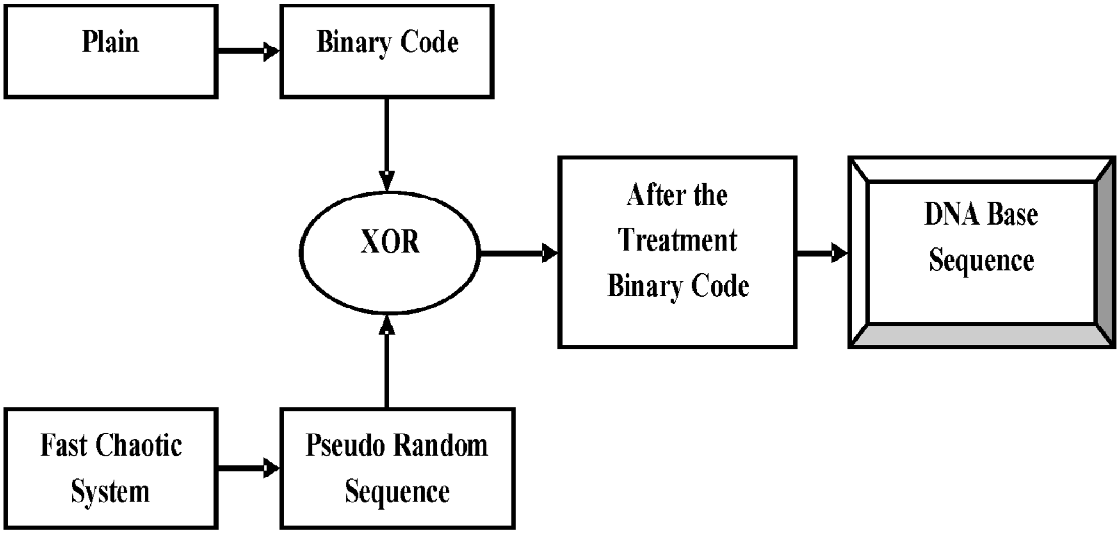


Figure 2: Fast Chaos-Based Cryptography Mechanism

(C) Encryption

The Fast Chaos-based DNA Cryptography System (FCB-DC) introduced which can lead to higher security level than the traditional DNA cryptography algorithm by replacing the 2-bit binary encoding scheme with the 3-bit binary encoding scheme. The proposed algorithm produces a possible encoding method which would increase the complexity of the cryptosystem. After encrypting that would be stored in the cloud environment so that the security can also increase. The following steps processed to reach the high-security level.

Proposed Fast Chaos-Based DNA Encryption

Fast Chaos-based DNA Cryptography System (FCB-DC) is an improved method than the traditional DNA cryptosystem due to its increased cryptosystem complexity level. Encryption can be done easily by finding the DNA sequence. In the proposed system, the encoding system is utilised to generate the DNA sequence. In the field of information science, the most basic encoding method is binary encoding, because everything can encode by the two states of 0 and 1. In real biological environment, DNA is double helix structure composed of pair of biopolymers, polynucleotide, known as

1) Purine Adenine (A),
2) Pyrimidine Cytosine (C),
3) purine Guanine (G), and
4) pyrimidine Thymine (T).

The encoding can be done easier by representing the polynucleotide in the binary format. The complexity of the cryptosystem increased in the FCB-DC by introducing the 3-bit binary encoding scheme whereas in the traditional DNA-based cryptosystem 2-bit binary encoding is used. The representation of polynucleotide represents as follows:

A(0)	*T (1)*	*C(2)*	*G (3)*
000	**001**	**010**	**011**
A$_1$(4)	**T$_1$(5)**	**C$_1$(6)**	**G$_1$(7)**
100	**101**	**110**	**111**

Obviously, by these encoding rules, there are 8! = 40320 possible encoding methods can generate which is considerably more than the traditional DNA cryptosystem. Increasing more number of encoding methods can build cryptosystem becomes more complex. Some sample encoding methods among the 40320 possible encoding methods listed in the following table 1.

Table 1
Example of Possible Encoding Methods

S.No	*Encoding Methods*	*3-bit Binary Value*							
1	ATCGA$_1$T$_1$C$_1$G$_1$	000	001	010	011	100	101	110	111
2	TCG A$_1$T$_1$C$_1$G$_1$A	001	010	011	100	101	110	111	000
3	CG A$_1$T$_1$C$_1$G$_1$A T	010	011	100	101	110	111	000	001
4	G A$_1$T$_1$C$_1$G$_1$A TC	011	100	101	110	111	000	001	010
5	A$_1$T$_1$C$_1$G$_1$A TCG	100	101	110	111	000	001	010	011
6	T$_1$C$_1$G$_1$A TCG A$_1$	101	110	111	000	001	010	011	100
7	C$_1$G$_1$A TCG A$_1$ T$_1$	110	111	000	001	010	011	100	101
8	G$_1$A TCG A$_1$ T$_1$ C$_1$	111	000	001	010	011	100	101	110

By using the encoding methods listed in Table 1, we can generate the following factors. Those are DNA Sequence Primer

DNA Sequence Generation

DNA reference sequence selected from European Bioinformatics Institute's (EBI) online database which consists of around 163 million unique DNA sequences. However, to achieve more security, we will generate DNA sequence by randomly shifting rotating eight nucleotides given in Table 1. Thus at random system will produce the shift-key range from one to 64. Thus, the system can generate 64^{64} combinations, as sixteen combinations of four nucleotides can be shifted, i.e. generating 394 trillion (more than EBI database i.e. more than163 million) unique DNA reference sequences. In this work, following DNA sequence is used for performing FCB-DC.

G $A_1T_1C_1G_1$A TC C_1G_1A TCG A_1 T_1 TCG $A_1T_1C_1G_1$A $T_1C_1G_1$A TCG A_1 ATCG$A_1T_1C_1G_1$ G_1A TCG A_1 T_1 C_1 CG $A_1T_1C_1G_1$A T $T_1C_1G_1$A TCG A_1 $A_1T_1C_1G_1$A TCG

Primer Selection

A primer is a strand of short nucleic acid sequences that give out as a starting point for DNA synthesis. It is required for DNA replication because the enzymes that catalyse this process, DNA polymerases can only add new nucleotides to an existing strand of DNA. A primer in enhanced DNA cryptosystem is used to synthesise the DNA sequence to make the prediction. There are two rules mainly intense for the better primer selection. Those are:-

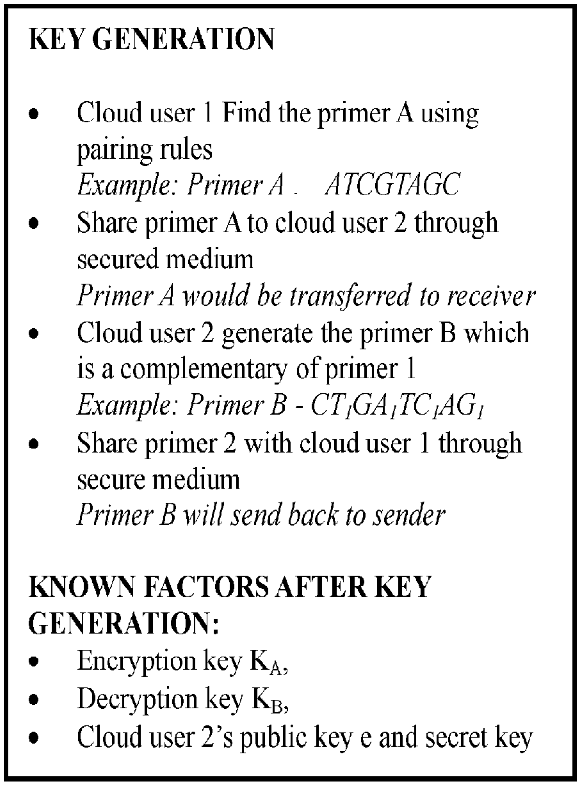
Watson_Crick Base Pairing Rule

1) Purine Adenine (A) always pairs with the pyrimidine Thymine (T).

2) Pyrimidine Cytosine (C) always pairs with the purine Guanine (G).

Complementary Rule

The base pair that is a complement to each other would pair in the complementary rule. For example 0 to 1 and 1 to 0 is the complementary pairing. After finding the primer A and the DNA sequence, encryption is done as follows:



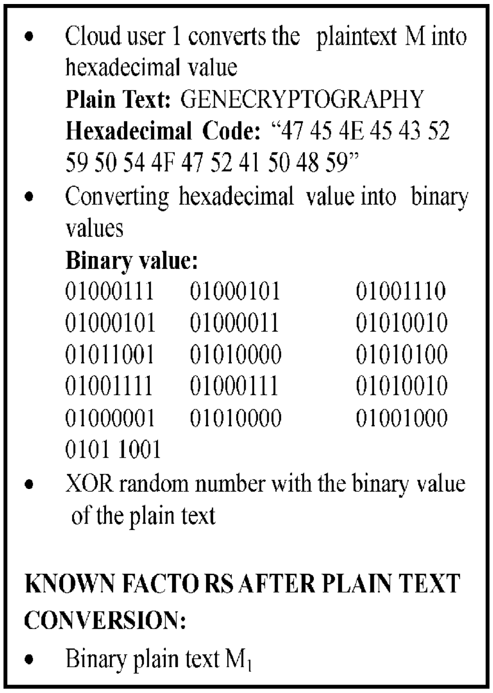
KEY GENERATION

- Cloud user 1 Find the primer A using pairing rules
  Example: Primer A . ATCGTAGC
- Share primer A to cloud user 2 through secured medium
  Primer A would be transferred to receiver
- Cloud user 2 generate the primer B which is a complementary of primer 1
  Example: Primer B - $CT_1GA_1TC_1AG_1$
- Share primer 2 with cloud user 1 through secure medium
  Primer B will send back to sender

KNOWN FACTORS AFTER KEY GENERATION:

- Encryption key K_A,
- Decryption key K_B,
- Cloud user 2's public key e and secret key

Plain Text Conversion



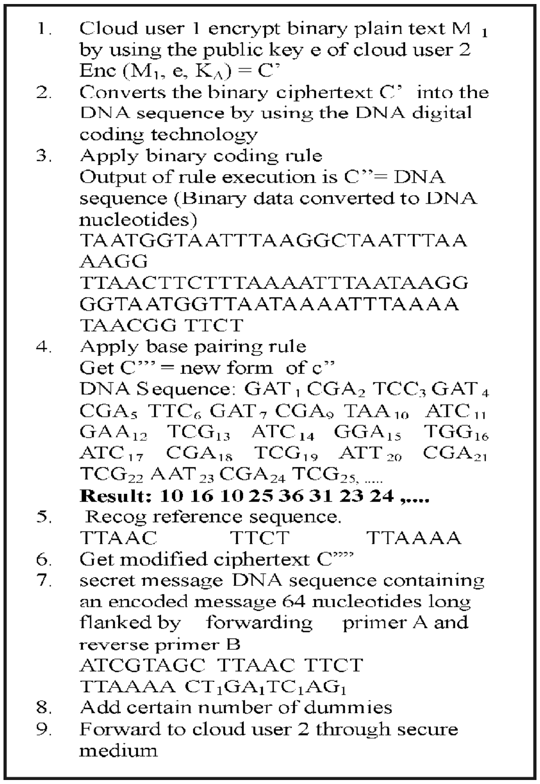
- Cloud user 1 converts the plaintext M into hexadecimal value
  Plain Text: GENECRYPTOGRAPHY
  Hexadecimal Code: "47 45 4E 45 43 52 59 50 54 4F 47 52 41 50 48 59"
- Converting hexadecimal value into binary values
  Binary value:
  01000111 01000101 01001110
  01000101 01000011 01010010
  01011001 01010000 01010100
  01001111 01000111 01010010
  01000001 01010000 01001000
  0101 1001
- XOR random number with the binary value of the plain text

KNOWN FACTO RS AFTER PLAIN TEXT CONVERSION:

- Binary plain text M_1

Encryption

In the first step, plain text modification would be done by adding the primer value with the plain text, so that hackers cannot obtain original content even they hack it.



1. Cloud user 1 encrypt binary plain text M $_1$ by using the public key e of cloud user 2
   Enc (M_1, e, K_A) = C'
2. Converts the binary ciphertext C' into the DNA sequence by using the DNA digital coding technology
3. Apply binary coding rule
   Output of rule execution is C"= DNA sequence (Binary data converted to DNA nucleotides)
   TAATGGTAATTTAAGGCTAATTTAA
   AAGG
   TTAACTTCTTTAAAATTTAATAAGG
   GGTAATGGTTAATAAAATTTAAAA
   TAACGG TTCT
4. Apply base pairing rule
   Get C''' = new form of c"
   DNA Sequence: GAT_1 CGA_2 TCC_3 GAT_4 CGA_5 TTC_6 GAT_7 CGA_9 TAA_{10} ATC_{11} GAA_{12} TCG_{13} ATC_{14} GGA_{15} TGG_{16} ATC_{17} CGA_{18} TCG_{19} ATT_{20} CGA_{21} TCG_{22} AAT_{23} CGA_{24} TCG_{25}, ...
   Result: 10 16 10 25 36 31 23 24 ,....
5. Recog reference sequence.
   TTAAC TTCT TTAAAA
6. Get modified ciphertext C''''
7. secret message DNA sequence containing an encoded message 64 nucleotides long flanked by forwarding primer A and reverse primer B
   ATCGTAGC TTAAC TTCT
   TTAAAA $CT_1GA_1TC_1AG_1$
8. Add certain number of dummies
9. Forward to cloud user 2 through secure medium

(D) Decryption

After the intended receiver cloud user 2 gets the DNA mixture i.e. ciphertext, it can easily pick out the secret message DNA sequence by using the correct primer pairs. Cloud user 2 translates the secret message with DNA sequence into the binary ciphertext C' by using the DNA digital coding technology. Then, Cloud users two can decrypt the binary ciphertext C' into the binary plaintext M_1 by using his secret key e. After the binary plaintext M_1 has been recovered, cloud user two can retrieve the plaintext M by using reverse plain text conversion process.

4. EXPERIMENTAL SETUP

The Fast Chaos-Based DNA Cryptography (FCB-DC) mechanism with efficient data computation and data storage uses Amazon EC2 dataset, a well-known and widely recognised cloud service provider using on-demand cloud services. CloudSim 3 Simulator simulates the performance evaluation. JAVA language is employed in the experimental work for evaluating the data confidentiality of cloud data storage.

Amazon EC2 dataset information is used with the tests aimed at comparing the DNA-based Cryptography for the multi-cloud system (DNA-CMCS) and Random Function and Binary Arithmetic Operations (RFBAO) with the Fast Chaos-Based DNA Cryptography (FCB-DC) mechanism.

To study the FCB-DC mechanism using the simulator, we proposed a simulation environment that has the following parameters: the transactional data to be stored in the cloud server ranges between 10KB to 70KB. The following settings including various sizes of transactional data, data confidentiality, execution time, communication overhead and space complexity during data storage in a cloud environment are evaluated.

(A) Data Confidentiality

Data Confidentiality calculated the information stored on a system protected against unintended or unauthorised access. Data Confidentiality has measured the capacity of the system to protect its data frequently. Data confidentiality is defined on mathematically as follows.

$$DC = Size\,(TD) - Size\,(compromised\ TD) \tag{7}$$

In the equation (7), the data confidentiality 'DC' is obtained according to the size of transactional data and the size of compromised data respectively. Higher the data confidentiality gave more efficient and measured regarding kilobytes (KB).

Table 2
The Performance Comparison of Data Confidentiality

TransactionalData size (KB)	*Data Confidentiality (KB)*		
	FSB-DC	*DNA-CMCS*	*RFBAO*
10	11.3	7.6	7.1
20	19.5	15.5	14.5
30	29.4	23.2	21.8
40	38.3	33.7	31.5
50	50.5	45.4	41.6
60	57.7	56.2	53.2
70	67.3	62.8	58.4

Table 2 illustrates data confidentiality based on transaction data size for proposed FCB-DC method and existing DNA-CMCS and RFBAO methods. From the Table 2, it is clear that for the increase in transaction data

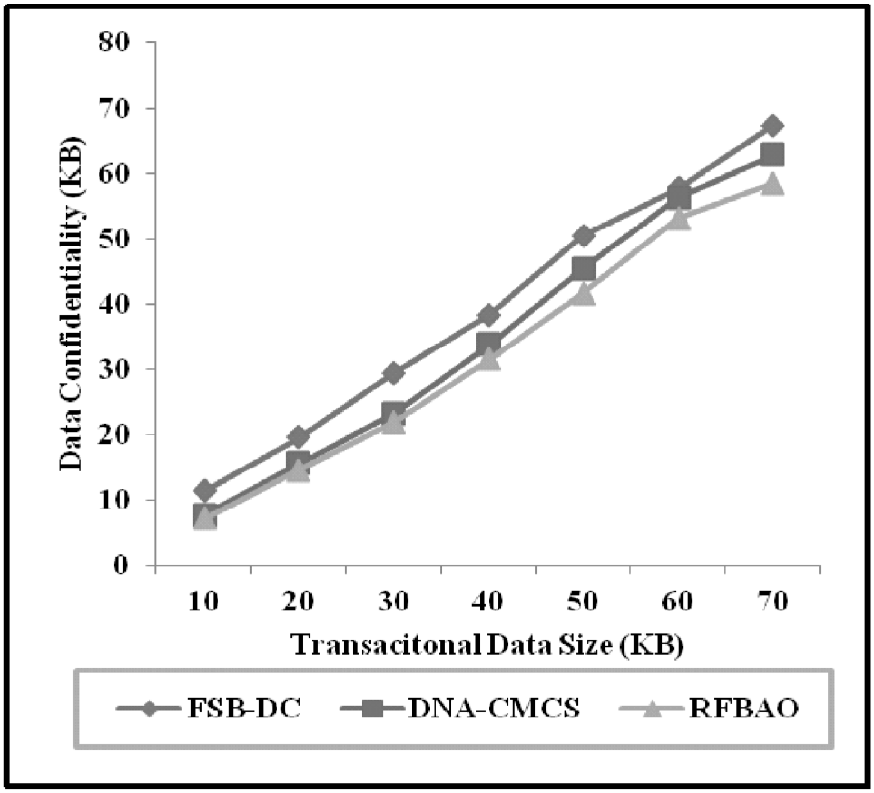


Figure 3: Data Confidentiality Comparison

size, the data confidentiality also increased for all methods. However, proposed FCB-DC method provides better performance regarding improving data confidentiality when compared to existing methods.

As shown in Figure 3, proposed FCB-DC method provides higher data confidentiality when compared to existing methods. The data confidentiality of proposed FCB-DC method is improved by 15% when compared to existing DNA-CMCS method and 21% when compared to existing RFBAO method respectively.

(B) Execution Time

Execution time is defined as the total time taken to performing encryption on submitted plain text to ensure the security level. The performance of execution time is measured regarding milliseconds (ms). The mathematical formulation for execution time is as given below.

$$ET = \Sigma_{size=1}^{n} TD_{size} * (Time_{data\ encoding}) \qquad (8)$$

In equation (8), the execution time 'ET', is described using transaction data size TD_{size} and time for '$Time_{data\ encoding}$' respectively. Lower execution time ensures that cloud users have access to the cloud data owner files in an easily accessible manner.

The effectiveness of the proposed FCB-DC framework, the experimental result of execution time is reported in Table 3. The results reported here confirm that with the increase in the transactional data size, the execution time also increases.

Table 3
The Performance Comparison of Execution Time

TransactionalData size (KB)	*Execution Time (ms)*		
	FSB-DC	*DNA-CMCS*	*RFBAO*
10	276	312	325
20	308	368	425
30	450	482	520
40	502	540	586
50	565	693	723
60	602	712	765
70	628	789	799

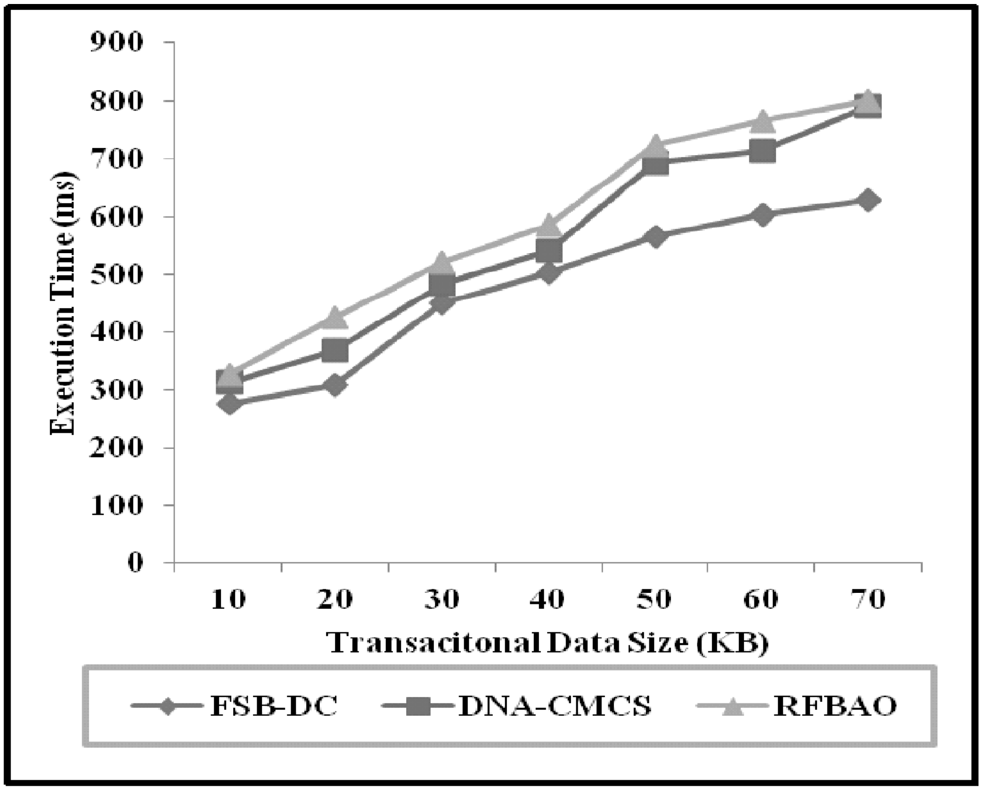


Figure 4: Execution Time Comparison

Figure 4 presents the variation of execution time on transactional data size. This FCB-DC framework, in turn, reduces the execution time by 16% compared to DNA-CMCS and 24% about RFBAO.

(C) Communication Overhead

Communication Overhead is the proportion of time to spend communicating with the group instead of getting productive work done. Communication is essential. With the increase in the size of transactional data the Communication Overhead also increased. The mathematical formulation for communication overhead is as given below.

$$CO = \Sigma_{i=1}^{n} \frac{Size(TD_i) * TD_{loss}}{timestamp} \tag{9}$$

In equation (9), the communication overheadCO is measured by multiplying the transaction data size and the transaction data lost in specified timestamp. It is measured in terms of bits per second (bps). Lower the communication overhead is more efficient.

Table 4
The Performance Comparison of Communication Overhead

Transactional Data size (KB)	*Communication Overhead (bps)*		
	FSB-DC	*DNA-CMCS*	*RFBAO*
10	3.57	4.84	5.61
20	4.89	5.45	6.27
30	8.04	8.71	9.81
40	11.18	12.43	13.51
50	13.42	15.42	16.05
60	14.89	15.62	17.04
70	16.72	17.92	18.2

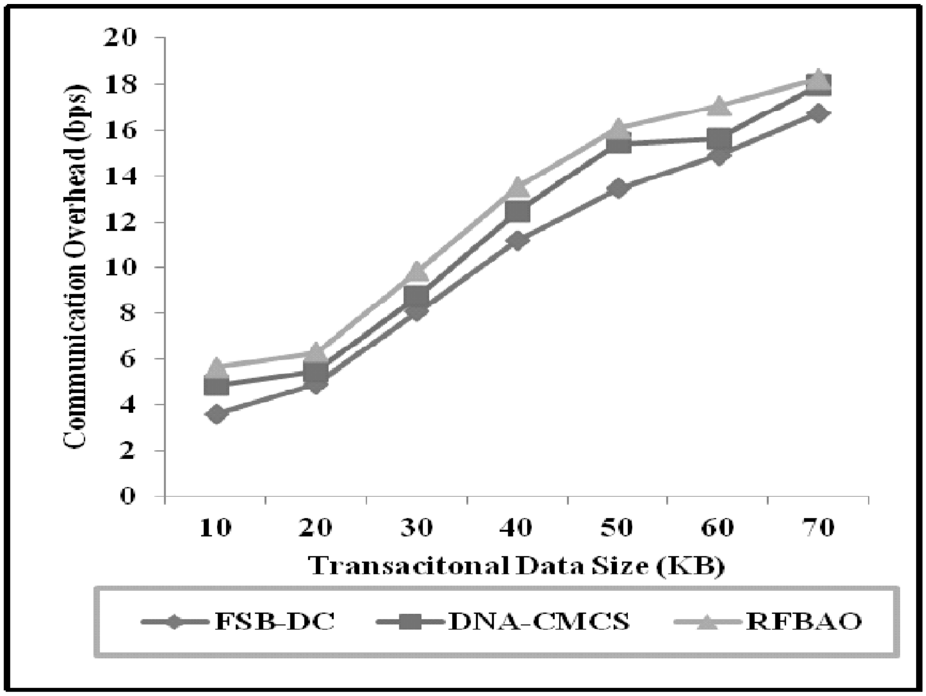


Figure 5: Communication Overhead Comparison

In the FCB-DC framework, the communication overhead results shown in Table 4 and Figure 5. The 3-bit binary coding performed with randomness control is concentrated and reduces the communication overhead by 13% compared to DNA-CMCS and 24% about RFBAO.

(D) Space Complexity

Space complexity is a measure of the total amount of working space taken by the algorithm on input size. It means that the entire usage in the worst case needed at any point in the algorithm. The mathematical formulation for space complexity is as given below.

$$SC = Mem\ (FCB - DC) * TDS_i \tag{10}$$

In equation (10), the space complexity attained by measuring the memory required for algorithms used in the Fast chaos-based cryptography framework on the input transactional data sizeTDS_i respectively.

Table 5
The performance comparison of Space Complexity

Transactional Data size (KB)	*Space Complexity (bps)*		
	FSB-DC	*DNA-CMCS*	*RFBAO*
10	12	17	19
20	21	28	33
30	27	35	37
40	35	41	44
50	48	52	56
60	55	58	61
70	62	64	68

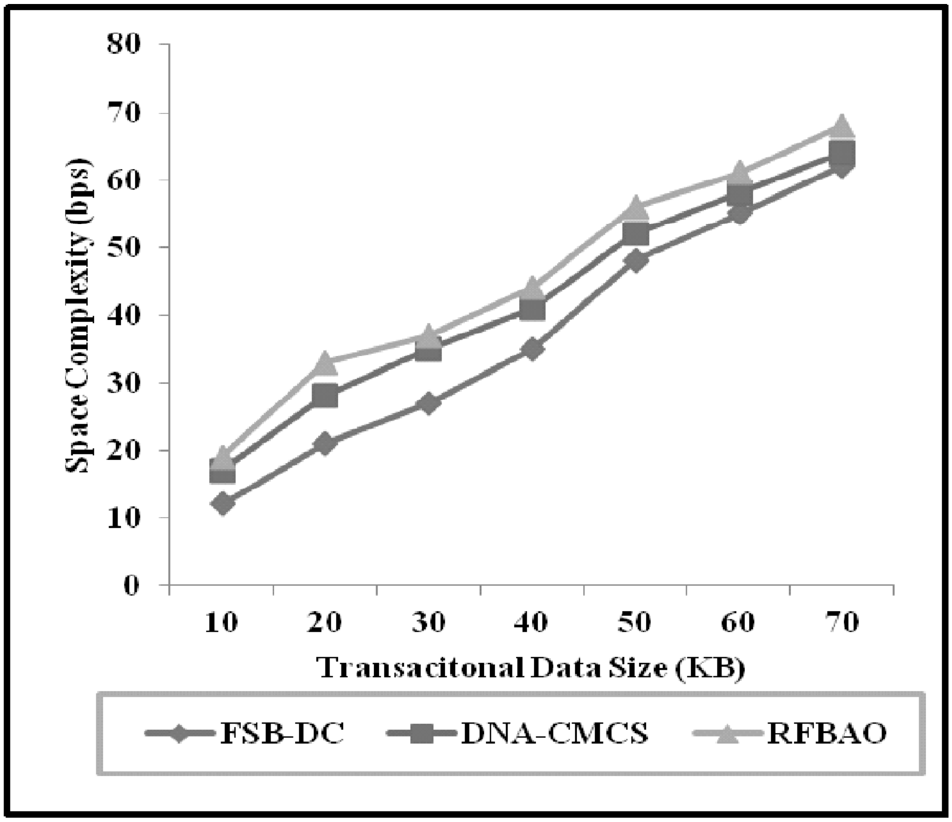


Figure 7: Space Complexity Comparison

Table 5 and figure 7 illustrates the measure of space complexity on varied transactional data sizes. Therefore, the space complexity is reduced using FCB-DC by 20% compared to DNA-CMCS and 31% about RFBAO.

5. CONCLUSION

Secured data transmission through unsecured cloud medium becomes most challenging task in the real world. The proposed research of this work concentrates on the achieving higher confidentiality and security in the cloud framework. Thus user satisfaction level can be fulfilled via introducing the fast chaos-based DNA cryptosystem which uses the 3-bit binary coding system. Thus the cryptosystem complexity is increased more. The intruders prevented from the data collision by introducing the fast chaos cryptosystem which can lead to more randomness. Thus the guessing of plain text may not be possible. In addition to that, the proposed system also prevents the unauthorised users from data access. The experimental tests conducted proved that the proposed framework can provide highly secured cloud environment for the cloud users than the existing methods.

REFERENCES

[1] R.H. Ranalkar and B.D. Phulpagar, "DNA-based Cryptography in Multi-Cloud: Security Strategy and Analysis", International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), Vol. 3, No. 2, pp.189-192, 2014.

[2] V. Siddaramappa, "Data Security in DNA Sequence Using Random Function and Binary Arithmetic Operations", International Journal of Scientific and Research Publications, Vol. 2, No. 7, pp. 1-3, 2012.

[3] S. Sakr, A. Liu, D.M. Batista and M. Alomari, "A Survey of Large Scale Data Management Approaches in Cloud Environments", IEEE Communications Surveys & Tutorials, Vol. 13, No. 3, pp. 311-336, 2011.

[4] M. Tu, P. Li, I. Yen, B. Thuraisingham, and L. Khan, "Secure Data Objects Replication in Data Grid", IEEE Transactions on Dependable and Secure Computing, Vol. 7, No. 1, pp. 50-64.

[5] R. Pragaladan and Dr.S. Sathappan, "High Confidential Data Storage using DNA Structure for Cloud Environment", International Conference on Computational Systems and Information Systems for Sustainable Solutions, IEEE Xplore, (DOI: 10.1109/CSITSS.2016.7779391), pp. 382-387, 2016.

[6] K. Menaka, "Message Encryption Using DNA Sequences", 2014 World Congress on Computing and Communication Technologies - CPS, IEEE DOI 10.1109/WCCCT.2014.35, pp. 182-184, 2014.

[7] D. Tulpan, C. Regoui, G. Durand, L. Belliveau and S. Léger, "HyDEn: a hybrid steg anocryptographic approach for data encryption using randomized error-correcting DNA codes. Bio Med research international, 2013.

[8] M. Alzain, B. Soh and E. Pardede, "MCDB: Using Multi-Clouds to Ensure Security in Cloud Computing", IEEE Conference on Dependable, Autonomic and Secure Computing, pp. 784 – 791, 2011.

[9] V. Gramoli and R. Guerraoui, "A Comparison of Secure Multi-Tenancy Architectures for Filesystem Storage Clouds", in Fabio Kon, Anne-Marie Kermarrec (ed.) Middleware 2011: ACM/IFIP/USENIX 12th International Middleware Conference, Lisbon, Portugal. Berlin Heidelberg: Springer-Verlag Berlin Heidelberg, pp. 471-490, 2011.

[10] S. Lee, H. Park and Y. Shin, "Cloud Computing Availability: Multi-clouds for Big Data Service' in Convergence and Hybrid Information Technology", 6th International Conference, ICHIT 2012 Proceedings, Volume 310, New York: Springer, pp. 799-806, 2012.

[11] S. Prakash, K. Subramanyam and S. Prasad, "Multi-Clouds Model for Service Availability and Security", International Journal of Computer Science & Engineering Technology (IJCSET), Vol. 4, No. 2, pp. 158-161, 2013.

[12] K.D. Bowers, A. Juels and A. Oprea, "HAIL: A high-availability and integrity layer for cloud storage", CCS'09: Proc. 16th ACM Conf. on Computer and Communications Security, pp.187-198, 2009.

[13] H. Abu-Libdeh, L. Princehouse and H. Weatherspoon, "RACS: a case for cloud storage diversity'', SoCC'10: Proc. 1st ACM Symposium on Cloud Computing, pp. 229-240, 2010.

[14] C. Cachin, R. Haas and M. Vukolic, "Research Report: Dependable Storage in the Intercloud, Published in RZ3783 in 2010 (6 Pages) edn., Switzerland: IBM Research – Zurich, 2010.

[15] P. Mahajan, S. Setty, S. Lee, A. Clement, L. Alvisi, M. Dahlin and M. Walfish, "Depot: Cloud storage with minimal trust", OSDI' 10: Proceedings of the 9th USENIX Conference on Operating Systems Design and Implementation, pp. 1-16, 2010.

[16] F. Rocha and M. Correia, "Lucy in the Sky without Diamonds: Stealing Confidential Data in the Cloud", Proceedings of 1st International Workshop Dependability of Clouds, Data Centers and Virtual Computing Environments, pp. 1-6, 2011.

[17] A.J. Feldman, W.P. Zeller, M.J. Freedman and E.W. Felten), "SPORC: Group Collaboration using Untrusted Cloud Resources", Published in OSDI, pp. 1-14, 2010.

[18] A. Shraer, C. Cachin, A. Cidon, I. Keidar, Y. Michalevsky and D. Shaket, "Venus: Verification for untrusted cloud storage", CCSW' 10: Proceedings - ACM Workshop on Cloud Computing Security, pp. 19-30, 2010.

[19] M. François, D. Defour and C. Negre, "A Fast Chaos-Based Pseudo-Random Bit Generator Using Binary 64 Floating-Point Arithmetic", Informatica, Vol. 38, pp. 115–124, 2014.

[20] V. Patidar, K.K. Sud and N.K. Pareek, "A Pseudo Random Bit Generator Based on Chaotic Logistic Map and its Statistical Testing", Informatica, Vol. 33, No. 4, pp. 441–452, 2009.