Taylor & Francis
Taylor & Francis Group

Automatika

Journal for Control, Measurement, Electronics, Computing and Communications

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/taut20

A cryptographic method to have a secure communication of health care digital data into the cloud

K. Selvakumar & S. Lokesh

To cite this article: K. Selvakumar & S. Lokesh (2024) A cryptographic method to have a secure communication of health care digital data into the cloud, Automatika, 65:1, 373-386, DOI: 10.1080/00051144.2023.2301240

To link to this article: https://doi.org/10.1080/00051144.2023.2301240

Published online: 09 Jan 2024.

AUTOMATIKA
2024, VOL. 65, NO. 1, 373–386
https://doi.org/10.1080/00051144.2023.2301240

A cryptographic method to have a secure communication of health care digital data into the cloud

K. Selvakumar[a] and S. Lokesh [b]

[a]Research Scholar University College of Engineering, Nagercoil, India; [b]Department of Computer Science and Engineering, PSG Institute of Technology and Applied Research, Coimbatore, India

ABSTRACT

Cloud computing is a technology that holds great promise and has potential to revolutionize the healthcare sector. Many security and privacy issues are brought up by the cloud's centralization of data for both patients and healthcare professionals. There is a need for maintaining secrecy in communication in exchanging medical data between the sender and the receiver, which can be done by cryptography. This article presents a cryptographic algorithm (encryption and decryption) to have a secure communication of digital health care confidential data using DNA cryptography and Huffman coding. The interesting property is the cipher size obtained from our algorithm is equal to the size of the cipher obtained from the character set of given data. Security analysis is provided to show the security of data when stored and transmitted to the cloud. The cryptographic requirements, key space analysis, key and plain text sensitivity, sensitive score analysis, sensitivity and specificity, optimal threshold, randomness analysis, uniqueness of implementation, entropies of binary bits, DNA bases, DNA bases with Huffman code, Huffman encoded binary bits and cloud service provider's risk are analyzed. The method proposed is compared with other cryptographic methods and results that it is more secure and stronger than other methods.

ARTICLE HISTORY
Received 11 September 2023
Accepted 17 December 2023

KEYWORDS
Cryptography; communication; Huffman Algorithm; security; encryption; decryption; cryptosystem; symmetric key; asymmetric key

1. Introduction

Cloud computing has been gaining popularity recently and it has emerged as an essential one to all from the time of the pandemic covid-19. The adoption of cloud services around the world in all directions leads to the storage of huge data in the cloud and the corresponding cloud service. It is the right time to use DNA computing to store huge digital data and to offer secured cloud service to cloud users with full security.

The healthcare sector has embraced information and communication technology (ICT) and the internet to improve patient care, expedite procedures, and boost healthcare results. But as healthcare data is more digitalized, it becomes more important than ever to protect patient privacy and security. Many security and privacy issues are brought up by the cloud's centralization of data for both patients and healthcare professionals. Concerns about security, privacy, efficiency, and scalability are preventing cloud technology from being widely adopted as a result.

There is a need for maintaining secrecy in communication in exchanging medical data between the sender and the receiver. This can be done by cryptography. By encoding the messages one can maintain security in cryptography. Hashing functions, secret-key functions, and public key functions are the three categories of cryptographic functions. In public key cryptography, two keys are used. One key is used in secret-key cryptography.

In the cryptosystem, the sender of a message encrypts the message with the help of a key and sends it to the receiver over the internet. The receiver will decrypt the encrypted message with the help of getting the key from the sender. The message is referred to as plain text, the process of encoding is referred to as encryption, the encrypted message is referred to as cipher text, the process of decoding is referred to as decryption, and the text used to encrypt and decrypt is referred as the key. The entire system is referred to as a cryptosystem. There are two types of cryptosystems namely symmetric and asymmetric cryptosystems. In the case of a symmetric cryptosystem with the same key both the process of encryption and decryption will be performed. The well- known examples of symmetric cryptosystems are AES and DES. However, in the case of an asymmetric cryptosystem with two different keys, the process of encryption and decryption will be performed. One key, the public key for encryption,

CONTACT K. Selvakumar k_selvakumar10@yahoo.com Research Scholar University College of Engineering, Konam, Kanyakumari District, Nagercoil, Tamilnadu, India

This article was originally published with errors, which have now been corrected in the online version. Please see Correction (http://dx.doi.org/10.1080/00051144.2024.2307179)

and another key, the private key for decryption. A well-known example of an asymmetric cryptosystem is RSA.

Data compression reduces the storage space required to store data by changing its format. The security of digital data is a standing issue because digital data are connected to the internet. The need of all is to secure data during the storage and transmission of data. In this article, a method using variable key lengths in the algorithm will prevent guessing the key length by an attacker.

1.1. Works related to this article

Qi and Fan [1]; Qi et al. [2]; Qi and Qi [1]; Qi et al. [3], represent the sequences from Huffman coding by graphs to study long DNA sequences. In Qi et al. [3], applied graphs to the sequences to protect the data. Marty et al. [4] protected the data by storing it. Kazuo et al. [5] protected the data by a one-way function. Auto et al. [6] protected the data with a novel DNA computing method. Ibrahim et al. [7], protected the data with a DNA cryptographic algorithm. Hossain et al. [8] protected the data by a novel method. Maria et al. [9] applied a method to reduce the computation time. Krishna Gopal et al. [10] imposed a constraint to reduce the error occurring during the conversion of binary into DNA strings. The algebraic attacks on stream ciphers, cause a lot of impacts and so using threshold functions, the algebraic immunity is studied by Pierrick [11]. Ana and Pantelimon applied probabilistic methods to estimate the nonlinearity of Boolean functions [12]. In elliptic curve cryptography, a study on the linear congruential generator is made by Jaime [13]. Wrya et al. gave encoding and decoding algorithms using matrices in [14].

Gary Doeblien [15] identified the difficulties in ensuring data security and privacy in digital healthcare as well as the significance of cyber security for e-Health. Applications on cloud computing can be secured by applying a lightweight cryptographic algorithm, as demonstrated by Fursan Thabit et al. [16]. In Sivan and Ahmad Zukarnain [17], discuss the growing need for cloud computing, what it is, the challenges and opportunities it presents, and how healthcare organizations should set up and prepare their defenses before deploying the newest, most advanced service model. Yazan Al-Issa et al.'s study [18] examined various cloud security and privacy issues as well as the application of cloud computing in the healthcare sector. By using a Password-Based Key Derivation Function (PBKDF2) to secure enhanced versions of secure hash fixed-based output cryptographic algorithms (SHA-512), Dhanalakshmii and Victo Sudha George [19] devised a way to secure data for E-Health applications in cloud environments. This helps to secure patient data in e-health cloud environments. A unique privacy model for Electronic Health Records (EHR) systems was proposed by Nowrozy et al. [20], using machine learning (ML) techniques and a conceptual privacy ontology. A systematic study on privacy in electronic health records was carried out by Rodrigo et al. [21].

Cloud-based cryptography and compressions discussed in Abed et al. [22], Ahmad and Shin [23], Ailenberg and Rotstein [24], Alsaffar et al. [25], Alsaffar [26], Alhija et al. [27], Cao et al. [28], Chen et al. [29], De Silva and Ganegoda [30], Dong et al. [31], Doricchi et al. [32], Goldman et al. [33], Golin et al. [34], Jameel and Fadhel [35], Mehedi et al. [36], Mehedi et al. [36], Amanullah et al. [37], Bhattacharyya et al. [38], Li et al. [39], Altarawneh [40], and Rupa and Shah [41] are different from our work in this article.

1.2. Main results

Both encryption and decryption algorithms are presented in this article. The process is illustrated with an example. The performance of the method concerning encryption and compression is discussed. The cryptographic requirements, key space analysis, key and plain text sensitivity, sensitive score analysis, sensitivity and specificity, optimal threshold, randomness analysis, uniqueness of implementation, entropies of binary bits, DNA bases, DNA bases with Huffman code, and Huffman encoded binary bits, and cloud service provider's risk are analyzed.

The transmission entropy of the cipher reached the maximum value since $p(0) = p(1) = 0.5$ to the sequence of the cipher. The size of the data is 32 bits and the cipher is 6 bits so 81.25% of the storage space be saved by adopting this method. Security analysis is given to show its performance. The randomness of the cryptographic method is calculated as 100%. Hence, it became difficult for the adversary to guess the original data. Using arithmetic and geometric means the uniqueness of the implementation of our method is calculated as 0.5 and it is observed that it reached 0.5 the ideal value for uniqueness, which leads to high security. The method of this article is compared with other cryptographic methods to show it is more secure and stronger than other methods.

> The interesting feature of our method is the size of the cipher obtained from our algorithm is equal to the size of the cipher obtained from the characters of the given data. And, due to DNA computing and Huffman coding no degradation and no data loss. And hence no risk to the cloud service providers. The way of using variable key lengths in the algorithm will prevent guessing the key length by an attacker.

1.3. Construction of the article

In section 2, a cryptographic method is introduced in the article. The architecture of the method is also given. The components of the architecture such as cloud

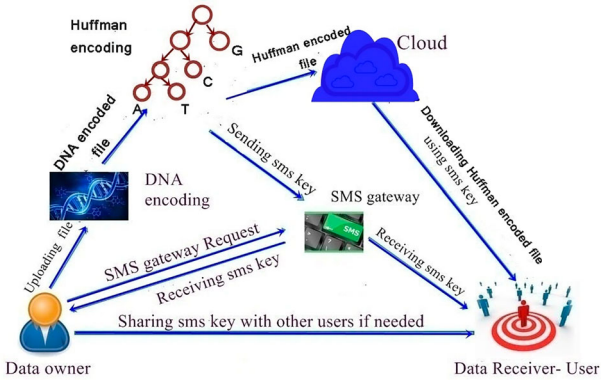

Figure 1. Proposed cloud architecture.

service provider, DNA coding, and Huffman coding are discussed. In section 3, the encryption algorithm with the flow diagram of the process is given. The encryption process is illustrated in section 3 using an example. In section 4, the decryption algorithm with the flow diagram of the process is given. Section 5, provides the security analysis of the method. In Section 6, a comparative study of methods is given. The conclusion and future research plans are given in section 7. Finally, declarations and references are given at the end of this article.

2. The proposed system

In this section, an architecture of a DNA coding and Huffman coding system (Refer to Figure 1.) is proposed. This technique uses variable force lengths. Its length cannot be guessed by the attacker. Since the security is maintained in this method compared to traditional methods cloud users can adopt this method. On adopting this method there will be an increase in industry growth and production.

To store data, including files, business data, films, and photos, cloud storage uses remote servers. Through an internet connection, users upload data to servers, where it is stored on a virtual machine on a physical server. Cloud companies commonly distribute data around several virtual machines in global data centres to ensure availability and redundancy. The cloud provider will spin up extra virtual computers to manage the load as storage requirements increase. Through an internet connection and software such as a web portal, browser, or mobile app via an application programming interface (API), users can access data in Cloud Storage.

2.1. DNA Coding

In the case of DNA computing, the storage space of data will be reduced since the two binary bits are stored in a mono nucleotide {A, T, C, G} of a DNA molecule (Refer to Figure 2). And, the security of the data will also be maintained due to the complex nature of DNA sequence.

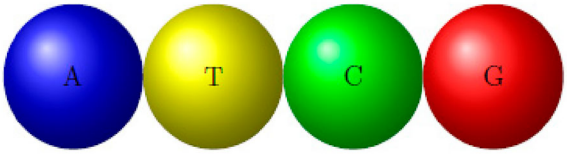

Figure 2. DNA molecules.

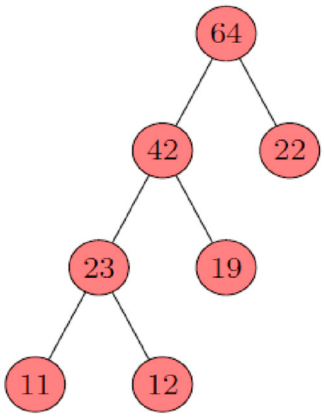

Figure 3. Huffman binary tree.

2.2. Huffman coding

For lossless data compression, Huffman coding is an entropy encoding algorithm that was coded by Huffman [42]. It is an optimal compression algorithm to compress the data using the frequency of individual letters [43,44]. The Huffman binary tree is given in (Refer to Figure 3).

3. Encryption algorithm

In this section, an encryption process (Refer to Figure 4), and an algorithm (Refer Algorithm 1) are presented. A text message is encrypted using this algorithm. And, a cipher is also generated from the characters of the data. The resulting ciphertext contains information that provides enhanced protection against intruder attacks.

The encryption process is given in the form of a flow diagram. First, take a piece of data and convert it into hexadecimal data, then the hexadecimal data into binary data. Using XOR operation convert the resultant binary data into another binary data using Key 1. Now convert this binary data into DNA data using Key 2. Using XOR operation convert the resultant DNA data into another DNA data using Key 3. Now convert the DNA data into Huffman data using Key 4.. Using the XOR operation convert the resultant Huffman data into another Huffman data using Key 5. Repeat this process till the cipher size is equal to the cipher obtained from the characters of the given data. Finally, replacing binary bits 0 by u and 1 by v.

When the data is uploaded to the cloud, an SMS or electronic mail key is sent to the The mobile phone of

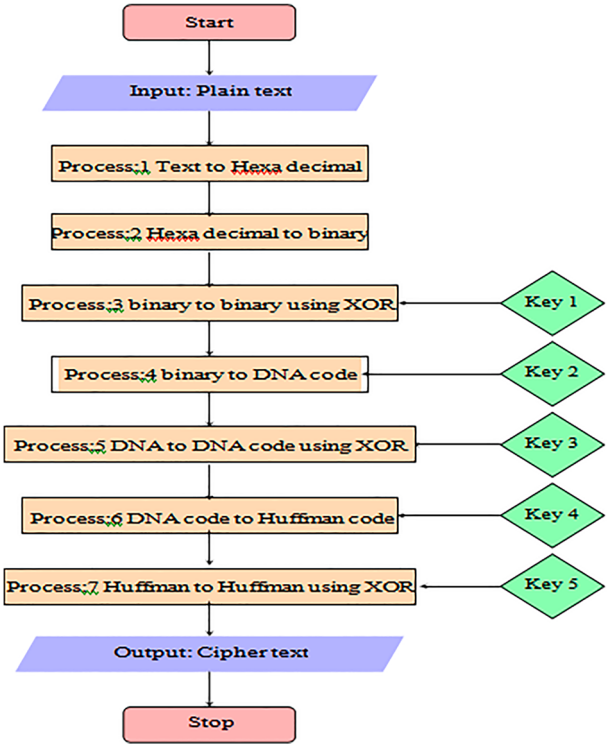

Figure 4. Encryption process.

Algorithm 1 Encryption algorithm

Require: Original plaintext
Ensure: Secured Cipher text
1: Select an original plaintext to encrypt.
2: Generate Hexa decimal code to the original plaintext (Level −1)
3: Generate binary sequence to the Hexa decimal code (Level −2)
4: Generate binary sequence by applying XOR using a key, a unit sequence of the same length (Level −3)
5: Generate DNA sequence from binary sequence using the key (Level −4)
6: Generate DNA sequence by applying XOR using a key, it is a sequence with a DNA base C of the same length (Level −5)
7: Generate the frequency of the DNA bases to the generated DNA code (Level −6)
8: Generate the Huffman tree to the generated frequency of the DNA bases (Level −7)
9: Generate Huffman tree numbering left and right edges by 0 and 1. (Level −8)
10: Assign binary bits to DNA bases from the Huffman tree(level −9)
11: Generate Huffman binary sequence to the DNA sequence. (level −10)
12: Generate binary sequence by applying XOR using a key, a unit sequence of the same length (Level −11)
13: Repeat steps 7–11 till the cipher size is equal to the cipher got from the characters of the given data.
14: Generate a sequence by replacing binary bits 0 by u and 1 by v. (Level −12)
15: The resultant from the previous step is a secured cipher text, an encrypted and compressed one. It will be put in the cloud. (Level −13)

the data owner and the same key can only be used to access the data from the cloud.

Cloud computing employs encryption, access control, constant monitoring, and regular backup and recovery procedures as well as other technologies, processes, and policies to safeguard the cloud environment. Organizations may make sure that their data is safe and secure in the cloud by putting these precautions in place.

Table 1. Frequency and probability table of di-nucleotides.

DNA bases	TC	AT	TA	TG	Length of DNA sequence
Frequency	1	2	2	3	8
Probabilities	0.125	0.25	0.25	0.375	Total = 1

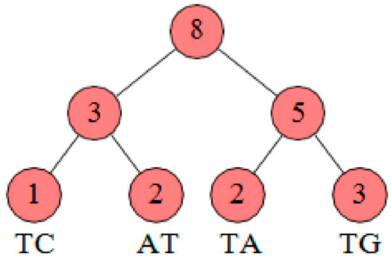

Figure 5. Huffman binary tree.

Table 2. Huffman code table of di-nucleotides.

DNA bases	TC	AT	TA	TG	Total bits
Huffman code	00	01	10	11	8

3.1. Encryption result

In this section, we provide an example that will be useful for security analysis. data Using the algorithm the following steps are given for an Example.

Step 1 Original plain text to be encrypted.
data

Step 2 Conversion of the data into the Hexa decimal form (Level −1) gives 64 61 74 61

Step 3 Conversion of the Hexa decimal form into binary form (Level −2) gives 01100100 01100001 01110100 01100001

Step 4 Conversion of binary sequence into another binary sequence by applying XOR using the key 11111111111111111111111111111111 of the same length (Level −3) gives 10011011 10011110 10001011 10011110

Step 5 Conversion of binary sequence in step 4 into DNA sequence (Level −4) gives GTGCGTCG-GAGCGTCG

Step 6 Conversion of DNA sequence into another DNA sequence by applying XOR using a key CCCCCC-CCCCCCCCCC of the same length (Level −5) gives TGTATGATTCTATGAT

Step 7 Frequency and probability of di-nucleotides (Level-6) is given in Table 1.

Step 8 Creation of the Huffman tree (Level −7) is given in Figure 5.

Step 9 Labelling the left and right edges by 0 and 1 in the Huffman tree and the end nodes by di-nucleotides (Level −8) the resultant is given in Figure 6.

Step 10 Assigning the binary bits to the di-nucleotides along the branches of the Huffman tree, the required bits to encode the di-nucleotides are given in Table 2.

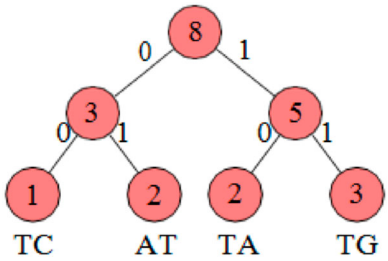

Figure 6. Labelled Huffman binary tree.

Table 3. Frequency and probabilities of {NE, NW, SE}.

DNA bases	NE	NW	SE	Length of Huffman sequence
Frequency	1	2	1	4
Probabilities	0.25	0.5	0.25	Total = 1

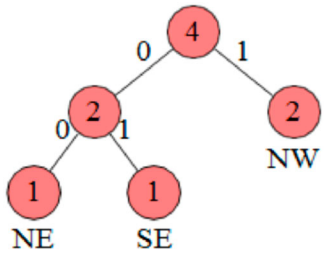

Figure 7. Huffman binary tree.

Table 4. Huffman-coded table of {NE, NW, SE}.

DNA bases	NE	NW	SE	Length of Huffman sequence
Frequency	1	2	1	4
Probabilities	0.25	0.5	0.25	Total = 1
Huffman code	00	1	01	

Step 11 Conversion of di-nucleotide into Huffman binary sequence (Level −10) gives 11 10 11 01 00 10 11 01

Step 12 Conversion of the sequence in step 11 by a key of the same length of the form 1111111111111111 using XOR (Level −11) gives 00 01 00 10 11 01 00 10

Step 13 Conversion of the sequence in step 12 into an un-guessable form using the key (Level −11a) gives NENWSENW

Step 14 Frequency and probability of {*NE, NW, SE*}. (Level −11b), is in Table 3.

Step 15 Creation of the Huffman tree (Level −11c), is given in Figure 7.

Step 16 The frequency and probability of {*NE, NW, SE*}. (Level −11d), is in Table 4.

Step 17 Conversion of the sequence in the previous step using Huffman code using the key (Level −11e) gives 00 1 01 1

Step 18 Conversion of the sequence in the previous step using the operation XOR using the key 111111 (Level −11f) gives 11 0 10 0

Step 19 Conversion of the sequence of the previous step by replacing 0 with u and 1 with v, a key (Level −12) gives vv u vu u

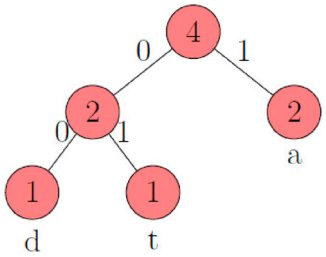

Figure 8. Huffman tree of the characters {*d, a, t*}.

Table 5. Frequency table of the characters {d, a, t}.

Characters	d	a	t	Length of sequence
Frequency	1	2	1	4
Probabilities	0.25	0.5	0.25	Total = 1
Huffman code	00	1	01	cipher 001011

Step 20 Put the resultant cipher text from the previous step in the cloud (Level −13).

The cipher text from the previous step is an encrypted one. It will be put in the cloud.

3.2. *Huffman code for an Example using characters*

On applying the Huffman coding to compress the data of an Example, one must take the characters {*d, a, t*}, the frequency distribution table is (Refer to Table 5) and the corresponding Huffman tree is (Refer to Figure 8)

From the Huffman tree, the cipher is 001011. The cipher obtained from the encryption algorithm of the proposed method is 001011 in step 17 of an Example. and it is the same as the cipher 001011 got directly using the characters in an Example. Again, the length of the cipher obtained using the algorithm in step 17 is 6 and the length of the cipher got directly using characters is 6. The compression achieved is (32-6)/32 = 26/32. And the percentage of compression is 81.25%. The data saves the storage space of 32-6 = 26 bits. Using the method in this article only 6 bits of space is enough to store data of size 32 bits

4. Decryption algorithm

In this section, a decryption process (Refer to Figure 9), and an algorithm (Refer Algorithm 2) are presented. A text message is decrypted using this algorithm. When the cipher, encrypted data is downloaded by the end user from the cloud, using an e-mail key or SMS key received at the computer/mobile p26.32hone from the data owner, the end user can decrypt and get the original data and can use and store it in their computer.

Using the Algorithm 2, the following steps are given for an Example.

Step 1 The cipher text is downloaded from the cloud (Level −13).

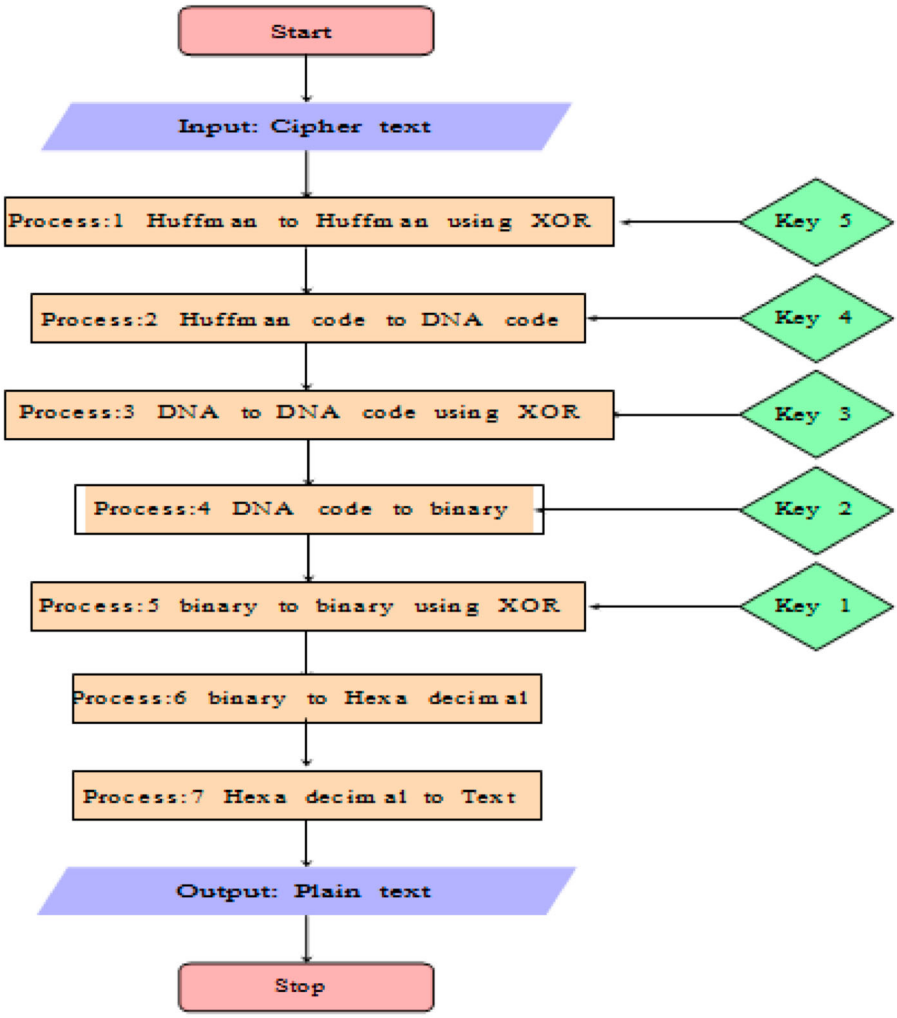

Figure 9. Decryption process.

Algorithm 2 Decryption algorithm

Require: Secured Cipher text
Ensure: Original plaintext
1: Select the Cipher text to decrypt.
2: Generate a sequence by replacing u by 0 and v by 1 from the cheer. (Level 1)
3: Generate binary sequence by applying XOR using a key, a unit sequence of the same length (Level −11)
4: Generate the Huffman code sequence from the previous sequence. (level 2)
5: Assign to the DNA bases the Huffman code (level 3)
6: Generate Huffman tree numbering left and right edges by 0 and 1. (Level −4)
7: Generate frequency of the DNA bases from the Huffman tree (Level 5)
8: Generate DNA code from the frequency of the DNA bases (Level 6)
9: Generate DNA sequence from the previous step (Level 7)
10: Repeat the above process until you get a binary sequence of length that is equal to the length of the given data. (Level 8)
11: Generate DNA sequence using XOR with the DNA sequence of the same length with the base C only from the previous step (Level 9)
12: Generate binary sequence from DNA sequence (Level 10)
13: Generate binary sequence using XOR with a unit sequence of the same length as the binary sequence got from the previous step (Level 11)
14: Generate the Hexa decimal code from the binary sequence (Level 12)
15: Generate the original plaintext from the Hexa decimal code (Level 13)
16: The resultant from the previous step is original data, a decrypted one. It will be put in the computer of the end user in any part of the world.

Table 6. Frequency table of {NE, NW, SE} from step 5.

DNA bases	NE	NW	SE	Length of Huffman sequence
Frequency	1	2	1	4
Probabilities	0.25	0.5	0.25	Total = 1
Huffman code	00	1	01	

The cipher text is an encrypted one and stored on your computer.

Step 2 Downloaded cipher text from the cloud will be of the form (Level −12).

vv u vu u

Step 3 Conversion of sequence in step 2 into a binary form by replacing u by 0 and v by 1 (Level −11a) gives 11 0 10 0

Step 4 Conversion of sequence in step 3 into a binary form by using XOR with a binary unit sequence 111111 (Level −11b) gives 00 1 01 1

Step 5 Generate a DNA sequence from the Huffman binary code (Level −11c) gives NE NW SE NW

Step 6 Frequency and probability table of {*NE, NW, SE*}. (Level −11d), is Table 6

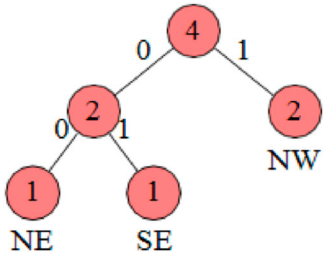

Figure 10. Huffman tree of the step 6.

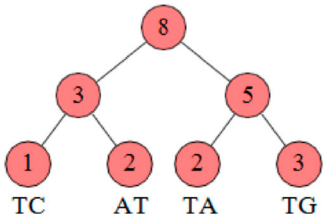

Figure 11. Huffman tree of the step 9.

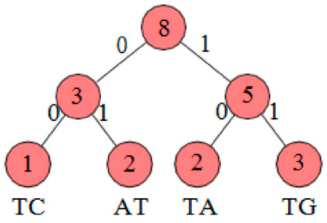

Figure 12. Huffman tree of the step 10.

Table 7. Frequency table of step 11.

DNA bases	TC	AT	TA	TG	Total bits
Huffman code	00	01	10	11	8

Step 7 Creation of the Huffman tree (Level −11e) is given in Figure 10.

Step 8 Conversion of the sequence in step 5 by using a key (Level −11f) gives 00 01 00 10 11 01 00 10

Step 9 Conversion of the sequence in step 8 by a key of the same length of the form 1111111111111111 using XOR (Level −10) gives 11 10 11 01 00 10 11 01

Step 10 Creation of the Huffman tree (Level −9), is given in Figure 11.

Step 11 Labelling the left and right edges by 0 and 1 in the Huffman tree and the end nodes by di-nucleotides (Level −8), the resultant Figure is Figure 12.

Step 12 Assigning binary bits to the di-nucleotides along the branches of the Huffman tree, the required bits to encode the di-nucleotides are given in Table 7.

Step 13 Generation of DNA sequence using the key (Level −6) gives TGTATGATTCTATGAT

Step 14 Conversion of DNA sequence into another DNA sequence by applying XOR using a key CCCCCC-CCCCCCCCCCC of the same length. (Level −5) gives GTGCGTCGGAGCGTCG

Step 15 Conversion of DNA into a binary sequence using the key (Level −4) gives 10011011 10011110 10001011 10011110

Step 16 Conversion of binary sequence into another binary sequence by applying XOR using the key 11111111111111111111111111111111 of the same length. (Level −3) gives 01100100 01100001 01110100 01100001

Step 17 Conversion of the binary form into Hexa decimal form into (Level −2) gives 64 61 74 61

Step 18 Conversion of the Hexa decimal form into an original plain text. (Level −1) data

Following the above steps, the original text can be decrypted from this method.

5. The security analysis of the method

In the following, to study the performance of the method we analyze the security of the data using the method of this article.

5.1. Cryptographic requirements

From the cipher output, an attacker should not guess the key and input data, to that extent a method should produce a cipher output. If a single bit of key is altered it should give different input, to that extent cipher should depend on the key and the input.

5.2. Key space analysis

The security strength of this cryptographic method is 230 bits. And so the method can face brute-force attacks.

5.3. Key and plaintext sensitivity

The ideal value of the bit change percentage is 50%. Results from an Example, the bitchange percentage was 81.25% greater than the ideal value 50%. An attacker can not find any properties of the plaintext or the key used when the cipher text is obtained since the algorithm is strong.

5.4. Security score analysis of the proposed method

Assign a score of +1 for the bit symbol 1 and assign a score of for the bit symbol 0. Then the security score of a binary sequence is given by.

$$\text{Security score} = \frac{\textit{sum of the scores of all bits}}{\textit{length of binary sequence}}$$

If the security score is in the interval [−1,1], then the data is secured. If the security value in an algorithm is increased from a negative value to a positive value, then the cryptographic method is more secure. The security

value should not be greater than 1 and less than −1. That is, in general, $-1 \leq$ Security score ≤ 1 The security score of a binary sequence will be of the form,

$$\text{Security score} = \begin{cases} 1, & \text{if number of 0 bits} = 0, \\ 0.5, & \text{if number of 1 bits} = \frac{3}{4} \text{ length of binary sequence}, \\ 0, & \text{if number of 1 bits} = \text{number of 0 bits}, \\ -0.5, & \text{if number of 1 bits} = \frac{1}{4} \text{ length of binary sequence} \\ -1, & \text{if the number of i bits} = 0. \end{cases}$$

For Example 1., the value of the security score is given by

$$\text{Security score} = \begin{cases} -0.1875, & \text{for the binary sequence of origial data}, \\ -0.0025 & \text{for the first DNA encoded sequence}, \\ 0 & \text{for the final Huffman coded binary sequence} \end{cases}$$

The security score has increased from −0.1875 to 0. This indicates the cryptographic method is secured and robust in performing DNA encryption and then the Huffman compression. Security score predicts that the method successfully secures digital data.

5.5. The Sensitivity and Specificity of the method

The sensitivity and Specificity of the method are discussed in this section using data threshold. Data thresholds are applied to prevent anyone from viewing a report present in data. The higher the threshold higher the precision.

The optimal threshold is defined as the ratio of the number of 1 bit divided by the sum of the number of 1 bit and the number of 0 bits from the binary sequence. If the threshold value of the cipher is greater than the threshold value of the original data then the cipher is more sensitive and specific. And, so no one can view a report present in the data.

For Example, the binary sequence of length 32 of the original data contains 13 numbers of 1 bit and 19 numbers of 0 bits. The threshold value of the original data is

$$\text{Optimal Threshold} = \frac{13}{32} = 0.40625$$

And, the Huffman-coded binary sequence of length 6 of the cipher of the original data contains, 3 numbers of 1 bit and 3 numbers of 0 bits. The threshold value of the cipher of the original data is given by

$$\text{Optimal Threshold} = \frac{3}{6} = 0.5.$$

The threshold value of the cipher is 0.5 and it is greater than the threshold value of the input data which is 0.40625. This shows that the cryptographic method prevents anyone from viewing a report present in the data. The optimal threshold leads to the highest sum of sensitivity and specificity of the proposed method.

5.6. Randomness analysis of the method

The entropy of the method is used in this section to perform the randomness analysis of the method. The randomness is given by

$$\text{Randomness} = -\sum_{i=1}^{2} P_i log_2 P_i$$

where P_i, i = 1(1)2 refers to the proportions of the binary bits 1 and 0 of the encoded Huffman binary sequence of the data. If the randomness is close to 50%, it will become difficult for the adversary to guess the original data.

For Example, 1., the randomness of the Huffman encoded binary sequence is equal to 1.0 bits per symbol. That is, the randomness of the cryptographic method is 100%. Hence, it became difficult for the adversary to guess the original data.

5.7. Uniqueness

In this section, we have to find the uniqueness of our implementation. The ideal value of uniqueness is 50%, and it means half of the bits in the final output are different. The uniqueness is calculated from the arithmetic /geometric mean of R_1 and R_2 where $R1$ is a ratio between the number of 1 bit in a binary sequence and the length of the binary sequence and R_2 is a ratio between the number of 0 bits in a binary sequence and Length of the binary sequence.

For Example, $R_1 = \frac{13}{32}$, and, $R_2 = \frac{19}{32}$, for the original data. And for Example, $R_1 = \frac{3}{6}$, and $R_2 = \frac{3}{6}$, for the cipher of the data. The uniqueness value is

$$\text{Uniqueness} = \begin{cases} 0.5, & \text{arithmetic nean of } (R_1, R_2) \text{ of original data} \\ 0.5, & \text{arithmetic mean of } (R_1, R_2) \text{ of the cipher.} \end{cases}$$

On taking the arithmetic mean of R_1 and R_2, one can observe value remains 0.5 fixed. That is, the ideal value

of uniqueness is 50%. Again, the unique value is

$$\textit{Uniqueness} = \begin{cases} 0.4911323, & \textit{geometic nean of } (R_1, R_2) \textit{ of original data} \\ 0.5, & \textit{geometic mean of } (R_1, R_2) \textit{ of the cipher.} \end{cases}$$

On taking the geometric mean of R_1 and R_2, one can observe the value 0.5 reached the ideal value of uniqueness 0.5. This predicts that the cryptographic method is unique and leads towards higher security.

5.8. Entropy of the binary bits

Entropy is the measure of uncertainty in bits and this concept was introduced by Shannon in [45,46]. The uncertainty of the cipher is the number of plaintext bits that must be recovered from scrambled cipher text to get the message back, and this is measured via entropy. The entropy of a variable is the weighted average of optimal bit representation size such as the average size of an optically encoded message.

The entropy H of the binary bits X = {0, 1} in the binary sequence B of the original data with two symbols 0 and 1 due to the event is given by Jones and Mewhort [45]; Othman et al. [46]; Shannon [47]; Shannon [47],

$$\mathrm{H} = -\sum_{i=1}^{2} P_i log_2 P_i$$

where P_i, i = 1(1)2 refers to the probabilities of binary bits 0 and 1 of the binary sequence of the original data B. Entropy H can also be defined as

$$\mathrm{H} = -log_2[(P(0).^P 1)(P(1).^P 2)]$$

where P_i, i = 1(1)2 refers to the probabilities of binary bits 0 and 1 of the binary sequence of the original data.

This means, that the higher the probability of an event less the uncertainty. The highest uncertainty is only achieved when the values are equally distributed. The probability of an event ranges from 0 to 1 and the entropy can range from 0 to 1. Figure 13 gives an insight into the entropy of event encryption using $X = \{0, 1\}$ where two outcomes are considered.

(1) The value of entropy is 0 for both the least and highest probability, which proves that if the probability of occurrence is 0, an event entropy will be 0, indicating that the event will never happen. Similarly, if the probability of an event is 1 means this event will always happen against the entropy is 0 because, in this scenario, there is no uncertainty about the information.
(2) The entropy of the system is maximum "1" when the probability is "1/2 = 0.5". This indicates that all events have the same chance to occur. If the probability increases from "0.5" then entropy decreases and similarly, if the probability decreases the entropy also decreases because in a former event is less likely to occur whereas in later the event is more likely to occur. For a system where the number of events increases the entropy also increases, for example, it has two possible outcomes, and the probability range from 0 to 1, its distribution differs. Each event has equal probability $p = 1/2$ only then maximum entropy will be achieved (Refer to Figure 15a). Here the entropy reaches a maximum value which is 1.

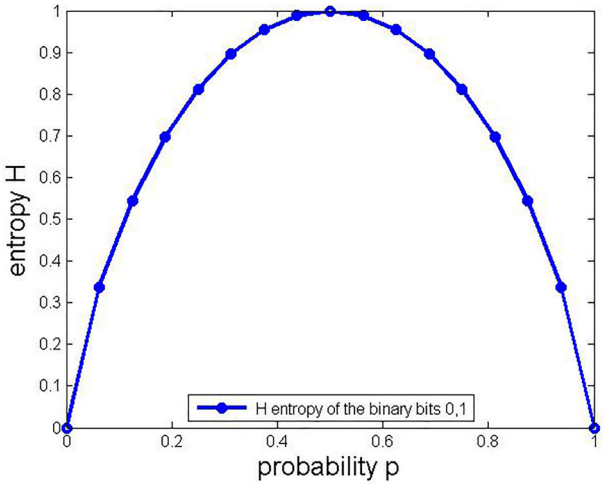

Figure 13. Entropy curve of the binary bits 0 and 1.

Table 8. Entropy calculation table.

Binary bases	0	1	Total
Frequency	19	13	32 bits
Probabilities P_i	0.59375	0.40625	1
Binary codes	0	1	
Number of bits n_i	1	1	
$P_i n_i$	0.59375	0.40625	Average length = 1
P_i.log P_i	−0.134422849	−0.158927691	H = 0.974489403

For Example, the entropy of the binary sequence of the original data is equal to 0.974489403 bits per symbol (Table 8).

5.9. Entropy of the DNA bases

The entropy H of the DNA bases {*TC, AT, TA, TG*} is given by

$$\mathrm{H} = -\sum_{i=1}^{2} P_i log_2 P_i$$

where P_i, i = 1(1)4 refers to the probabilities of DNA bases {*TC, AT, TA, TG*}. For Example, the entropy of the DNA four bases is equal to 1.913629062 bits per symbol. The maximum entropy value from the entropy curve is 2 (Refer to Figure 14)

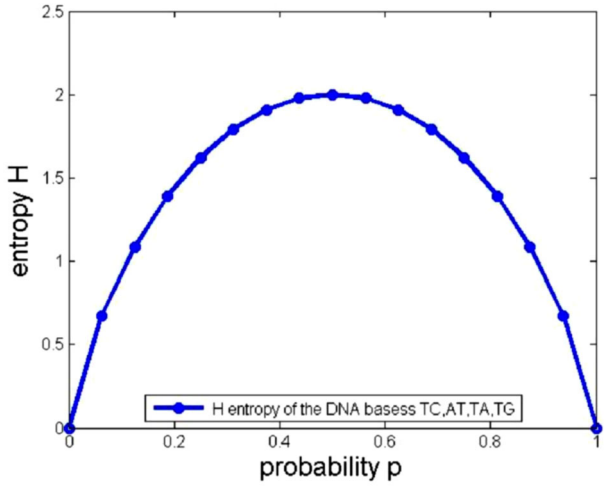

Figure 14. Entropy curve of the DNA bases TC, AT, TA, TG.

As per the expectations to have strong encryption, the following conditions are the requirements, namely,

(1) The probability value of encoding with DNA bases should be lesser than encoding the original data by binary bits.
(2) The average length of the binary tree of the DNA bases should be greater than the average length of the binary tree of the binary bits.
(3) The entropy value of the DNA bases should be greater than the entropy value of binary bits.
(4) The maximum entropy value of the DNA bases should be greater than the maximum entropy value of binary bits.

All these requirements are fulfilled in the method of this article.

5.10. Entropy of the DNA bases with Huffman code

When compressing an ideal gas volume, the entropy increases since the molecules collide more times per second with each other. Similarly, as the molecules have more room to move, the entropy decreases when expanding an ideal gas. The entropy H of the DNA bases {*NE, NW, SE*} with four symbols is given by.

$$\mathrm{H} = -\sum_{i=1}^{2} P_i log_2 P_i$$

where P_i, i = 1(1)3 refers to the probabilities of DNA bases *NE, NW, SE*. For Example, the entropy of the DNA four bases with Huffman code is equal to 1.5 bits per symbol.

To have strong encryption, the following conditions are the requirements, namely,

(1) The probability value of encoding with DNA bases should be lesser than encoding the original data by binary bits.
(2) The average length of the binary tree of the DNA bases should be greater than the binary bits.
(3) The entropy value of the DNA bases should be greater than the entropy value of binary bits (Refer to Figure 15c).
(4) The maximum entropy value of the DNA bases should be greater than the maximum entropy value of binary bits.

All these requirements are fulfilled in the method of this article.

5.11. Entropy of the Huffman encoded binary bits

The entropy H of the encoded binary sequence of the data with 2-symbols is given by

$$\mathrm{H} = -\sum_{i=1}^{2} P_i log_2 P_i$$

where P_i, i = 1(1)2 refers to the probabilities of the binary bits 0 and 1 of the encoded Huffman binary sequence of the data.

For Example, the entropy of the Huffman encoded binary bits 0 and 1 is equal to 1 per symbol. The information entropy reaches its maximum value of 1 since p(0) = p(1) = 0.5 and a binary message is more informative.

To have a strong compression, the following conditions are the requirements, namely,

(1) The probability value of encoding with Huffman-coded DNA bases should be greater than the probability value of encoding with binary-coded DNA bases.
(2) The average length of the binary tree of the Huffman-coded DNA bases should be lesser than the binary tree of binary-coded DNA bases.
(3) The entropy value of the Huffman-coded DNA bases should be lesser than the entropy value of binary-coded DNA bases.
(4) The probability value of the Huffman-coded binary sequence should be greater than the probability value of the Huffman-coded DNA-based sequence.
(5) The average length of the binary tree of the Huffman-coded binary bits should be lesser than the binary tree of Huffman-coded DNA bases.
(6) The entropy value of the Huffman-coded binary bits should be lesser than the entropy value of Huffman-coded DNA bases.

All these requirements are fulfilled in the method of this article.

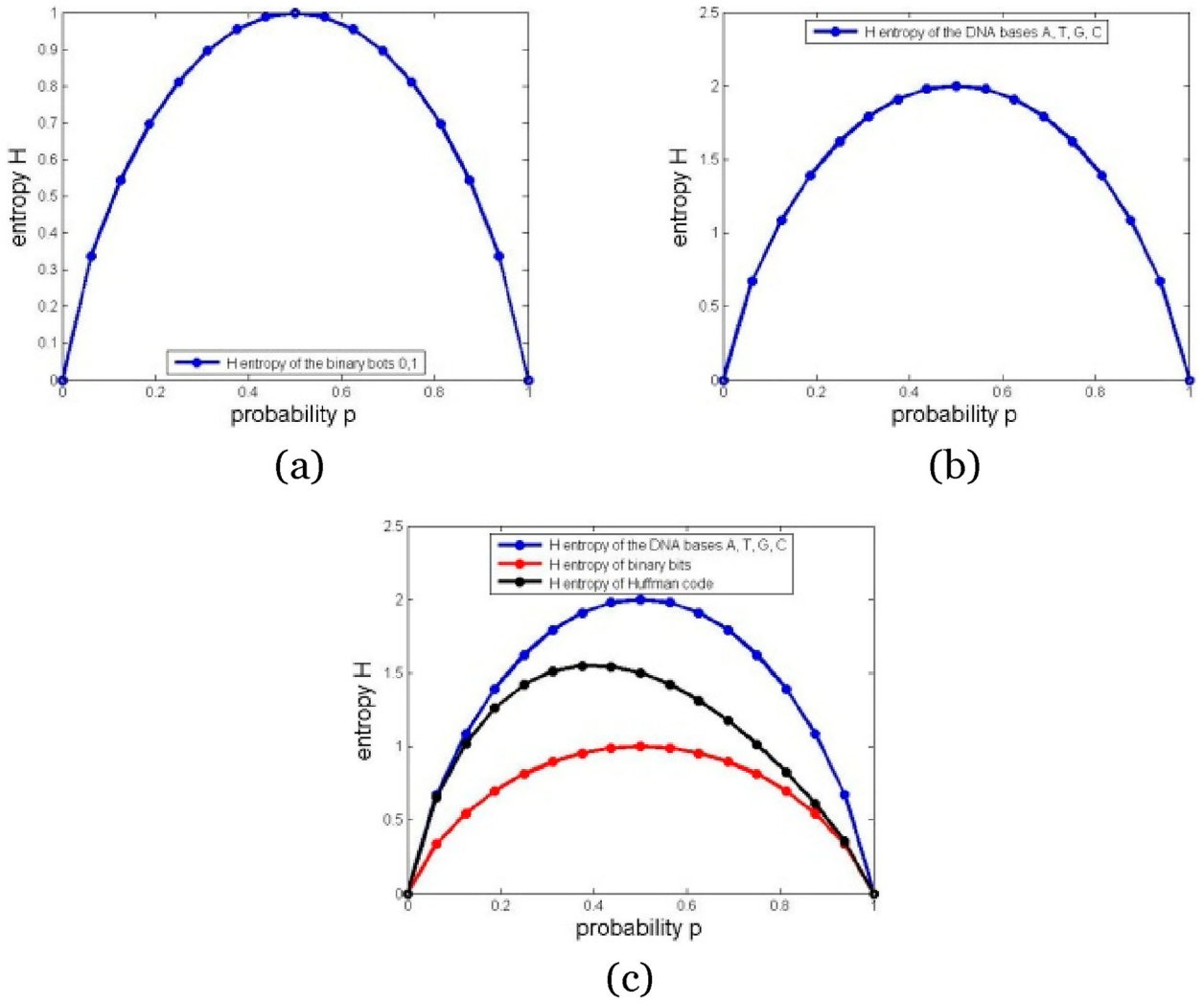

Figure 15. Comparison of information entropy of binary and DNA bits. (a) Entropy of the binary bits. (b) Entropy of the DNA bases. (c) Entropy comparison.

5.12. Cloud service provider's Risk

Risk is a very important part of the cloud to focus on stability and risk mitigation. The risk can be calculated as

$$Risk = P(\text{loss of data}) \times (\text{Amount of loss of data}).$$

The method applied in this article is a no loss of data and so there is no risk in storing the data in the cloud. Modern features enable the development of the modern IT industry by eliminating features of traditional services.

6. Comparison with existing methods

In this section, a comparison of our method with existing methods for encoding and decoding times in Figure 16. Next, our method is compared for the encryption and decryption times in Figure 17. Finally, in Figure 18, the method in this article is compared with the methods available in the literature by Auto et al. [6], Hossain et al. [8], Ibrahim et al. [7], Kazuo et al. [5], and Maria et al. [9]. From Figure 6, it is evident that the decoding time is lesser than the encoding time as the size of the data increases from 1KB to 300 KB. Similarly, From Figure 6, it is evident that the decryption time is lesser than the encryption time as the size of the data increases from 1KB to 300 KB. In Figure 16, Method-1 [7], Method-2 [5], Method-3 [6], Method-4 [8], and Method-5 Maria et al. [9] are compared with the proposed one for the whole process. It is observed from Figure 18, that the time of the whole process is increasing as the size of the data increases on applying all the methods. But, the proposed method takes more time than other methods for the same data of the same size.

Cloud computing is a technology that holds great promise and has the potential to revolutionize the healthcare sector. Numerous advantages come with cloud computing, including rapid implementation, resource sharing, cost and energy savings, and flexibility.

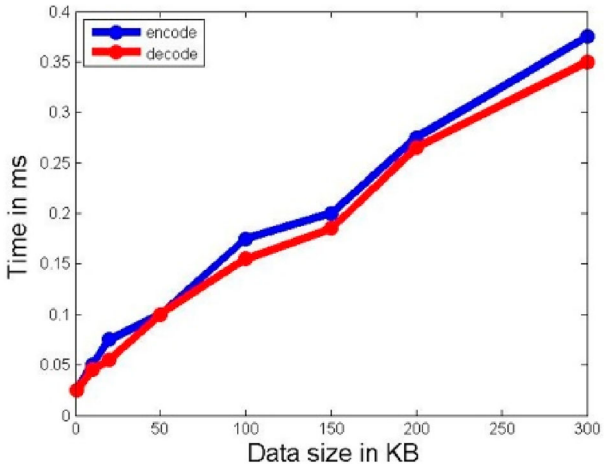

Figure 16. Encoding and decoding times.

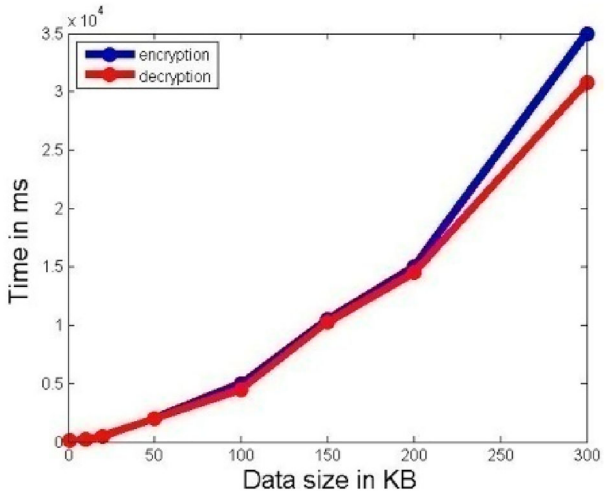

Figure 17. Encryption and decryption times.

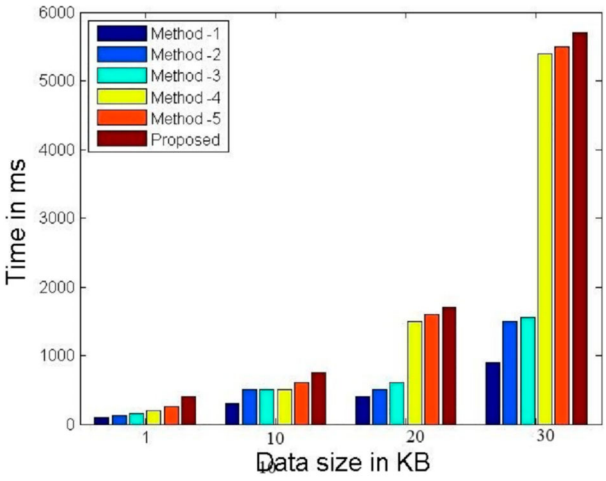

Figure 18. Comparison with existing methods.

7. Conclusions

Cloud computing is a technology that holds great promise and has the potential to revolutionize the healthcare sector. Numerous advantages come with cloud computing, including rapid implementation, resource sharing, cost and energy savings, and flexibility. Many security and privacy issues are brought up by the cloud's centralization of data for both patients and healthcare professionals. There is a need for maintaining secrecy in communication in exchanging medical data between the sender and the receiver. This can be done by cryptography.

This article presents a cryptographic method (an encryption and decryption algorithm) to have a secure communication of digital health care confidential data using DNA cryptography and the Huffman algorithm. Cloud users can transfer their applications and data to the cloud since our method will secure data more than traditional methods. In this article, the security of digital data and the reduction of the storage space of the data are considered. The cryptographic method proposed in this article uses both DNA cryptography and the Huffman algorithm to generate the key for encryption and decryption. This method uses symmetric key cryptography. The key size is 230 bits. The interesting property of our method is the cipher size of the cipher got from our algorithm is equal to the size of the cipher got from the characters of the given data. The result shows that the size of the cipher is 6 bits for the test plain text of size 32 bits so 81.25% of the storage space be saved on adopting this method. In our method, in the cipher, since $p(0) = p(1) = 0.5$, the information entropy reaches its maximum value and a binary message is more informative. The security analysis is provided to show the security of the data during both storage and transmission. The cryptographic requirements, key space analysis, key and plain text sensitivity, sensitive score analysis, sensitivity and specificity, optimal threshold, randomness analysis, uniqueness of implementation, entropies of binary bits, DNA bases, DNA bases with Huffman code, and Huffman encoded binary bits, and cloud service provider's risk are analyzed. It is observed that on applying the Huffman algorithm after applying DNA cryptography to the input data, the size of the encrypted file gets reduced from the size of the binary bits of the given data file. Our method is compared with other cryptographic methods and showed that it is more secure and stronger than other methods. Using the security score the security of the method is shown. Using sensitivity, randomness, and uniqueness of the data and cipher, it is shown that the method leads to high security.

Because DNA sequences are complicated to manipulate or decode, DNA cryptography offers a high degree of security. Since DNA sequences can be used as cryptography keys, it is practically hard to break them with existing technology. The method of this article is very readily applied, since we are not discovering keys utilizing modulo notions. Huffman coding is less appropriate in scenarios where the distribution of symbols is unknown or fluctuates dynamically since it necessitates knowing the frequency of each symbol ahead of

time. Huffman trees can be complicated and challenging to comprehend, which makes code maintenance and debugging more challenging with weak structures.

Cloud computing is a technology that holds great promise and has the potential to revolutionize the healthcare sector. Numerous advantages come with cloud computing, including rapid implementation, resource sharing, cost and energy savings, and flexibility. In the future, this work will be extended to colour images and asymmetric keys will be applied. In particular, based on the works in [48], PKE and SKE (Public Key Encryption and Symmetric Key Encryption) will be applied in the future. And to improve the accuracy of the method new methods will be applied.

Acknowledgements

We are very grateful to the reviewers of this article and the editor of the journal.

Ethical statement

The authors state that we did not include any studies involving humans or animals.

Disclosure statement

No potential conflict of interest was reported by the author(s).

ORCID

S. Lokesh http://orcid.org/0000-0003-2067-6756

References

[1] Qi ZH, Fan TR. PN-curve: A 3D graphic representation of DNA sequences and their numerical characterization. Chem Phys Lett. 2007;442(4-6):434–440. doi:10.1016/j.cplett.2007.06.029

[2] Qi XQ, Wen J, Qi ZH. New 3D graphic representation of DNA sequence based on dual nucleotides. J Theor Biol. 2007;249(4):681–690. doi:10.1016/j.jtbi.2007.08.025

[3] Qi Z-H, Li L, Qi X-Q. Using Huffman coding method to visualize and analyze DNA sequences. J Comput Chem. 2011: 3233–3240.

[4] Marty CB, Wallace DC, Baldi P. Data structures and compression algorithms for genomic sequence data. Bihoinformatics. 2009;25(14):1731–1738. doi:10.1093/bioinformatics/btp319

[5] Kazuo T, Akimitsu O, Isao S. Public-key system using DNA as a one-way function for key distribution. BioSystems. 2005;81(1):25–29. doi:10.1016/j.biosystems.2005.01.004

[6] Auto A, Khalifa A, Reda SZ. DNA-based data encryption and hiding using playfair and insertion techniques. J Commun Comput Eng. 2012;2(3):44–49.

[7] Ibrahim FE, Moussa MI, Abdelkader HM. A symmetric encryption algorithm based on DNA computing. Int J Comput Applic. 2014;97(16):41–45. doi:10.5120/17094-7634

[8] Hossain EMS, Alam KMR, Biswas MR. (2016). A DNA cryptographic technique based on dynamic DNA sequence table, Proceedings 19th ICCT, 18–20.

[9] Maria I, Sofia NR, Mahdi H, et al. An enhanced DNA sequence table for improved security and reduced computational complexity of DNA cryptography. ICST. 2020: 106–120.

[10] Krishna Gopal B, Sourav D, Manish KG. On conflict-free DNA codes. Cryptogr Commun. 2020: 1–29.

[11] Pierrick M. On the fast algebraic immunity of threshold functions. Cryptogr Commun. 2021: 1–24.

[12] Ana S, Pantelimon S. Improving bounds on probabilistic affine tests to estimate the nonlinearity of Boolean functions. Cryptogr Commun. 2022;14:459–481. doi:10.1007/s12095-021-00529-4

[13] Jaime G. Attacking the linear congruential generator on elliptic curves via lattice techniques. Cryptogr Commun. 2022;14:505525.

[14] Wray KK, Li C, Ferdinando Z. Encoding and decoding of several optimal rank metric codes. Cryptogr Commun. 2022;14:1281–1300. doi:10.1007/s12095-022-00578-3

[15] Gary Doeblien. (2023). Cybersecurity for safeguarding data privacy and security, *Calibre One Managed ICT*.

[16] Fursan Thabit SA, Abdulrazzaq HAA-A, Jagtap S. A new lightweight cryptographic algorithm for enhancing data security in cloud computing. Global Transit Proc. 2021;2:91–99. doi:10.1016/j.gltp.2021.01.013

[17] Sivan R, Ahmad Zukarnain Z. Security and privacy in cloud-based E-health system. Symmetry (Basel). 2021;13(742):1–14.

[18] Yazan A-I, Ashraf Ottom M, Tamrawi A. E-health cloud security challenges: A survey. J Healthc Eng. 2019;7516035:1–15.

[19] Dhanalakshmii G, Victo Sudha George G. An enhanced data integrity for the E-health cloud system using a secure hashing cryptographic algorithm with a password-based Key derivation Function2 (KDF2). Intern J Eng Trends Technol. 2022;70(9):290–297. doi:10.14445/22315381/IJETT-V70I9P229

[20] Nowrozy R, Ahmed K, Wang H, et al. Towards a universal privacy model for electronic health record systems. An ontology and machine learning approach. Informatics. 2023;10(60):1–28.

[21] Rodrigo T, Antunes N, Morais H. Privacy in electronic health records: a systematic mapping study. J Public Health. 2023: 1–20.

[22] Abed LN, Rashid MN, Al Okashi OM. Partial crypto-compression for cloud-based photo storage using DCT and Daubechies 4 wavelet. Int J Tech Phys Probl Eng. 2022;52(14):193–201.

[23] Ahmad I, Shin S. A novel hybrid image encryption compression scheme by combining chaos theory and number theory. Image Commun. 2021;98:116418.

[24] Ailenberg M, Rotstein OD. An improved Huffman coding method for archiving text, images, and music characters in DNA. Short Techn Rep. 2009;17(3): 747–754.

[25] Alsaffar QS, Hatem NM, Almashhdini FN. An encryption based on DNA and AES algorithms for hiding a compressed text in colored image. IOP Conf Ser Mater Sci Eng. 2021;1058(012048):1–12.

[26] Alsaffar QS. An encryption by using DNA algorithm for hiding a compressed message in Image. Wasit J Eng Sci. 2022;10(1):1–10. doi:10.31185/ejuow.Vol10.Iss1.249

[27] Alhija M, Turab N, Abuthawabeh A, et al. DNA cryptographic approaches: state of art, opportunities, and cutting edge perspectives. J Theoretical Appl Inform Tech. 2022;100(18):5346–5358.

[28] Cao B, Zhang X, Cui S, et al. Adaptive coding for DNA storage with high storage density and low coverage. Syst

Biol Applic. 2022;8(23):1–12. doi:10.1038/s41540-021-00210-9
[29] Chen C, Wen J, Wen Z, et al. DNA strand displacement-based computational systems and their applications. Front Genet. 2023: 1–12.
[30] De Silva PY, Ganegoda GU. New trends of digital data storage in DNA. Biomed Res Int. 2016;8072463:1–14. doi:10.1155/2016/8072463
[31] Dong Y, Sun F, Ping Z, et al. DNA storage: research landscape and prospects. Natl Sci Rev. 2020;7:1092 1107.
[32] Doricchi A, Platnich CM, Gimpel A, et al. Emerging approaches to DNA data storage: challenges and prospects. ACS Nano. 2022;16:17552–17571. doi:10.1021/acsnano.2c06748
[33] Goldman N, Bertone P, Chen S, et al. Toward practical high-capacity low-maintenance storage of digital information in synthesized DNA. Nature. 2013;494(7435): 7780. doi:10.1038/nature11875
[34] Golin MJ, Mathieu C, Young NE. Huffman coding with letter costs: a linear-time approximation scheme. SIAM J Comput. 2012;4(3):684713.
[35] Jameel EA, Fadhel SA. Digital image encryption techniques: Article Review. Technium. 2022;4(2):24–35. doi:10.47577/technium.v4i2.6026
[36] Mehedi M, Gurjot SG, Karanjeet C, et al. A robust and lightweight secure access scheme for cloud-based E-healthcare services. Peer-to-Peer Netw Applic. 2021;14:30433057.
[37] Amanullah M, Mishra VP, Mayavan L, et al. An Effective double verification-based method for certifying information safety in cloud computing. Int J Intell Syst Applic Eng. 2023;11(8s):268275.
[38] Bhattacharyya S, Athithan S, Pal S, et al. An IoT-enabled Intelligent and Secure Manufacturing Model Using Blockchain in Hybrid Cloud Communication System. Secur Commun Netw. 2023: 1–12. doi:10.1155/2023/7556728
[39] Li F, Wang J, Song Z. Privacy protection of cloud computing based on strong forward security. Int J Cloud Applic Comput. 2023;13(1):1–9.
[40] Altarawneh K. A strong combination of cryptographic techniques to secure cloud-hosted data. J Namib Stud. 2023;33(S2):346360.
[41] Rupa CHG, Shah MA. Novel secure data protection scheme using Martino homomorphic encryption. J Cloud Comput. 2023;12:47), doi:10.1186/s13677-023-00425-7
[42] Huffman DA. A method for the construction of minimum-redundancy codes. Proc IRE. 1952;40(9): 1098–1101. doi:10.1109/JRPROC.1952.273898
[43] Ellis Horowitz, Sha Sahri, and Dinesh Mehta. 2009. Fundamentals of data structures in C^{++}, 2nd ed. England: University Press.
[44] Ellis Horowitz, Sha Sahri, and Sengudhavar Rajasekhar. 2009. Computer Algorithms/ C^{++}, 2nd ed.. England: University Press.
[45] Jones MN, Mewhort DJK. Case-sensitive letter and bigram frequency count from largescale English corpora. Behav Res Meth Instrum Comput. 2004;36: 388–396. doi:10.3758/BF03195586
[46] Othman H, Hassoun Y, Owayjan M. (2019). "Entropy model for symmetric key cryptography algorithms based on numerical methods", *ICAR 2015*, At Beirut, Lebanon, 1–2.
[47] Shannon CE. A mathematical theory of communication. Bell Syst Tech. 1948;27(4):623–656. doi:10.1002/j.1538-7305.1948.tb00917.x
[48] Zhang K, Chen J, Lee HT, et al. Efficient public key encryption with equality test in the standard model. Theoret Comput Sci. 2019;755:65–80. doi:10.1016/j.tcs.2018.06.048