National College of Ireland

An Improved Genetic Algorithm using Tree Traversal and Advanced Encryption Standard For Public Cloud Storage

MSc in Cloud Computing
Programme Name

Deep Roychoudhury
Student ID: 19109130

School of Computing
National College of Ireland

Supervisor: Sean Heeney

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Deep Roychoudhury
Student ID:	19109130
Programme:	Programme Name
Year:	2020
Module:	MSc in Cloud Computing
Supervisor:	Sean Heeney
Submission Due Date:	17/12/2020
Project Title:	An Improved Genetic Algorithm using Tree Traversal and Advanced Encryption Standard For Public Cloud Storage
Word Count:	6220
Page Count:	21

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	17th December 2020

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

An Improved Genetic Algorithm using Tree Traversal and Advanced Encryption Standard For Public Cloud Storage

Deep Roychoudhury
19109130

Abstract

Public cloud data storage is one of the major concerns in today's world. There are various cloud service providers (CSPs) that provide encryption for cloud storage where the data is first transferred to the cloud and then it is encrypted. This creates a lack of trust among the users or consumers of CSPs as the data is directly transferred to the public cloud. Again, there is a possibility in the public cloud that the storage of multiple organizations could be stored on the same Availability Zone resulting in possible exposure to data. In this research paper, a solution has been provided to avoid the trust issue among the Enterprise level organizations by providing a Restful web-service to encrypt and decrypt the data. The computation has been done in Amazon Web Service (AWS) and the data storage is done on Microsoft Azure Blob Storage (Azure) to avoid computational and data exposure. A layered approach has been used to encrypt the data before saving the data to the cloud using the Tree Traversal, Genetic Algorithm, and Advanced Encryption Standard (AES). Furthermore, a practical and statistical evaluation using IBM SPSS shows that the layered encryption algorithm provides an enhanced data security in the public cloud.

1 Introduction

The field of cloud computing deals with many benefits such as scalability, elasticity, profitability or cost effectiveness, etc. Reliability is a field in cloud computing that still finds it hard to win the battle for cloud computing. The primary driver for the unreliability of cloud computing is shared resources. Although, there are a lot of asymmetric and symmetric encryption solutions available in this field, which are also provided by various CSPs, but still fails to pursue organizations. The papers Milap J Bhuva (2019) and Ren and Xue (2018) offer two different solutions using symmetrical and asymmetrical techniques for encryption. The data of multiple firms could be on the same availability zones and regions. One exposure to organizational data may result in a bad reputation. Therefore, much research has been done to enhance the strength of encryption. Much of the research is more focused on performance assurance, but the safety endpoint is not used. In paper Mr.B.Bharathi (2017), authors have discussed about various performance based metrics to test cryptographic algorithm based upon mathematical modelling using time complexity. Research has attempted to improve security in different ways, such as location, algorithms, application architecture, etc. In paper ShaluMall and Saroj (2018), the

authors provided an architecture whereby the processed data is divided into blocks and each block is stored in separate locations. When developing a cryptographic algorithm, the most important aspect is the use of the pseudo-random number generator (PRNG). In paper Hossein Nematzadeh (2020) the authors have tried to develop a PRNG for generating random data using Cellular Neural Networks (CNN). Apart from these there are various PRNGs that are available and has been approved by NIST Andrew Rukhin (2010). This research aims to develop a tiered cipher algorithm using Linear Congruential Generator as PRNG. Speaking of security, there are a number of models that can be used with microservice architecture to ensure security. The paper Zhenyu Wen and Xu (2020), talks about using micro-service orchestration framework to provide data security. Microservice applications offer many benefits over monolith applications. It is easier to build, maintain, delivers scalability, and also helps improve productivity and speed. This research is based for Enterprise level organizations and can be used as a micro service for encryption. The research is statistically evaluated for correlation of layered algorithms and strength determination using IBM SPSS Toolkit Norman H. Nie (2020).

1.1 Research Question

- Can several layers of safety add to the strength of the genetic algorithm?
- To what degree does the algorithm generate randomly encoded text?

1.2 Research Objectives

The main objectives of the research questions are: 1) Development of a cloud architecture with two different CSPs to avoid logic and data exposure. 2) Development of a Restful Webservice for Enterprise-level organizations so that it can be deployed as a micro-service. 3) Compute the entropy values of each overlay algorithm for strength. 4) Evaluate the strength of the algorithm/encryption with practical and statistical analysis. 5) Documenting the research work with configuration manual.

2 Literature Review

2.1 Symmetric and Asymmetric Algorithms

Two of the most common concepts for executing encryption or decryption are symmetrical and asymmetric algorithms. The research paper Akashdeep Bhardwaj (2016), describes the symmetric and asymmetric algorithms. The asymmetrical algorithm is an algorithm that uses a pair of keys to encrypt and decrypt. The basic concept is to use a public key to encrypt the data and form an encrypted text and a private key to decrypt the encrypted text to plain text. The paper describes the use of two main asymmetrical algorithms used in data encryption. The RSA (Rivest, Shamir, and Adelman) and Diffie-Hellman are the most widely used asymmetric algorithms. RSA is used extensively in web and cloud environments. In this process, a request is initially sent to a cloud service provider for authentication, and then the data is transferred to the requester using the RSA algorithm. Prime numbers are used to produce public and private keys based on mathematical equations. The plain text is encrypted in blocks and the cipher text is

multiplied with the product of plain text such that the outcome is the result of product which is also known as multiplicative homomorphic.

Deffie-Hellman Key Exchange algorithm is about exchanging cryptographic keys by deciding a common shared key. In this algorithm shared keys are used to get a common session key. This common session key is then used to generate a third session key which cannot be guessed easily by the attacker. Deffie-Hellman is not used for large datasets since, it is subjected to Man-in-the-Middle attacks.

The symmetric algorithm uses a single shared key to encrypt and decrypt the data. This algorithm provides a lower overhead when compared to asymmetric algorithms. Most common symmetric algorithms used so far are AES, 3DES, RC6 and Blowfish.

There are various disadvantages of using a symmetric key such as if a third person gets access to the shared key, the person can decrypt the data very easily. There could be tampering of data and false data could lead to operational overhead. There has been an emerging population of tools such as brute force, hashcat, burp suite, etc. that can crack the symmetric algorithms easily. Hence, privacy and authenticity could be hampered easily.

The most widely used symmetric algorithm is Advanced Encryption Standard (AES). The paper Elaine Barker (2019), a publication of NIST, provides a list of accepted symmetric algorithms. AES-128, AES-192 and AES-256 has been accepted or approved by NIST for both encryption and decryption. This paper has taken into account the use of AES-256 encryption and decryption.

2.2 Pseudo Random Number Generators

The Andrew Rukhin (2010) suggests various random and pseudorandom number generators for the purpose of encryption and decryption. There are various PRNGs such as linear congruential generator (LCG), quadratic congruential generator I (QCG-I), quadratic congruential generator-II (QCG-II), cubic congruential generator II (CCG-II), exclusive or generator (XORG),. . . ,etc. Among all these generators LCG is the most preferred PRNG since it is very fast. LCG is represented by the Equation 2. The only disadvantage of using LCG is it is not free of sequential correlation on successive calls. This research paper will take into account the LCG for generating random numbers for Genetic Algorithm. In the paper ShaluMall and Saroj (2018) the authors have also used LCG as the PRNG for their test for Genetic Algorithm because of its easy implementation.

2.3 Genetic Algorithm

The study of inheritance is referred to as genetics. The mutation of the parents' genes causes the creation of a child possessing both genes. The concept of genetics is based on two cross-cutting and mutative themes. These two genetic factors result in the development of a cryptographic algorithm called a genetic algorithm. Figure 1 shows the implementation of genetic algorithm. In ShaluMall and Saroj (2018) and Hebah H. O. Nasereddin (2020), the authors speak of implementing genetics for data security in the cloud. Here, the authors explained the crossing of two blocks of data with the help of the pseudo-random number generator (PRN) and the mutation of the two crossings. The two resulting data are stored in two distinct files and in two distinct areas or servers. Therefore, if a data set is obtained, it would be very difficult to recover the other part and check whether the two data make sense. Though, the idea is good in terms of security,

but this will translate into a lot of latency while recovering the data. Therefore, the overhead costs of this algorithm is the primary concern and may lead to increased complexity of time. In another article, Hebah H. O. Nasereddin (2020), author demonstrated the implementation of the genetic algorithm using both crossing and mutation. But the concept in this article is quite simple since, the data has been considered as 8-bit data in which the crossing takes place using the first six bits and the mutation takes place using the last two bits. The random nature of crossing and mutation is not addressed in this paper. The resulting data may be easily decrypted if the data is not random.

According to the two articles, the idea of the genetic algorithm is similar, there is an expectation that the data will be random, but that there will be overhead due to the division of the data between two different areas, while others lack random nature and are unable to ensure data security. The concept of genetic cross-breeding and mutation has the capacity to keep data safe in hybrid cloud systems if randomness and latency are eliminated. Therefore, the genetic algorithm may be improvised by implementing methods to satisfy random character and latency.

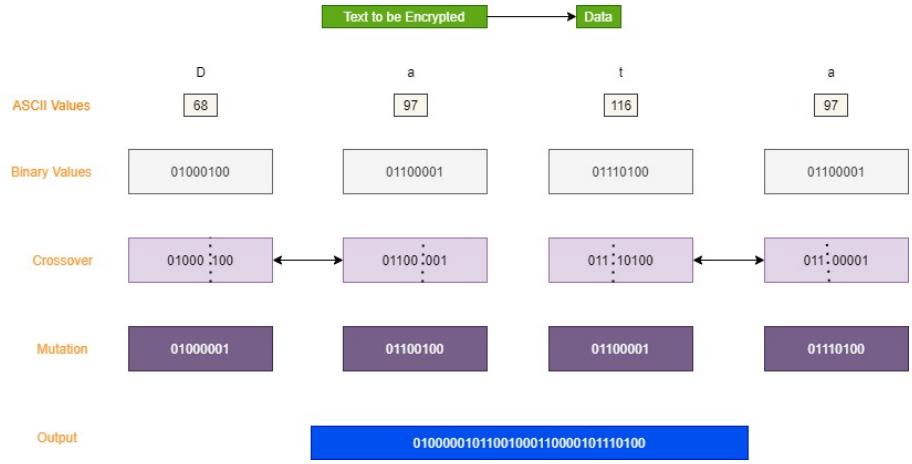

Figure 1: Genetic Algorithm

2.3.1 DNA Cryptography

There are various variants of genetic algorithms for encryption. One of these variances is the use of the DNA algorithm in the genetic algorithm to encrypt the data. The research paper E. Vidhya (2020) refers to the use of the Deffie-Hellman genetic algorithm and key exchange algorithm to improve the critical power of the DNA algorithm. Shannon entropy has been used to measure the key strength. Deoxyribonucleic acid (DNA) encryption is an algorithm that uses DNA sequences that are Adenine (A), Guanine (G), Cytosine (C) and Thymine (T) to secure data. DNA sequences are used together with the genetic algorithm and Deffie-Hellman to produce a strong encryption key. The DNA sequences in relation to the corresponding binary value are given in the figure 2.

DNA SEQUENCES	A	G	C	T
BINARY VALUE	00	01	10	11

Figure 2: DNA Sequence

In the article, the authors suggested converting raw text into binary numbers. The key was generated using Deffie-Hellman as well as the genetic algorithm. The key and binary numbers are crossed, followed by the mutation. The final outcome is converted according to the DNA sequences. While shannon entropy values show some strength, encrypted text can be decrypted in less time as there are only four DNA sequences. The likelihood of cracking the cipher may be high.

2.3.2 Hybrid Steganographic Authentication Algorithm

Similar to DNA encryption another case of using the genetic algorithm is the use of steganography. Steganography involves using an image, video or file with another to get hidden information. The Hybrid Steganographic authentication algorithm has been used in research paper Venkatramana and Geethab (2019). The paper proposed an algorithm to encrypt the image using symmetrical block encryption of the blowfish, then using the genetic operator to encrypt the information in the image again. While encryption is for image rather than text, the layered encryption algorithm provides a better correlation coefficient. The layered approach has been used in this paper to improve the security and confidentiality of public cloud storage.

2.4 Least Significant Bit Steganography And Data Encryption Standard Algorithm

Steganography is used in cryptography to enable secret communication. The paper Hambali Moshood Abiola (2020) talks about using least significant bit (LSB) for steganography with the binary numbers. Similar to Venkatramana and Geethab (2019), this paper uses layered approach to encrypt the data. The authors proposed using LSB steganography and Data Encryption Standard (DES) for data encryption. In this encryption algorithm, the single file is converted into binary and the less significant bit of each pixel or alphabets is replaced by 0 or 1. Then, the stego image is transmitted to the DES algorithm for later encryption resulting in encoded text.

Figure 3 displays the formation of stego-picture using the least significant bit. While the multi-level approach appears to be sufficiently robust, there is a lack of practical and theoretical analysis of the data.

$SI(I, j) = I_c(i, j)$, if $LSB(I_c(i, j)) = M_S$; otherwise
$SI(i, j) = I_c(i, j) - 1$, if $LSB(I_c(i, j)) = 1$ and $M_S = 0$
$SI(i, j) = I_c(i, j) + 1$, if $LSB(I_c(i, j)) = 0$ and $M_S = 1$

Pixels before Embedding:

Px 1: 10001100 01001111 01010100
Px 2: 00001111 11010101 11011010
Px 3: 11101000 11110110 10000001

Pixels after Embedding "1010111", *i.e.,* alphabet "W" using LSB Algorithm:

Px 1: 1000110**1** 0100111**0** 0101010**1**
Px 2: 0000111**0** 11010101 1101101**1**
Px 3: 1110100**1** 11110110 10000001

Figure 3: LSB Stego Image Manipulation

Moreover, the block size of DES is 64 bits, which is much smaller compared to Advanced Encryption Standard (AES) which is 128 bits. DES key size (56) is also low and DES performance is also poor. In this research project, AES has been considered since, it is a lot more reliable and faster than DES with up to 256-bit key size.

2.5 Tree Traversal

Tree Traversal is a data structure and algorithmic concept with nearly infinite cryptographic capabilities. The main benefit of using the tree traversal algorithm in encryption is that it offers flexibility to the organization as required. There can be infinite possibilities for placing data in the tree. For the purpose of this paper binary search tree as shown in Figure 9, pre-order traversal, post-order traversal and in-order traversal has been used to shuffle the data obtained. A tree can be any length and can go any direction. The logic of a tree can be arranged by the user.

2.6 DESCAST Algorithm

In Sengupta and Chinnasamy (2015), authors proposed research using the hybrid DESCAST algorithm for cloud security. The proposed algorithm of the paper is shown in Figure 4. The algorithm uses a layered approach to encrypt the data. The data is initially encrypted by the CAST algorithm and then by the DES algorithm. CAST is a symmetrical key block cipher with a 64-bit block size and a key size between 40 and 128 bits. The authors proposed this algorithm to resolve the complexity and calculation time of the individual DES and CAST algorithms.

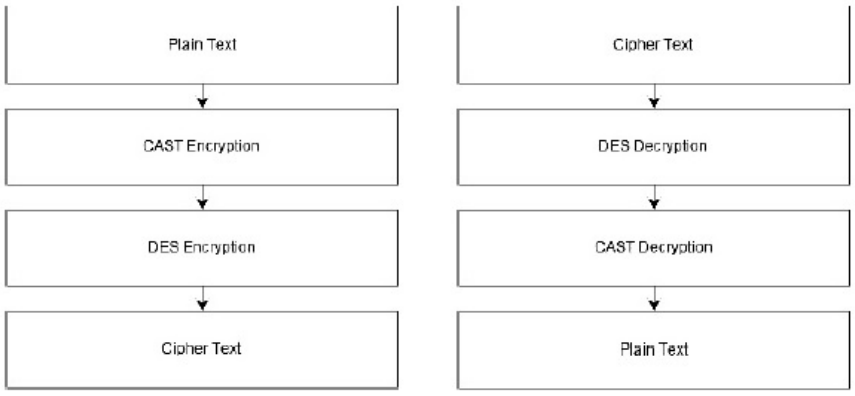

Fig. 3. (a) Encryption; (b) Decryption.

Figure 4: DESCAST Encryption

2.7 Enhanced Mutual Trusted Access Control Algorithm (EMTACA)

Enhanced Mutual Trusted Access Control algorithm is an enterprise level algorithm that takes into account the trust and reputation of the user in cloud. The paper Sarojini G (2016) describes how, alongside the entity relationship for user access, certain parameters such as resource use, user access frequency, vulnerability, etc. are used to grant user access. Figure 5 shows the trust and reputation system.

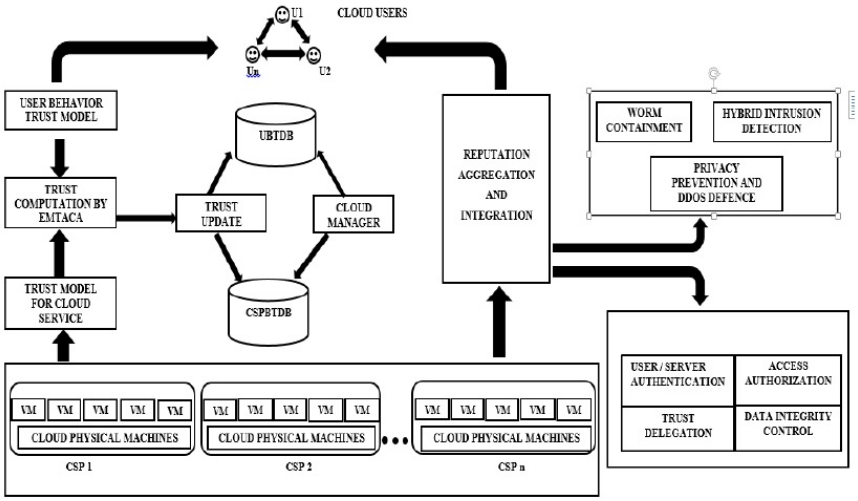

Figure 5: Trust and Reputation System

The performance evaluation shows that the performance of this system is superior to the entity's regular framework. While, this system is secure, the maintenance cost of this system would be very high as there are several applications to manage the tasks. There is a high computation requirement for this system.

3 Methodology

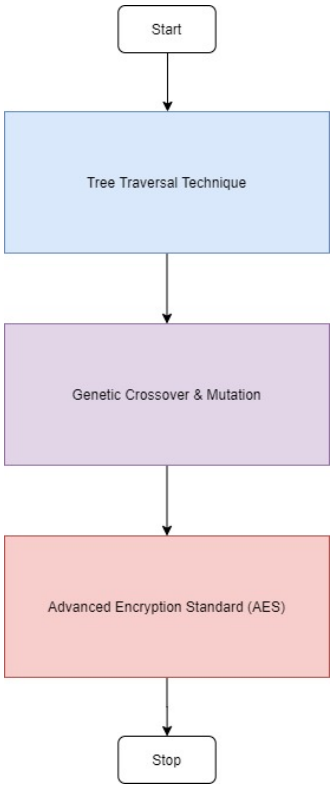

Figure 6: Encryption Steps

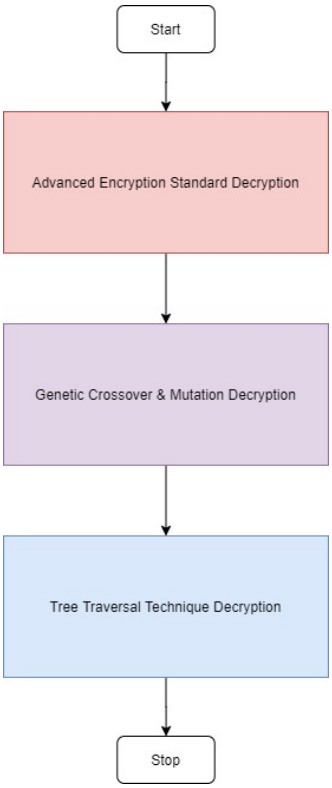

Figure 7: Decryption Steps

The methodology proposed in this paper has been developed with consideration of both random and latent factors. The strength of an algorithm can be determined by its ability to generate random numbers. The greater the randomness, the greater the complexity of extracting data from an cipher text. The proposed general steps for encrypting and decrypting are illustrated in the Figure 6 and 7.

The data can be encrypted using the suggested algorithm by performing a POST request call to the Restful web service as shown in the figure 8. The data owner will send the username, password, the group name that will have access to the data, the secret message that needs to be encrypted and the name of the file in which it will be saved. The decryption steps is the reverse of encryption steps. It makes use of entity relationship as shown in Figure 13 to authorize user using a GET request. The authorization will check if the user is eligible to access the data and whether the user credentials are fine.

Once the data owner has made the post request, the encryption data will be transmitted to the following algorithms in order.

- Tree Traversal Encryption Algorithm (or binary shifting algorithm)
- The Genetic Encryption Algorithm
- The AES Algorithm

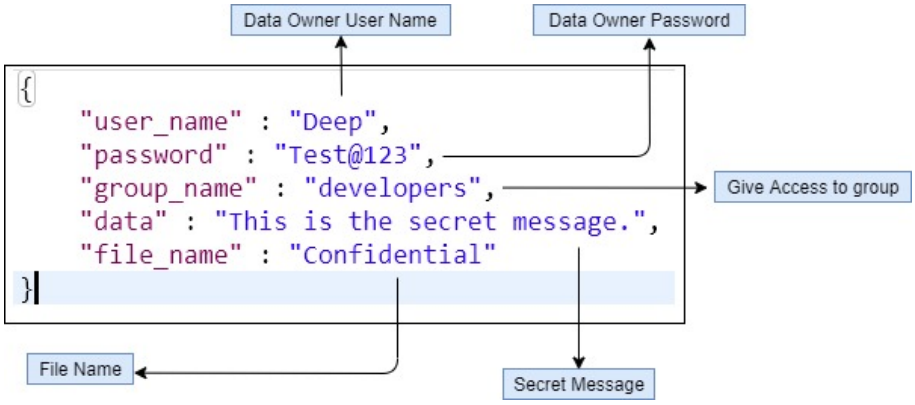

Figure 8: Post Request for Encryption

3.1 Tree Traversal Encryption Algorithm

– The post request sends the data to the tree traversal algorithm using the Model View Controller (MVC) design pattern. The data received is then translated into ASCII characters. At the end of the ASCII characters, the character positions are added and the data is passed in the tree. The tree traversed data, gives the user ability to arrange the data as per the requirement. The data has been converted to a binary search tree (BST). Figure 9 shows the structure of Binary search tree.

Each circle represents a node and each nodes can have at most 2 child nodes. The left child is smaller than the parent node and right child is greater than the parent node. Adding the position index on the binary search tree data, increases the complexity by two times. The permutation computation of the new number increases by 10 times since the length increases by 2. For example, the permutation of a three-digit number will be

6 whereas the addition of two additional digits will change the length to five, resulting in the permutation value of 120.

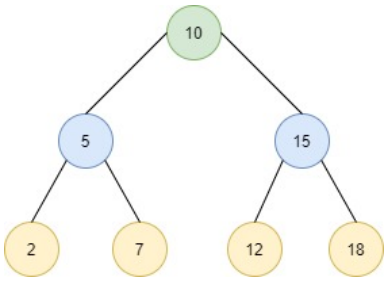

Figure 9: BST Structure

Thus, the probability of retrieving the data is even lower. The data is again scanned in pre-order, post-order and in-order based on the random value generated using the size of the value. The paper Hossein Nematzadeh (2020) used BST to encode the image with DNA and also proved the robustness of the tree method. One of the biggest benefits of using tree traversal technique is that it can be traversed in different possible ways unlike the usual linear data structures such as array, linked list, queues, stacks, etc. Tree Traversal provides the ability to shuffle the data as per the requirements of organization.

The Figure 10 shows the flow diagram of binary shifting of data used in the proposed encryption algorithm. Since, the tree formed in this case is of binary form i.e., a single parent node can have at most 2 child nodes, it is named as binary shifting. In accordance with the diagram, the data owner enters the data for encryption using a post request. The data owner specifies the name of the data owner, the password and the user groups that will have the ability to decipher the data. The algorithm verifies whether the request comes from the data owner and implements it.

This research methodology has taken hundredth position as the position setter component in order to reduce the time complexity. The end result of this output is then converted into a binary search tree that mixes the data into a tree structure. The data is then arranged into pre-order, post-order and in-order traversal based upon the number attained from the equation 1.

$$X = (\alpha N) \mod \beta \tag{1}$$

where N is the size of the string, α is the constant that is multiplied by N, β is the modulus constant, to fetch a value between 1 and β. The output after the tree traversal is then passed onto the genetic algorithm for next level of encryption.

3.2 Genetic Encryption Algorithm

Figure 11 demonstrates the application of the genetic algorithm in this research. It is clear from Section 3 that the genetic algorithm uses crossing and mutating to obtain binary encryption text. The data from binary shifting algorithm is transferred to the genetic encryption algorithm. The algorithm first retrieves the identifier of the owner of the data, then the identifier of the group authorized to access the file, from a relational database. If data are available, these will be submitted for further processing.

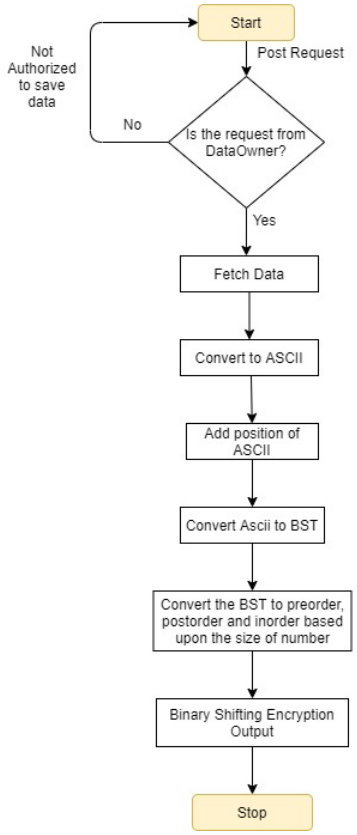

Figure 10: Tree Traversal Algorithm

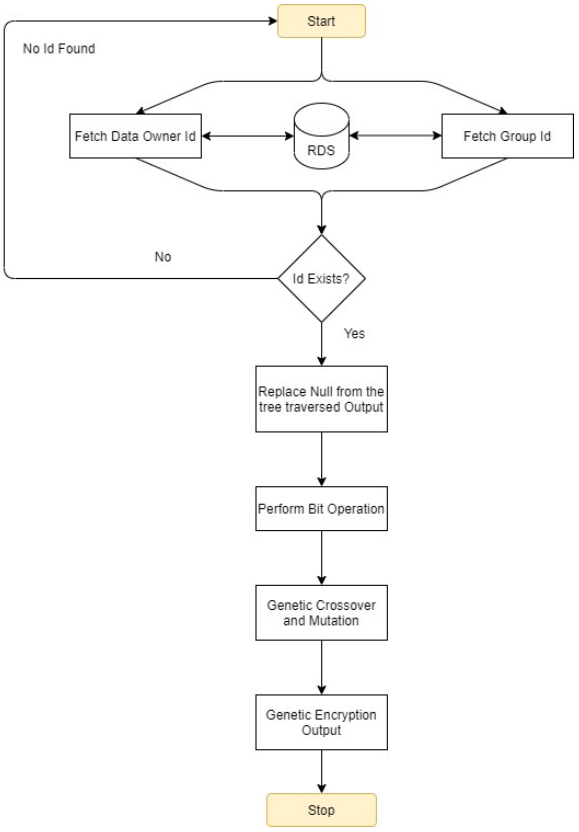

Figure 11: Genetic Algorithm

Below are the steps that the algorithm will follow:

- The null value from the original value of ASCII values obtained from tree traversal will be replaced.
- After the replacement, bit operation will be performed .i.e. numbers will be converted to binary format in ones and zeros. The binary numbers formed will be checked for equal lengths.
- Finally, the data obtained will be subjected to genetic cross-over and mutation phase, where the numbers from two blocks of data will be interchanged depending upon the pseudo random number generated (PRNG) using linear congruential generator(LCG) value.

3.2.1 Linear Congruential Generator (LCG)

One of the most popular and oldest algorithms for the generation of pseudo-random numbers (PRNG) is the linear congruential generator. The main reasons for its popularity in the field of cryptography study is because of its easy implementation, speed and less memory needs. The research paper ShaluMall and Saroj (2018) has also implemented linear congruential generator to generate PRNG. The equation that defines the linear congruential generator is the one depicted in the equation. 2.

$$X_{i+1} = (aX_i + c) \mod m \tag{2}$$

where a is the multiplier, c is the increment and, m is the modulus number.

The proposed research has considered a to be Data Owner Id, X_i as the group Id of the group who can access the data and c as the length of the data to be encrypted. m value has been considered as 3 for ease of implementation. The mod value will generate a value between 1 and 3 in order to determine the cross-over value.

3.3 Advanced Encryption Standard (AES)

Symmetric algorithms use single key for its encryption and decryption and have the capability to process large amount of data. Symmetric algorithms also provide very low overhead and hence, provide a higher speed of encryption and decryption. One such symmetric algorithm is AES. In N. Indira (2020) authors have developed light weight pro-active padding using AES circular shifting. AES encryption has been approved by NIST (National Institute of Standards and Technology) as mentioned in (Transitioning the Use of Cryptographic Algorithms and Key Lengths). In the proposed methodology AES with fixed block size of 128 bits and key size of 256 bits has been considered. As reviewed by National Security Agency (NSA), in order to maintain security of a top level, AES 256 bit keys should be used. In the end, the output is encoded to Base64, i.e., the ratio of output bytes and the input bytes will be 4:3. For Example, if the input is of length n bytes, the output will be represented by the equation 3.

$$Output = 4[\frac{1}{3}n] \tag{3}$$

The AES encryption used in this research took the group name and password of the data owner as salt and key for encrypting the data.

4 Design and Solution Development

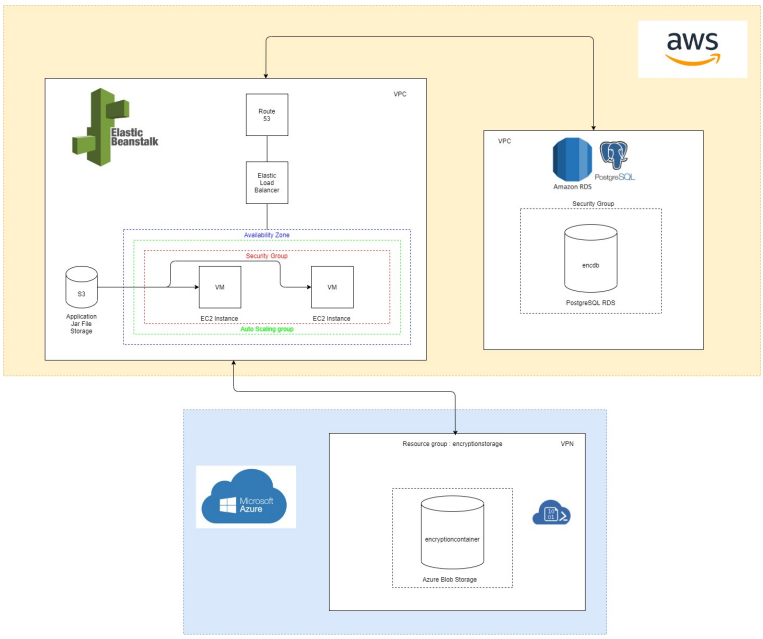

Figure 12: Cloud Architecture of multi-layered architecture

The cloud architecture of the research is shown in Figure 12. In this architecture, the cipher-text is stored in a different public cloud service provider i.e., microsoft azure. The application and the relational database of the application is in amazon web services (AWS). Having two different vendors will increase the trust of an organization to store the data in public cloud since, it would be difficult to decrypt the cipher-text from a different service provider. The application layer and the cipher-text storage layer are isolated from each other and both the layers are unaware about the logic.

Figure 13 gives an idea about the entity relationship of the database in Aws RDS. It can be seen from the diagram that, the spring boot framework based Java application only needs three tables to perform operation.

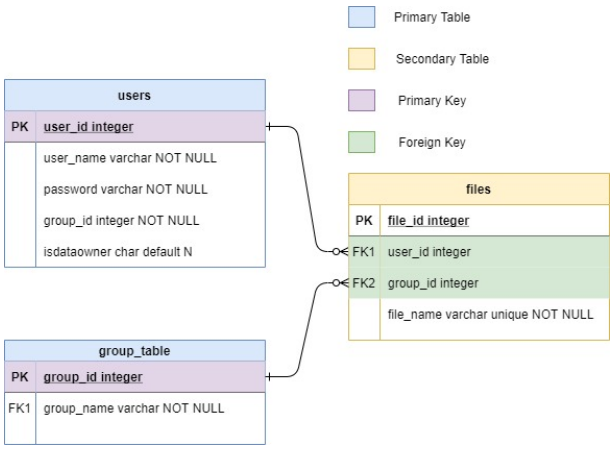

Figure 13: RDS Entity Relationship

The users table stores the details of the user, the group table stores the information i.e., group description and the id of group and the files table, stores the file names and its id with user id and group id as the foreign key.

5 Evaluation and Result Analysis

The evaluation process for this research consists of measuring the strength of the encrypted data and the security of keeping the data stored in the public cloud. The assessment is carried out using research methods for statistical analysis such as the Pearson correlation test and regression analysis. The tests conducted in this article, help us understand the benefit of using the research to store data in the public cloud. To understand the strength and orientation of the relationship among each algorithm, a simple and multiple linear regression model has been used.

The Pearson correlation coefficient helps us understand the correlation between two algorithms and whether it correlates positively or negatively. A positive or direct and negative or indirect correlation suggests that if the coefficient of the first algorithm increases, so does the coefficient of the second algorithm and vice versa.

5.1 Shannon Entropy

Shannon Entropy is the measure of average randomness or uncertainty of a variable's possible outcome. It can also be classified as the measure of strength of ciphertext.

File Size in Kb	Tree Traversal Entropy	Genetic Algorithm Entropy	AES Entropy
1 Kb	10.6925	55.6491	17.2059
10Kb	21500.3284	128948.5372	32268.0773
33Kb	74913.4593	451530.4254	112942.0139
50Kb	119437.0997	708435.8078	177159.2616
75Kb	182023.8137	1068046.2561	267123.7846
100Kb	237157.3683	1429980.2235	357586.3146
125Kb	300195.5227	1779579.2483	445133.4469
150Kb	361061.0814	2137625.6325	534658.3627
184Kb	444247.6791	2630739.7117	657975.4392
200Kb	478468.2102	2852647.6795	713472.1869
223Kb	534570.7477	3183418.1149	796071.9650
301Kb	730597.6478	4308037.7467	1077445.0366
377Kb	907804.9450	5400034.5287	1350569.4190
401Kb	970542.8392	5733751.1262	1433981.4144
500Kb	1184018.4828	7151495.3160	1788749.8988

Table 1: Shannon Entropy

The paper Gan and Learmonth (2015), Ayman Mousa (2013) and Yue Wu (2013) talks about various tests with entropy based measurements to check the randomness. Shannon Entropy is calculated using the Equation 4.

$$H(x) = -\sum_{n=1}^{n}[P(X_i) * log_b P(X_i)] \tag{4}$$

where, $P(X_i)$ is the probability of occurrence of a symbol in the variable, which is multiplied by logarithm of the probability with base b (i.e., base 2). The higher the entropy value, the higher the diversity and it will be more difficult to fetch the original details.

For Example, if we have a sequence of numbers such as 84326404235, then, probability of 8, P(8) will be 1/11. Similarly, probaility of 2 will be, P(2) = 2/11. Shannon Entropy, H = P(8)*$\log_2 P(8)$ + P(4)*$\log_2 P(4)$ + P(3)*$\log_2 P(3)$ + P(2)*$\log_2 P(2)$ + P(6)*$\log_2 P(6)$ + P(4)*$\log_2 P(4)$ + P(0)*$\log_2 P(0)$ + P(4)*$\log_2 P(4)$ + P(2)*$\log_2 P(2)$ + P(3)*$\log_2 P(3)$ + P(5)*$\log_2 P(5)$

Shannon entropy values of files of varying sizes are shown in Table 1, while shannon entropy probability values are shown in Table 2. Table 1 shows that, as file sizes increase, entropy values also increase for all three algorithms. We can also see that entropy values vary between algorithms for different sizes. This means that the algorithm makes a difference and provides diversity or randomity to the encrypted data. The level of diversity can be analyzed in more detail with the correlation test and regression tests.

Table 2 indicates the likelihood that the symbols will reappear. It can be seen that, the tree traversal probability is 0.0303 for 1Kb file whereas, as the size increase the probabilities becomes exponentially small. The exponential numbers rounded to the fourth decimal place give 0.0000, which is nearly negligible. A similar case is observed for the AES algorithm where the probability becomes almost negligible after the file size becomes higher than 300Kb. In contrast, a uniform probability is observed for Genetic Algorithm.

File Size in Kb	Tree Traversal Probability	Genetic Algorithm Probability	AES Probability
1 Kb	0.0303	0.2793	0.0132
10Kb	0.0000	0.2148	0.0000
33Kb	0.0000	0.2774	0.0155
50Kb	0.0000	0.2701	0.0156
75Kb	0.0000	0.2250	0.0157
100Kb	0.0000	0.2692	0.0156
125Kb	0.0000	0.2215	0.0000
150Kb	0.0000	0.2234	0.0156
184Kb	0.0000	0.2241	0.0157
200Kb	0.0000	0.2243	0.0000
223Kb	0.0000	0.2703	0.0156
301Kb	0.0000	0.2252	0.0000
377Kb	0.0000	0.2752	0.0000
401Kb	0.0000	0.2740	0.0000
500Kb	0.0000	0.2226	0.0000

Table 2: Probability of Symbol

5.2 Pearson's Correlation Test

The Correlation test is a test which helps us describe the strength and direction of the relationship between any two algorithms. A common statistical measure, Pearson's correlation will be used to measure the correlation coefficient in this paper. In addition, the relationship between the algorithms was studied using a scatter plot to fully understand the data. If the scatter plot generates a perfect straight line, it implies that the algorithms are highly correlated. The Pearson coefficient of correlation is the equation 5.

$$r = \frac{\sum[(X - \overline{X})(Y - \overline{Y})]}{(n-1)s_x s_y} \tag{5}$$

where $\overline{X}$ and $\overline{Y}$ are the x intercept or the mean of the values of x and y intercept or the mean of the values of y. s_x and s_y are the standard deviation of X and Y respectively.

5.2.1 Tree and Genetic Correlation

Figure 14 shows the Pearson's correlation coefficient value for the relation between the shannon entropy value of tree traversal algorithm and the genetic algorithm.

A significance level of 1% or 0.01 has been considered for this evaluation. It can be seen that both tree and genetic algorithm are mutually significant to each other at 0.01 significance level. This implies that a change in the value of tree shannon entropy brings about a change in the value of genetic shannon entropy and vice versa. The Pearson coefficient of tree and genetic shannon entropy is 1.000. This implies if shannon entropy value of tree increases by 1, the shannon entropy value of genetic algorithm will also increase by 1. This indicates tree and genetic shannon entropy values have a perfect and strong correlation. To further visualize the correlation, Figure 15 shows the linearity check of the correlation obtained. The scatter plot is completely linear without any deviation of the data points. The linear equation formed by the scatter plot is y=1.74E3+0.17*x.

Correlations

Descriptive Statistics

	Mean	Std. Deviation	N
SE of Tree	436436.6612	367115.5413	15
SE of Genetic	2597621.734	2193523.113	15

Correlations

		SE of Tree	SE of Genetic
SE of Tree	Pearson Correlation	1	1.000**
	Sig. (2-tailed)		.000
	N	15	15
SE of Genetic	Pearson Correlation	1.000**	1
	Sig. (2-tailed)	.000	
	N	15	15

**. Correlation is significant at the 0.01 level (2-tailed).

Figure 14: Tree and Genetic Output Correlation

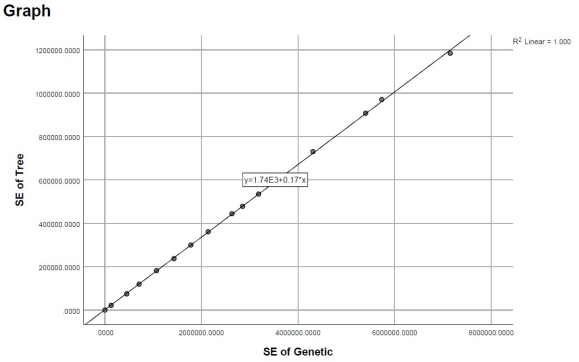

Figure 15: Correlation Scatter Plot

5.2.2 Genetic and AES Correlation

The Correlation test of Genetic and AES shannon entropy values provide us a similar outcome as that of Tree and Genetic Correlation as shown in Figure 16. Both Genetic and AES shannon entropy values are significant at 0.01 significance level.

Correlations

Descriptive Statistics

	Mean	Std. Deviation	N
SE of Genetic	2597621.734	2193523.113	15
SE of AES	649676.9218	548619.4819	15

Correlations

		SE of Genetic	SE of AES
SE of Genetic	Pearson Correlation	1	1.000**
	Sig. (2-tailed)		.000
	N	15	15
SE of AES	Pearson Correlation	1.000**	1
	Sig. (2-tailed)	.000	
	N	15	15

**. Correlation is significant at the 0.01 level (2-tailed).

Figure 16: Genetic and AES Output Correlation

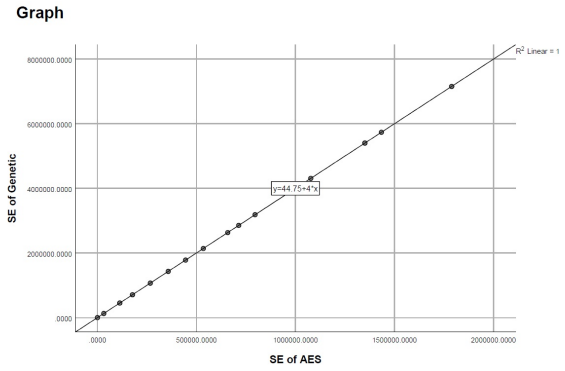

Figure 17: Correlation Scatter Plot

The Pearson's Correlation coefficient for both genetic and AES is 1.000 which implies if there is an increase in Genetic shannon entropy value by 1, there will be an increase in AES shannon entropy value by 1 as well and vice-versa. Both have a very strong and perfect correlation with each other. The visualization diagram is shown in Figure 17. It provides a straight linear line which almost no deviation. The equation formed in this case is y=44.75+4*x. where y is the y-intercept and x is the x-intercept.

5.2.3 Tree and AES Correlation

Figure 18 shows the shannon entropy value based correlation between Tree and AES. Both AES and Tree are significant enough at 1% significance level and suggests that a significant change in Tree results in a significant change in AES and vice-versa. The descriptive statistics in Figure 18 shows that the mean and standard deviation of both AES and Tree shannon entropy values. Similar to Genetic and AES correlation, the Pearson's correlation value is 1.000 for both tree and aes. This implies that both have

a strong and perfect correlation. The visualization in Figure 19 further confirms the correlation between Tree and AES. A linear line is formed with the data points of both the categories. The linear equation formed for this correlation is y=1.74E3+0.67*x.

Correlations

Descriptive Statistics

	Mean	Std. Deviation	N
SE of Tree	436436.6612	367115.5413	15
SE of AES	649676.9218	548619.4819	15

Correlations

		SE of Tree	SE of AES
SE of Tree	Pearson Correlation	1	1.000**
	Sig. (2-tailed)		.000
	N	15	15
SE of AES	Pearson Correlation	1.000**	1
	Sig. (2-tailed)	.000	
	N	15	15

**. Correlation is significant at the 0.01 level (2-tailed).

Figure 18: Tree and AES Output Correlation

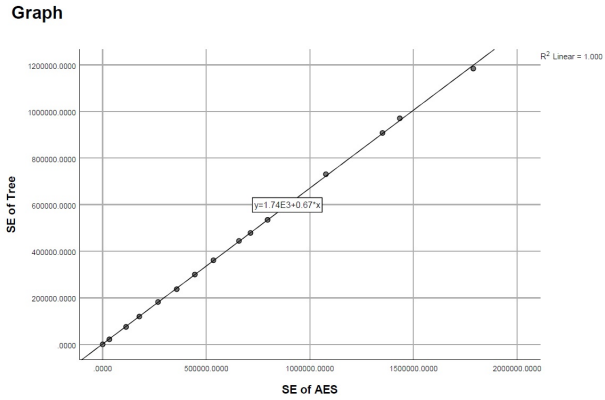

Figure 19: Correlation Scatter Plot

The Pearson's Correlation Test suggests that by implementing different layers to the genetic algorithm resulted in increase in the randomness or strength of the cipher-text. An increase by 1 at each layer results in increase by 1 in the next layer. All the layers have a very strong and perfect correlation between each other.

5.3 Regression Analysis

Regression analysis has been done to help build a model that estimates the strength of the algorithm using the obtained shannon entropy value of all the algorithms. In the research project, Tree algorithm's shannon entropy is responsible for Genetic algorithm's shannon entropy, Genetic algorithm's shannon entropy is responsible for AES algorithm's shannon entropy and a combined Tree and Genetic algorithm's shannon entropy is responsible for AES algorithm's shannon entropy. The Regression modelling helps us understand the causal relationship or the influence of one algorithm on the other. Interpreting the regression coefficient with the regression model an overall estimate of strength of the algorithm can be obtained. The regression model is represented using the Equation 6.

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n \tag{6}$$

where, (x_1,x_2,...) are the independent variables or the variables that are to be compared with respect to a dependent variable. In short, the measure of change of Y when x changes by one unit with the values of other independent variables as constant is known as multiple regression model. β_0, β_1, ..., are the regression coefficients which are chosen to reduce the squared residual. In short, the β values tell us to what degree each predictor affects the outcome if all other predictors are constant.

The model summary obtained from the multiple regression analysis provides three values such as R, R square and Adjusted R square. The R-value is the correlation between observed entropy values and the predicted values by the model. The R-square value is the proportion of variance taken into consideration by the model. The Adjusted

R-square is the modified version of R-square that has been adjusted for three algorithms used in the proposed research.

The regression analysis also provides an F-test using ANOVA which checks if the variance by the model is significantly greater than the error within the model. The F value helps us to know if the model is better in predicting values.

Since, from the correlation test in 5.2 it is clear that there is strong and perfect correlation between all the shannon entropy values, there is a possibility that an independent variable could be excluded from the regression test if all three are considered. The multiple regression model analysis has been done using 0.05 or 5% significance level.

5.3.1 Simple Linear Regression Modelling of Genetic Algorithm with Tree Entropy

Model Summary[b]

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	1.000[a]	1.000	1.000	33292.28618

a. Predictors: (Constant), SE of Tree

b. Dependent Variable: SE of Genetic

ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	6.735E+13	1	6.735E+13	60762.036	.000[b]
	Residual	1.441E+10	13	1108376319		
	Total	6.736E+13	14			

a. Dependent Variable: SE of Genetic

b. Predictors: (Constant), SE of Tree

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	-9817.619	13630.224		-.720	.484
	SE of Tree	5.974	.024	1.000	246.500	.000

a. Dependent Variable: SE of Genetic

Figure 20: Linear Regression Model of Genetic with Tree

Model Summary[b]

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	1.000[a]	1.000	1.000	55.6768944

a. Predictors: (Constant), SE of Genetic

b. Dependent Variable: SE of AES

ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	4.214E+12	1	4.214E+12	1359316150	.000[b]
	Residual	40298.915	13	3099.917		
	Total	4.214E+12	14			

a. Dependent Variable: SE of AES

b. Predictors: (Constant), SE of Genetic

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	-11.187	22.742		-.492	.631
	SE of Genetic	.250	.000	1.000	36868.905	.000

a. Dependent Variable: SE of AES

Figure 21: Linear Regression Model of AES with Genetic

Figure 20 shows the regression analysis done to predict the Genetic entropy values with tree entropy values. The Regression Model formed from the coefficients table in Figure 20 can be given as Equation 7.

$$Y = -9817.619 + (5.974)x_1 \tag{7}$$

where x_1 is the value of Tree entropy. The coefficient, β_1 value, is 5.974 which provides us the information that, as the Tree entropy value increases by 1, the value of Genetic entropy value increases by 5.974.

From the Model Summary table, in the Figure 20, it can be seen that, the correlation between the observed value and the predicted value by the model (i.e.,R value), R square value and the adjusted R square value is 1.000. This implies that, the tree entropy value accounts for 100% of the variation in Genetic entropy value.

The ANOVA table in the figure shows that the F value (60762.036) which is greater than the error of the model (33292.28618). Hence, this regression model is good enough in predicting the values. The Sig. value gives the significance of this value to the model. The significance value is 0.000 which is significant enough at 0.05 significance level.

The coefficient table also displays the significance level of 0.000 for the entropy value of Genetic. This provides us the information that, the predictor or the entropy value of Genetic is making a significant contribution to the model.

The analysis done in this case clearly provides us the information that, the Tree entropy value is responsible or the cause of Genetic Entropy value. Hence, Tree algorithm is significant enough to strengthen the Genetic algorithm and hence, is a good model.

5.3.2 Simple Linear Regression Modelling of AES with Genetic

Similar to the simple linear regression modelling of Genetic algorithm, linear regression modelling of AES can be done using Genetic entropy value. Figure 21 shows the linear regression modelling of AES with Genetic entropy value. The Regression equation formed using the analysis of the Coefficients table can be inferred in Equation 8.

$$Y = -11.187 + (0.250)x_1 \tag{8}$$

where x_1 is the value of Genetic entropy. The coefficient, β_1 value is 0.250 which tells us that, if the genetic entropy value increases by 1, the value of AES entropy value increases by 0.250.

The Model summary table gives us the value of R, R square and adjusted R square which is 1.000. This explains that, there is 100% variation in AES entropy value based on genetic entropy value.

The ANOVA table gives us the F value to be 1359316150 which is very big compared to the error value. Hence, the prediction of this model is good enough. Furthermore, F value is significant enough at 5% significance level as can be seen from the Sig. value which is 0.000.

The coefficient table shows a similar significance level of 0.000 as in linear regression modelling for Genetic entropy value. To infer from this analysis, the AES entropy value is caused by the Genetic entropy value. Hence, when the Genetic algorithm is done before the AES algorithm it strengthens the AES Output. Hence, this model is significant and turns out to be a good model.

5.3.3 Multiple Regression Modelling of AES with Tree and Genetic Entropy

It can be seen from the Linear Regression Modelling of Genetic Algorithm 7 and AES algorithm 8, that the model is very good in increasing the strength at each level. Since, AES entropy value derives from both the Tree and Genetic Algorithm, a multiple regression should be done to understand the effect of both the algorithms on AES entropy values. Figure 22 shows the multiple regression model of AES entropy value.

Using the Multiple Regression Model we can obtain the equation for the model as shown in Equation 9. It can be seen from the equation that, the AES entropy value can be predicted using both Tree and Genetic entropy values.

$$\text{Y = -1.597 + (-0.006) } x_1 \text{ + (0.251) } x_2$$

$$\implies Y = -1.597 - 0.006x_1 + 0.251x_2 \tag{9}$$

where x_1 is the value of Tree entropy and x_2 is the value of Genetic entropy. The value of tree coefficient, β_1 is 0.006 and the value of genetic coefficient, β_2 is 0.251. From

the equation 9 we can infer that, as the value of tree entropy value increases by 1, the value of aes increases by 0.006, whereas, as the value of genetic entropy value increases by 2, the value of the aes increases by 0.251.

Model Summary[b]

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	1.000[a]	1.000	1.000	48.2934965

a. Predictors: (Constant), SE of Genetic, SE of Tree

b. Dependent Variable: SE of AES

ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	4.214E+12	2	2.107E+12	903364849.1	.000[b]
	Residual	27987.142	12	2332.262		
	Total	4.214E+12	14			

a. Dependent Variable: SE of AES

b. Predictors: (Constant), SE of Genetic, SE of Tree

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients		
		B	Std. Error	Beta	t	Sig.
1	(Constant)	-1.597	20.163		-.079	.938
	SE of Tree	-.006	.002	-.004	-2.298	.040
	SE of Genetic	.251	.000	1.004	623.961	.000

a. Dependent Variable: SE of AES

Figure 22: Multiple Regression Model of AES

The Model summary table in the Figure 9 shows the R value, R square value and Adjusted R square value as 1.000 for the combined Tree and Genetic entropy value. From this we can conclude that the Tree and Genetic entropy value accounts for 100% of the variation in AES entropy value.

The ANOVA table shows the F value (903364849.1) to be greater than the standard error value (48.2934965) of the model. Hence, the prediction of this model is very good since the significance value of the model is 0.000 which is less than 0.05 significant level.

Furthermore, the coefficients table shows the significance level of 0.000 for Genetic entropy and 0.040 for Tree entropy. Both the entropies are less than 0.05 significant level and are significant enough for the multiple regression model.

To infer from this Regression model, the use of Tree and Genetic algorithm increases the entropy value of AES and makes the encryption strong.

5.4 Results Analysis

The evaluation from both Pearson's Correlation Test 5.2 and Regression Analysis 5.3 tells us that, the use of Tree, Genetic and AES algorithm is a good fit for encrypting the data. The practical evaluation from shannon entropy table also displays the increase of strength from tree entropy value to AES entropy value. There is a linear increase of strength at each layer of encryption. The Pearson's Correlation Test shows the strong and perfect correlation of each algorithm on each other. The Linear Regression Analysis and Multiple Regression Analysis displays the causal relationship i.e contribution of each algorithm to increase of strength of other algorithm.

6 Conclusion and Discussion

The encryption algorithm of this research is a layered approach to providing data security and confidentiality for public cloud storage using the tree traversal, genetic algorithm and AES algorithm. Practical evaluation and statistical analysis have shown that the strength of cryptography increases with each layer. Moreover, the relation between each layer is linear and has a high correlation between the others. The future work of this research would include improving the performance of the algorithm for big data.

References

Akashdeep Bhardwaj, GVB Subrahmanyamb, V. A. H. S. (2016). Security algorithms for cloud computing, *International Conference on Computational Modeling and Security (CMS 2016)* **85**: 535–542.

Andrew Rukhin, Juan Soto, J. N. M. S. E. B. S. L. M. L. M. V. D. B. A. H. J. D. S. V. (2010). A statistical test suite for random and pseudorandom number generators for cryptographic applications, *NIST* **800**(22).

Ayman Mousa, Osama S. Faragallah, S. E.-R. E. M. N. (2013). Security analysis of reverse encryption algorithm for databases, *International Journal of Computer Applications* **66**(14).

E. Vidhya, R. R. (2020). Key generation for dna cryptography using genetic operators and diffie-hellman key exchange algorithm, *International Journal of Mathematics and Computer Science* **15**(4): 1109–1115.

Elaine Barker, A. R. (2019). Transitioning the use of cryptographic algorithms and key lengths, *Computer Security* **Revision 2**: 33.

Gan, C. C. and Learmonth, G. (2015). Comparing entropy with tests for randomness as a measure of complexity in time series, *arXiv* p. 21.

Hambali Moshood Abiola, Gbolagade Morufat Damola, O. Y. A. (2020). Cloud security using least significant bit steganography and data encryption standard algorithm, *Computer Science and Telecommunications* **58**(1): 17–29.

Hebah H. O. Nasereddin, A. J. D. (2020). An encryption and decryption technique using genetic algorithm, *International Research Association for Talent Development and Excellence* **12**(3): 2311–2316.

Hossein Nematzadeh, Rasul Enayatifar, M. Y. M. L. P. G. J. (2020). Binary search tree image encryption with dna, *www.elsevier.com/locate/ijleo* **202**: 62–72.

Milap J Bhuva, S. S. (2019). Symmetric key-based authenticated encryption protocol, *INFORMATION SECURITY JOURNAL: A GLOBAL PERSPECTIVE* **28**(1): 35–45.

Mr.B.Bharathi, Mr.G.Manivasagam, D. K. (2017). Metrics for performance evaluation of encryption algorithms, *International Conference on Emerging Trends in Engineering* **6**: 62–72.

N. Indira, S. Rukmanidevi, A. K. (2020). Light weight proactive padding based crypto security system in distributed cloud environment, *International Journal of Computational Intelligence Systems* **13**(1): 36–43.

Norman H. Nie, Dale H. Bent, C. H. H. (2020). Ibm spss software.
URL: *https://www.ibm.com/analytics/spss-statistics-software*

Ren, C. and Xue, S. (2018). Asymmetric cryptographic algorithm for optical images and its safety, *Nonlinear Optics, Quantum Optics* **48**(4): 321–332.

Sarojini G, Vijayakumar.A, S. (2016). Trusted and reputed services using enhanced mutual trusted and reputed access control algorithm in cloud, *2nd International Conference on Intelligent Computing, Communication Convergence (ICCC-2016)* **92**: 506–512.

Sengupta, N. and Chinnasamy, R. (2015). Contriving hybrid descast algorithm for cloud security, *Eleventh International Multi-Conference on Information Processing-2015 (IMCIP-2015)* **54**: 47–56.

ShaluMall and Saroj, S. K. (2018). A new security framework for cloud data, *8th International Conference on Advances in Computing and Communication (ICACC-2018)* **143**: 765–775.

Venkatramana, K. and Geethab, K. (2019). Dynamic virtual cluster cloud security using hybrid steganographic image authentication algorithm, *COMPUTATIONAL INTELLIGENCE AND CAPSULE NETWORKS* **60**(3): 314–321.

Yue Wu, Yicong Zhou, G. S. S. A. J. P. N. P. N. (2013). Local shannon entropy measure with statistical tests for image randomness, *Information Sciences* **222**: 323–342.

Zhenyu Wen, Tao Lin, R. Y. R. R. A. R. C. L. and Xu, J. (2020). Ga-par: Dependable microservice orchestration framework for geo-distributed clouds, *IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS* **31**(1): 129–143.