**A Hybrid DNA Based Image Encryption and Compression Techniques**

### Saravana Kumar R1, Vasantharajan V2, Aarthy J R3, Roshan Daniel Raj4, Harish Kumar S5, Gokul N6, Vijayakumar P7\*

### School of Electronics Engineering, Vellore Institute of Technology, Chennai

### e-mail : saravanaelectron@gmail.com

### e-mail : rajanvasanth270@gmail.com

### e-mail : aarthyjogi26@gmail.com

### e-mail : roshandanielraj@gmail.com

### e-mail : harishkumar020304@gmail.com

### e-mail : ngoku2003@gmail.com

### e-mail : vijayrgcet@gmail.com

### \*Corresponding Author

**Abstract:**  This paper presents an innovative method for securing text communication by encrypting and embedding text within images. The approach integrates a multi-layered security framework, utilizing Caesar Cipher encryption, LZW compression, and Visual Cryptography techniques. Initially, plaintext is encrypted with Caesar Cipher, transformed into binary, and represented as DNA strands. The DNA string is then compressed using the LZW algorithm. Subsequently, the compressed data is encoded in two-layer images using Visual Cryptography. This multi-faceted method ensures secure and covert communication, offering a robust and efficient means of protecting text messages from unauthorized access. The technique is particularly suitable for applications requiring secure data transmission, providing enhanced security for sensitive information. Moreover, the integration of these techniques leverages their combined strengths, ensuring not only data security but also maintaining the integrity and confidentiality of the transmitted messages. This comprehensive approach addresses multiple layers of security, making it a highly reliable solution for modern communication needs.

**Keywords: Ceaser Cipher Encryption; DNA Encoding; LZW Compression techniques; Visual Cryptography.**

**I. INTRODUCTION**

In today's world, securing our messages is becoming more important since hackers can easily track our communications. In this paper we discuss one of the ways to secure text in an image with help of Caesar Cipher Method, LZW Algorithm and Visual Cryptography technique. First our plaintext encrypted and converted into Ciphertext using Caesar Cipher encryption method. Secondly the encrypted ciphertext is split into each alphabetic character, and it is represented with their respective ASCII values. The ASCII values are converted into binary digits and all the binary digits are joined together. Now the binary values are split in groups of two and each value is assigned by the standard DNA strands. Now the converted DNA string is compressed with help of Lempel-Ziv-Welch Algorithm and finally it is encoded in two layer images using the Visual cryptography method. Caesar Cipher encryption technique was used to transmit secret messages, like military orders, in a way that would be unintelligible to anyone who intercepted them without knowing the decryption method. However, due to its simplicity, it's easily broken with even basic code-breaking techniques. So overcome the issues we have discussed some more algorithm for the security of the image.  In the digital world, communication happens not with letters and symbols we see, but with electrical pulses. Computers understand information through a language of 0s and 1s, known as binary. It is a process of translating the familiar characters we type, like letters, numbers, and symbols, into their corresponding binary code, allowing computers to understand and process the information we provide. After converting the plaintext into ciphertext is converted into binary values for secure side.

DNA, or deoxyribonucleic acid, is a molecule that holds the genetic instructions essential for the growth, development, functioning, and reproduction of all known living organisms and many viruses and it has been explored as a new carrier for data hiding in the literature. DNA can store massive amounts of information in its base pairs (A, C, G, T). This makes it attractive for steganography, where data is hidden within another medium. DNA is incredibly stable and can last for thousands of years under proper storage conditions. This makes it ideal for long-term data security. It mainly consists of four base pairs named Adenine (A), Cytosine (C), Guanine (G), Thymine (T).

The LZW (Lempel-Ziv-Welch) algorithm is composed of two main parts: the encoding algorithm, which translates strings into integer codes, and the decoding algorithm, which reverses this process. As the algorithm operates, it dynamically adds new integer codes for various string patterns to the code table. Initially, during the encoding phase, the code table is populated with the first 256 entries, with the rest remaining empty. LZW identifies repeated sequences in the data and incorporates them into the code table as the encoding continues. Consequently, the efficiency of the LZW algorithm improves as the input data contains more long and repeated sequences. Visual cryptography is a technique that encrypts visual information, such as text and images, in a manner that allows decryption through simple visual inspection. This emerging cryptographic method leverages human visual capabilities to decode encrypted images, providing a secure means of digital transmission that can be used once. The technology ensures that the information can be decrypted just by viewing, without the need for complex computations. In this technique there is no need for any specialized software or hardware, and it is easy to implement and understand. The objective of the research work is safely transferring the data from one network to another network using Caeser Cipher Encryption and Visual cryptography method. The project aims to secure the data in a way such that no intermediate or hacker can be able to detect it. By exploiting patterns and redundancies, the project seeks to achieve an improved security ratio without affecting the content of the data. The focus is on developing an algorithm for encrypting the data secretly, ensuring that accurate data is delivered to the receiver.

**II. RELATED WORKS**

Garima Bhargava et al. [1] proposed a technique which aims to improve image security by embedding a watermark image in the original image using a combination of compression and encryption techniques. The technique involves JPEG2000 compression, RC4 encryption, and discrete wavelet transform (DWT). The process includes stages such as preprocessing, DWT transformation, quantization, and division into different bit planes. The paper also explains the application of watermarking, its types, requirements, and life cycle. The methodology is divided into steps, including the use of discrete wavelet transform and the calculation of the watermark signal. Experimental results show the effectiveness of the proposed technique in terms of payload capacity, CPU time, PSNR, and normalized absolute error, indicating its superiority over the spread spectrum technique.

Prema T.Akkasaligar et al. [2] introduces a novel cryptosystem for the compression and encryption of medical images, aiming to ensure rapid, secure, and efficient transmission over digital communication networks. The system combines lossless Discrete Haar Wavelet Transform for compression and chaos-based DNA cryptography for encryption. It aims to address the challenges of large medical image transmission due to communication bandwidth and storage space limitations. The system employs unique encoding strategies and cryptographic techniques to enhance security while reducing time and space efficiency. Experimental results and cryptanalysis show the system's resistance against differential, exhaustive, and statistical attacks. The system's performance is assessed based on parameters like Number of Pixels Change Rate (NPCR), Unified Average Change Intensity (UACI), entropy, and compression ratio. The system's high sensitivity against pixel changes rate and resistance against exhaustive attacks further emphasize its robustness. The proposed cryptosystem offers a comprehensive solution for secure and efficient transmission of medical images, addressing confidentiality and integrity in healthcare applications.

Fadia Ali Khan et al. [3] proposed a methodology designed to protect sensitive data contained in images while also ensuring privacy. The approach combines DNA sequencing code, Arnold transformation, and a chaotic dynamical system to generate an initial S-box. Various tests are conducted on the S-box to confirm its randomness, such as NIST analysis, histogram analysis, nonlinearity analysis, strict avalanche criterion, bit independence criterion, equiprobable input/output XOR distribution, and linear approximation probability. The findings indicate that the proposed method offers enhanced security and resilience against attacks. A comparative analysis with current state-of-the-art techniques demonstrates the efficacy of the newly proposed algorithm. This research paper makes a significant contribution to the fields of privacy-preserving visual recognition and image encryption by presenting a unique and efficient strategy that surpasses existing approaches.

Jianzhang Chen et al. [4] proposed a new colour image encryption and compression algorithm that uses the DNA complementary rule and the Chinese remainder theorem. The algorithm shuffles the colour image and uses the Chinese remainder theorem to diffuse and compress it simultaneously. The authors emphasize the need for secure image encryption and compression due to the rise in cloud computing and big data. They discuss previous research on image encryption, including chaos theory, fractal theory, DNA encoding, modern cryptography, and quantum chaotic maps. The algorithm's key features include a large key space, resistance against common attacks, good encryption results, and effective compression with a compression ratio of 4. The authors also evaluate the algorithm's performance, highlighting its strong sensitivity, good randomness, and low correlation between plain and encrypted images. The study was supported by various research foundations and includes a list of references to related work in the field.

Xiulai Li et al. [5] proposed a novel methodology to encrypt iris characteristics utilizing deep learning techniques for classification, with a specific focus on iris attributes. Their approach underwent assessment via simulated experiments utilizing a standardized iris database. The outcomes illustrated a substantial improvement in the consistency of iris encryption, thereby enhancing the security of both encryption and decryption procedures. Due to the visual nature of images, which holds a pivotal role in the exchange of information, safeguarding image data is imperative. Potential threats such as malicious attacks can compromise this security, potentially resulting in data breaches or corruption. Image encryption mechanisms are crafted to fortify the safeguarding of image data, mitigating the likelihood of data exposure and ensuring the secure transmission of the initial content.

Noha O. Korany et.al, [6] proposed a novel image encryption technique that combines fuzzy integral theory and DNA-based encryption. It outlines a series of steps including generating pseudo-random sequences, shuffling binary streams, DNA encoding, splitting and diffusing images, fusion using Discrete Wavelet Transform (DWT), and hiding encrypted sub-images into carrier images using steganography. The proposed method undergoes rigorous security analysis demonstrating its robustness against various attacks. Overall, the approach offers high security and resistance to attacks, making it suitable for future multimedia communication systems.

Qiang Zhang, Ling Guo et.al. [7] proposed a novel image encryption technique combining chaotic systems and DNA sequences to address the security challenges in digital image transmission over networks. Traditional encryption methods are found inadequate for image encryption, necessitating the development of new approaches. The proposed algorithm utilizes chaotic maps and DNA sequence operations to encrypt images effectively. Simulation results and security analysis demonstrate the algorithm's effectiveness in resisting various attacks and ensuring secure image transmission. Overall, the hybrid encryption technique offers improved security, a larger key space, and resistance to known attacks, making it suitable for practical applications .

Jianfeng Zhao et al. [8] proposed a novel image encryption scheme utilizing a combination of a new four-dimensional chaotic system, Zigzag transformation, and DNA information. The proposed encryption algorithm involves several steps, including generating chaotic sequences, block scrambling using Zigzag transform, and applying DNA encoding and XOR operations. Various experiments and analyses demonstrate the effectiveness and security of the encryption scheme, including statistical histogram analysis, correlation of  adjacent pixels analysis, information entropy analysis, resistance against differential attacks, key sensitivity analysis, resistance against data loss and noise attacks, encryption quality analysis, time complexity analysis, and key space analysis. Overall, the proposed algorithm shows promising results in terms of security, efficiency, and resistance against various attacks.

Xingyuan Wang et al. [9] proposed an innovative approach to image encryption using a combination of a four-wing chaotic system, DNA encoding, and compressed sensing (CS). This method aims to enhance security while efficiently compressing images for transmission over networks. Key features include leveraging chaotic sequences to control DNA coding, using CS for compression and encryption, and generating initial values from hash functions to resist chosen-plaintext attacks. Experimental results demonstrate the effectiveness of the proposed scheme in terms of security and image quality preservation.

Ledya Novamizanti et al. [10] proposed a comprehensive approach to enhancing data security through a combination of LZW Compression, RSA Encryption, and DCT Steganography. It discusses the theoretical background of these techniques, proposes a system design for implementing them, and analyzes potential challenges and solutions. The key focus is on addressing synchronization issues between different stages of the process to ensure the effectiveness and reliability of data protection.

**III.   PROPOSED DNA BASED TEXT ENCRYPTION AND COMPRESSION TECHNIQUES**

The research work aims to develop a secure communication method that prevents unauthorized tracking or detection of data messages. To achieve this, we employed a combination of algorithms and techniques to ensure the confidentiality and integrity of the message. As a demonstration, we used the message "WELCOME TO VIT" as the data to be transmitted securely. First, we applied the Caesar Cipher algorithm to encrypt each letter of the message separately. This involved shifting each letter by a fixed number of positions in the alphabet, resulting in a ciphertext that appears meaningless to an unauthorized party. Next, we separated the letters of the ciphertext and represented each letter by its corresponding ASCII value. These ASCII values were then converted into 8-digit binary form, resulting in a string of binary digits that represent the encrypted message. To further conceal the message, we split the binary string into groups of two and replaced each group with a corresponding DNA[12] code. This DNA code was then compressed using the Lempel-Ziv-Welch (LZW) algorithm, which reduces the data size while preserving its content. The compressed DNA code was then hidden in a grayscale image using Visual Cryptography[11], a technique that encrypts data in images in such a way that only the intended recipient can decipher it. The overall flow of the research paper is explained as block diagram in Fig. No. 1 which gives detailed information as flowchart about the encryption and compression techniques involved in this paper. By combining these advanced techniques—Caesar Cipher, ASCII conversion, binary encoding, DNA mapping[13], LZW compression[15], and Visual Cryptography—we developed a multi-layered approach to secure communication. Each layer adds a unique element of security, ensuring that the message remains confidential and intact throughout the transmission process. This research contributes to the field of secure communications by presenting a comprehensive and innovative method to protect data from unauthorized access and detection.

**3.1 Caesar Cipher Method**

In cryptography, the Caesar cipher, also known as Caesar's cipher, the shift cipher, Caesar's code, or Caesar shift, is one of the most basic and well-known encryption methods. This substitution cipher replaces each letter in the plaintext with a letter a fixed number of positions down the alphabet. The Caesar cipher's encryption process is often integrated into more complex systems, like the Vigenère cipher, and is still used in modern applications such as the ROT13 system. However, like all single alphabet substitution ciphers, the Caesar cipher is easily deciphered and does not provide effective security in contemporary communication.

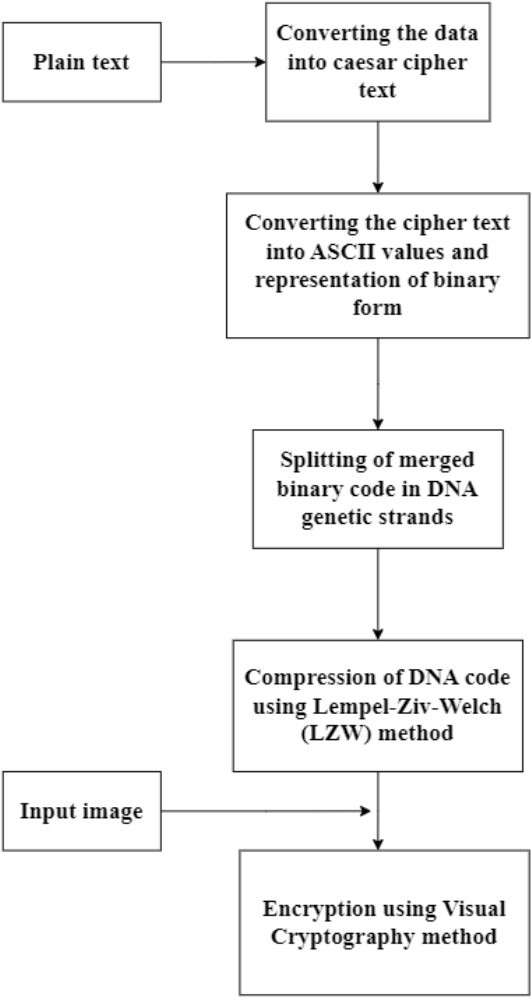


Fig. No. 1 Proposed DNA based Encrypting Process

Consider the text "WELCOME TO VIT" to be sent secretly to a receiver. The letters of the plaintext are identified with their corresponding plaintext positions using the following Table 01. Let's take a **private key as 15**, which is shared with the receiver. The plaintext positions for the letters W, E, L, C, O, M, E, T, O, V, I, and T are 22, 4, 11, 2, 14, 12, 4, 19, 14, 21, 8, and 19, respectively. After identifying the plaintext positions, the ciphertext positions are encrypted using the formula **C = E(K, P) mod 26.** where P is the private key and K is the plaintext position.

**Table 1: Mapping Table**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **A** | **B** | **C** | **D** | **E** | **F** | **G** | **H** | **I** | **J** | **K** | **L** | **M** |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| **N** | **O** | **P** | **Q** | **R** | **S** | **T** | **U** | **V** | **W** | **X** | **Y** | **Z** |
| 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |

Table 1 shows the mapping of letters (A-Z) to numbers (0-25) for Caesar Cipher encryption. Each letter is shifted by a fixed number of positions (key) in the alphabet to encrypt the message. The key is added to the letter's numerical value to get the encrypted letter's numerical value. Let us take the plaintext position of W is 22. Adding the private key 15 to this value, we get:

Cw = (22 + 15) mod 26 = 37 mod 26 = 11

The ciphertext position for the value 11 in the table is L. Thus, the ciphertext for the value W is represented by L.

Also, for the plaintext position of E is 4. Adding the private key value 15 to this value, we get:

CE = (4 + 15) mod 26 = 19 mod 26 = 19

The ciphertext position for the value 19 in the table is T. Thus, the ciphertext for the value E is represented by T. The same procedure is repeated for the all the letters of the plaintext.

**Table 2 : Encryption Progression**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Plaintext** | **W** | **E** | **L** | **C** | **O** | **M** | **E** | **T** | **O** | **V** | **I** | **T** |
| **Plaintext position** | 22 | 4 | 11 | 2 | 14 | 12 | 4 | 19 | 14 | 21 | 8 | 19 |
| **Private Key** | **15** | **15** | **15** | **15** | **15** | **15** | **15** | **15** | **15** | **15** | **15** | **15** |
| **Ciphertext**  **position** | 11 | 19 | 0 | 17 | 3 | 1 | 19 | 8 | 3 | 10 | 23 | 8 |
| **Ciphertext** | **L** | **T** | **A** | **R** | **D** | **B** | **T** | **I** | **D** | **K** | **X** | **I** |

Table 2 details the encryption of the plaintext using the Caesar Cipher with a key of 15. Each letter in the plaintext is first converted to its corresponding position (0-25) in the alphabet. The private key (15) is then added to each position, and the result is taken modulo 26 to stay within the alphabet range. These new positions are mapped back to letters, producing the ciphertext "LTARDBTIDKXI". This method ensures that each letter is shifted consistently, creating an encrypted message that appears meaningless without the key. Thus, the plaintext **'WELCOME TO VIT'** is converted into ciphertext as  **'LTARDBTIDKXI'.** The letters of the ciphertext are then converted into binary digits based on their ASCII values for further processing.

**3.2 Conversion of ciphertext into Binary digits**

ASCII, or the American Standard Code for Information Interchange, is a fundamental building block in the digital world. It acts as a translator between the human-readable characters we use and the binary code that computers understand. Each letter, number, symbol, and even control character in ASCII is assigned a unique numerical value between 0 and 127. This allows computers to store and transmit information efficiently. For example, the ASCII value of 'W' is 87, and the ASCII value of 'l' is 108. After identifying the ASCII values of all the ciphertext, it is converted into binary digits and merged together for secure processing. Let us take the ASCII value of 'L' is 76, and it is converted into an 8-digit binary representation as 01001100. The ASCII value of 'T' is 84, and its 8-digit binary representation is 01010100. We continue the same process for all the letters of the ciphertext.

**Table 3 : Conversion of Ciphertext into Binary Digit**

|  |  |  |
| --- | --- | --- |
| **Cipher text** | **ASCII Values** | **Binary Digit Representation** |
| **L** | 76 | 01001100 |
| **T** | 84 | 01010100 |
| **A** | 65 | 01000001 |
| **R** | 82 | 01010010 |
| **D** | 68 | 01000100 |
| **B** | 66 | 01000010 |
| **T** | 84 | 01010100 |
| **I** | 73 | 01001001 |
| **D** | 68 | 01000100 |
| **K** | 75 | 01001011 |
| **X** | 91 | 01011011 |
| **I** | 73 | 01001001 |

Now, merge all the binary values together and split in twos and encode in the form of standard DNA strands. The merged binary code for the ciphertext value is given by:

**01001100 01010100 00100000 10100100 10010001 00010001 00010010 10100010 01001001 00010010 10110101 10010110 10010001**

To encode this in the form of standard DNA strands, we split the binary code into pairs (twos):

**01 00 11 00 01 01 01 00 00 10 00 00 10 10 01 00 10 01 00 01 00 01 00 01 00 10 10 00 10 01 00 10 01 00 01 00 10 10 11 01 01 10 01 01 10 10 01 00 01**

**3.3 DNA Mapping process**

DNA encoding is a groundbreaking process that converts digital data, including text and images, into a format readable by DNA. This cutting-edge technology utilizes the distinctive structure of DNA to store information, employing the sequence of the four DNA bases—Adenine (A), Guanine (G), Cytosine (C), and Thymine (T)—as the medium. Specialized algorithms translate the data into sequences of A's, C's, G's, and T's, which are then synthesized and stored for long durations. To retrieve the stored information, the DNA sequence is read and converted back to its original digital form. Although DNA encoding promises a high-density storage solution, it remains in the developmental stage due to challenges like high costs and slow read/write speeds. Nevertheless, it holds great promise for long-term archival and future advancements in data storage technology. The genetic code for each bases is denoted by Table 4.

**Table 4 : DNA Mapping**

|  |  |
| --- | --- |
| **Nucleotide** | **Binary Value** |
| A | 00 |
| C | 10 |
| G | 11 |
| T | 01 |

The table-04 maps DNA nucleotides to their corresponding binary values, where A = 00, C = 10, G = 11, and T = 01. This allows binary data to be encoded using DNA sequences, enabling the representation of digital information in a biological format. Let us take the binary value of L in 8-digit is 01001100. Splitting it into pairs (twos), we get 01 00 11 00. Each pair is then represented by the corresponding base nucleotides as TAGA. Similarly, for T, the binary value is 01010100, and splitting it into pairs, we get 01 01 01 00, with the respective DNA strands being TTTA. Continuing this process for all the letters in the ciphertext and merging them together, we can achieve the encoded DNA string. Merged DNA Encoded Ciphertext is

TAGATTTATAATTTACTATATAACTTTATACTTATATACGTTCGTACT.

**3.4   Lempel-Ziv-Welch Algorithm based Compression Techniques**

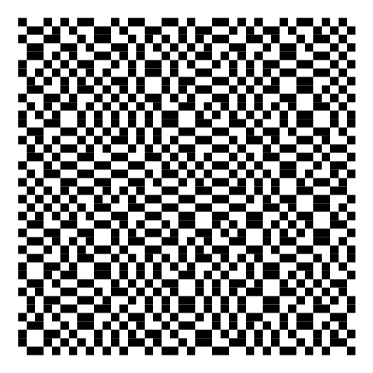
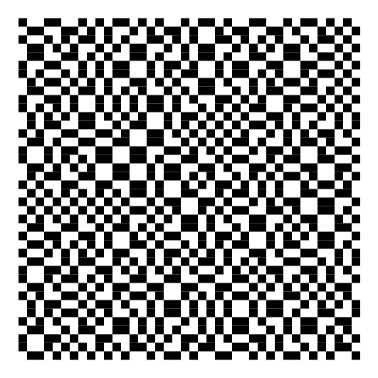
The DNA-encoded ciphertext contains a long string, and sometimes it's difficult to hide a huge string in an image. To compress the DNA code, we use the LZW algorithm. The LZW, or the Lempel-Ziv-Welch method is a lossless data compression process that works by building a dictionary of substrings as they appear in the data. The LZW method starts by initializing a dictionary with all possible single characters. Then, it reads the input data one character at a time, looking for matching substrings in the dictionary. If a match is found, the algorithm adds the next character to the substring and adds the new substring to the dictionary. If no match is found, the algorithm outputs the index of the longest matching substring and starts a new substring with the current character. This process continues until the end of the input data, resulting in a compressed output consisting of indices and new substrings In this implementation, the process begins with the index values 0, 1, 2, and 3.The compression process is explained in the Table-5. Let us consider the ccompressed DNA-Encoded Ciphertext: 3 0 2 0 3 8 6 0 7 9 1 4 15 4 10 14 16 19 20 2 4 14. The compression ratio to the original string is 0.62%.

**Table 5 : LZW Compression Techniques**

|  |  |  |
| --- | --- | --- |
| **DNA-Genetic Code** | **Index** | **Dictionary Value** |
| A | 0 | - |
| C | 1 | - |
| G | 2 | - |
| T | 3 | - |
| TA | 4 | 3 |
| AG | 5 | 0 |
| GA | 6 | 2 |
| AT | 7 | 0 |
| TT | 8 | 3 |
| TTA | 9 | 8 |
| ATA | 10 | 6 |
| AA | 11 | 0 |
| ATT | 12 | 7 |
| TTAC | 13 | 9 |
| CT | 14 | 1 |
| TAT | 15 | 4 |
| TATA | 16 | 15 |
| TAA | 17 | 4 |
| AAC | 18 | 10 |
| CTT | 19 | 14 |
| TATAC | 20 | 16 |
| CTTA | 21 | 19 |
| TATACG | 22 | 20 |
| GT | 23 | 2 |
| TAC | 24 | 4 |
| CT | 25 | 14 |

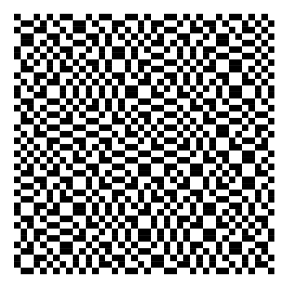
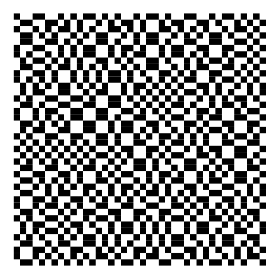
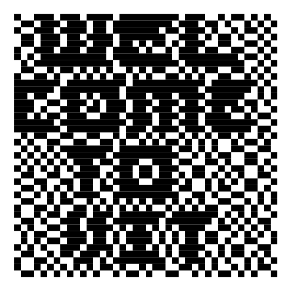
**3.5 Image hiding using Visual Cryptography technique**

Visual cryptography is a technique used to secure visual information, such as images, by dividing it into shares that can be combined to reconstruct the original image. It involves breaking down the image into multiple seemingly random shares, which often look like noise when viewed individually. Decryption is achieved by overlaying a specific number of these shares, revealing the original image. This method manipulates pixels cleverly; each pixel in the secret image is divided into subpixels in each share. Algorithms determine the color of these subpixels based on the secret image and the role of the share. For example, a white pixel in the secret image might be represented by a random arrangement of black and white subpixels in one share, but when combined with another share having a complementary pattern, the human eye perceives a solid white pixel. Visual Cryptography is a simple technique to understand, and decryption requires no special software or computational power, just the ability to stack the shares. This makes it ideal for scenarios where secure communication is paramount but access to sophisticated technology is limited. Visual Cryptography has applications in various fields, including data security, copyright protection, and medical image sharing. The DNA-encoded ciphertext is in the form of decimal numbers. To hide the ciphertext values in an image, we first convert the decimal numbers to binary form to hide them in the image. Then, 0's are represented by white pixels and 1's are represented by black pixels. When the pixels of both layers are overlapped with each other, we can detect the message. This process adds an additional layer of security to the message. Lets us take Figure 2(a) and 2(b) as

  
 **Figure 2(a) Figure 2(b)**

**[Layer 01 : 20x20 pixel] Layer 02 : 20x20 pixel]**

Although both images appear similar to the naked eye, they contain encrypted messages within them. When both layers are overlapped with each other at a specific point, we can detect the encoded message. The result of the overall mapping process is shown in the figure 2(c).

(Layer 1) (Layer 2) (Layer 3)

**Figure 2(a) Figure 2(b)**  **Figure 2(c)**

**IV.   RESULT AND DISCUSSION**

Python can be used as a simulation tool for text encryption using Caesar Cipher, compression using LZW Algorithm, and hiding of data in an image using Visual Cryptography. Python is used as the programming language to implement the compression techniques and perform the necessary computations. Python libraries like NumPy are utilized for array manipulation, and the code is executed using a Python interpreter or any Python development environment. The code can be run using any Python development environment or IDE, such as Anaconda, Jupyter Notebook, PyCharm, or simply by executing it in a Python interpreter. The following are the simulation parameters are used to analyze the performance of the proposed DNA based encryption:

* Time Complexity
* Plaintext Size
* Key Size

**4.1 Encryption time**

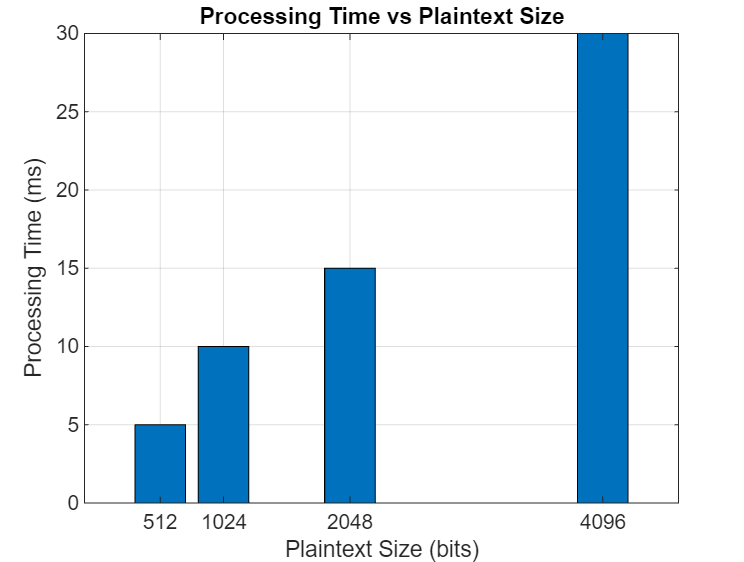
With a fixed key size of 64 bytes, the graph explains the process of varying plaintext sizes and the corresponding processing times for the execution of the encryption algorithm. Table 6 explains the relationship between plaintext size (in bits) and processing time (in milliseconds) for the encryption algorithm, with processing time increasing as plaintext size increases. The table reveals a direct proportional relationship between plaintext size and processing time, with processing time doubling as plaintext size quadruples.

**Table 6 : Processing time(ms) Vs Plaintext Size(in bits)**

Key Size = 64 bites

|  |  |  |
| --- | --- | --- |
| **S.No** | **Plaintext Size (in bits)** | **Processing time (ms)** |
| 1. | 512 | 5 |
| 2. | 1024 | 10 |
| 3. | 2048 | 15 |
| 4. | 4096 | 30 |

Fig. 3(a) visually communicates how encryption time varies with different sizes of plaintexts. Each bar's height indicates the time required to encrypt data of a specific size, highlighting the trend that larger plaintext sizes generally result in longer encryption times. This visualization is useful for understanding the computational demands of encryption as data size increases, aiding in optimizing encryption processes and selecting appropriate strategies for managing performance based on plaintext size constraints.



**Fig.3.(a) Processing time(ms) Vs Plaintext Size(in bits)**

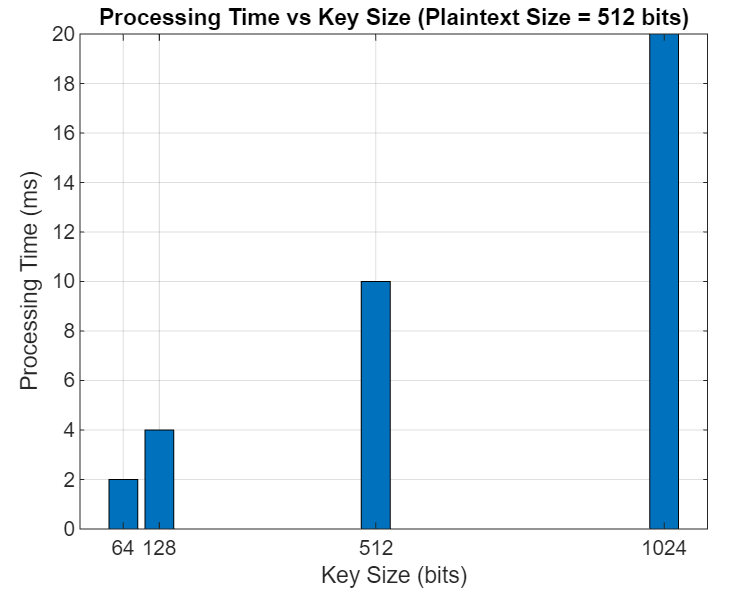
Now with taking the plaintext value as constant and varying the key size we result as shown in table Table.7.

**Table 7: Processing time(ms) vs Key size (in bits)**

Plaintext Size = 512 bit

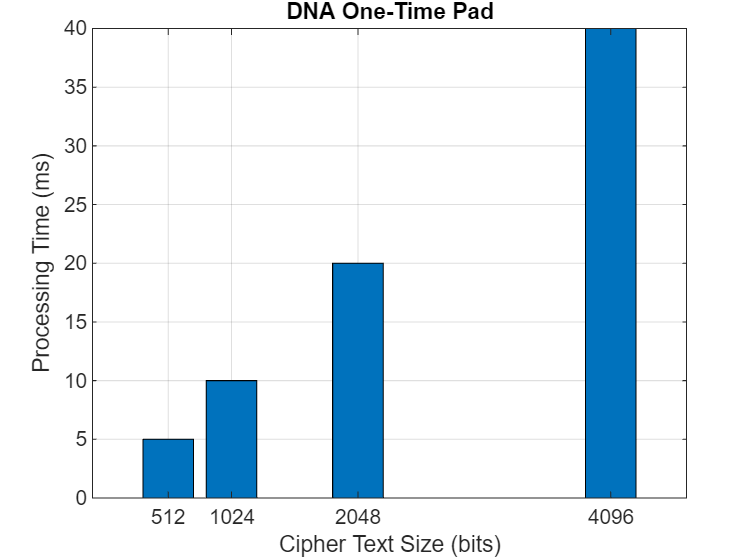
|  |  |  |
| --- | --- | --- |
| **S.No** | **Key Size (in bits)** | **Processing time (ms)** |
|  | 64 | 2 |
|  | 128 | 4 |
|  | 512 | 10 |
|  | 1024 | 20 |

Table 7 presents the relationship between key size (in bits) and processing time (in ms) for the encryption algorithm. It demonstrates that as the key size increases, the processing time also increases. The table indicates a direct proportional relationship between key size and processing time, with processing time doubling as the key size doubles. This data suggests that larger key sizes demand more processing time, which can affect the efficiency of the encryption process. The visualization of this relationship is illustrated in Fig 3(b).



**Fig.3(b) Processing time (ms) vs Key size (in bits)  
4.2 DNA encoding**

The DNA one-time pad is a highly secure encryption technique that utilizes DNA coding to encrypt plaintext. In this method, each plaintext bit is encoded into a corresponding DNA base (A, C, G, or T), resulting in a significant expansion of the data size. Although this technique offers unbreakable security, it comes at the cost of increased processing time.



**Fig.4. DNA One-Time Pad**

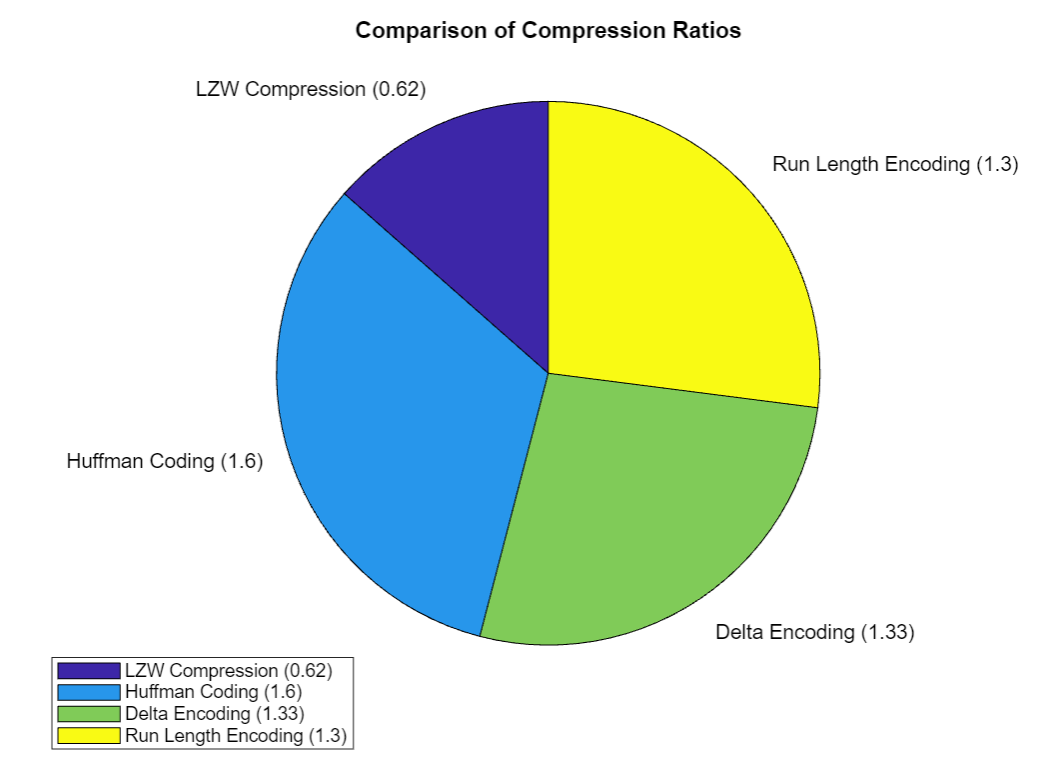
This bar graph in the Fig.4. visually communicates how the processing time varies with different sizes of cipher texts encrypted using the DNA One-Time Pad method. Each bar's height indicates the time required to process a cipher text of specific size, highlighting potential trends or performance implications based on cipher size. It serves as a useful tool for understanding the computational demands associated with encryption tasks and can guide decisions in optimizing performance or selecting suitable encryption strategies based on data size constraints. The process is explained as tabulation in table.8.

**Table 8: Ciphertext size (in bits) vs Processing time (ms)**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Ciphertext size (in bits)** | **Processing time (ms)** |
|  | 512 | 5 |
|  | 1024 | 10 |
|  | 2048 | 20 |
|  | 4096 | 40 |

**4.3 Compression Techniques**

There are several techniques for compression and we have used LZW algorithm for compressing the ciphertext and shown the differences in their compression ratios as pie chart in the Fig.5.



**Fig.5. Comparision of Compression Ratios**

The pie chart of Fig.5. visually compares the theoretical compression ratios of different techniques—LZW, Huffman coding, delta encoding, and run-length encoding—applied to the text "WELCOME TO VIT". Each slice in the chart represents one technique, with its size relative to the whole indicating the compression efficiency: smaller slices suggest higher compression ratios. This visualization offers a clear, immediate comparison of how each technique would potentially reduce the size of data, providing insights into their relative effectiveness for storage or transmission applications in data compression.

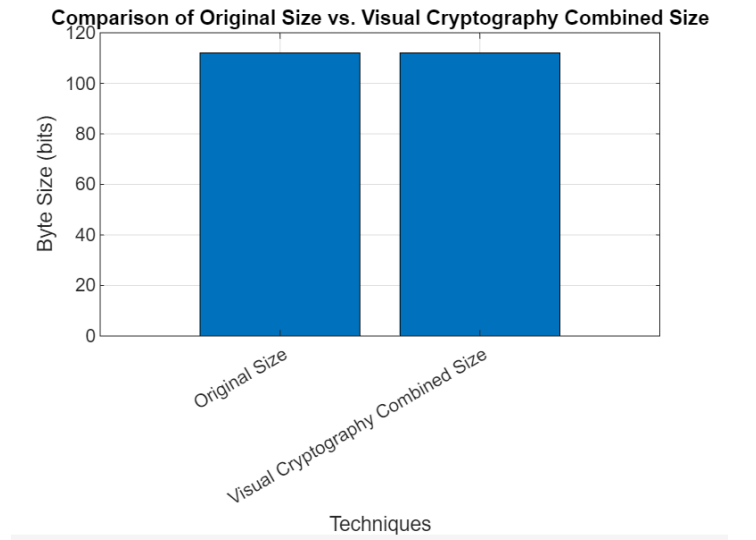
**Table 9 : Techniques vs Compression Ratio**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Techniques** | **Compression Ratio** |
| 1. | LZW | 0.62 |
| 2. | Huffman coding | 1.6 |
| 3. | Run length | 1.3 |
| 4. | Delta encoding | 1.33 |

 Table.9. compares the compression ratios of different data compression techniques. LZW achieves a compression ratio of 0.62, indicating a higher level of compression. Huffman coding, Run Length, and Delta encoding have compression ratios of 1.6, 1.3, and 1.33, respectively, reflecting varying efficiencies in data reduction.

**4.4 Visual Cryptography**

The comparison the original data size to the encrypted data size using Visual Cryptography, is shown in the Fig.6 and as a result both sizes remain the same. Visual Cryptography encrypts the data by splitting the original image into multiple shares, each containing part of the encrypted data. Despite the encryption, the overall size of the encrypted image does not increase, ensuring that the data retains its original dimensions and resolution. This makes Visual Cryptography an efficient method for secure image transmission without compromising on data size.



**Fig. 6 Original state vs Visual cryptography combined size**

**V. CONCLUSION AND FUTURE WORKS**

In this paper, we introduced an innovative approach to text encryption by combining the Caesar Cipher, DNA genetic coding, LZW compression, and Visual Cryptography. Our method harnesses the strengths of each technique to provide robust security and privacy for encrypted data. By converting the encrypted text into DNA genetic code, we added an additional security layer, making it more challenging for unauthorized parties to access the data. The LZW compression algorithm further enhanced efficiency by reducing the data size, facilitating more efficient storage and transmission. Finally, embedding the compressed data within an image using Visual Cryptography added another layer of security, making the encrypted data more difficult to detect. Our Python implementation demonstrates the feasibility and effectiveness of this approach. The results indicate that our method can successfully encrypt and decrypt text data, ensuring both confidentiality and integrity. This approach has potential applications in secure data transmission, digital forensics, and data hiding.

**REFERENCES**

[1] Bhargava, G., & Mathur, A. (2015). Enhanced spread spectrum image watermarking with compression-encryption technique. *Journal Title*, pp. 256-261. <https://doi.org/10.1109/ISCO.2014.7103956>

[2] Akkasaligar, Dr. Prema, & Biradar, Sumangala. (2020). *Medical Image Compression and Encryption using Chaos*  *based DNA Cryptography*, pp. 1-5. <https://doi.org/10.1109/B-HTC50970.2020.9297928>

[3] Masood, F., Masood, J., Zhang, L., Jamal, S. S., Boulila, W., Rehman, S. U., Khan, F., & Ahmad, J. (2022). A new color image encryption technique using DNA computing and Chaos-based substitution box. *Soft Computing*, 26, 1-17. <https://doi.org/10.1007/s00500-021-06459-w>

[4] L. Guo, J. Chen and J. Li, "Chaos-Based color image encryption and compression scheme using DNA complementary rule and Chinese remainder theorem," 2016 13th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), Chengdu, China, 2016, pp. 208-212, doi: 10.1109/ICCWAMTIP.2016.8079839.

[5] Li, X., Jiang, Y., Chen, M., & Li, F. (2018). Research on iris image encryption based on deep learning. *EURASIP*  *Journal on Image and Video Processing*, 2018(1). <https://doi.org/10.1186/s13640-018-0358-7>

[6] El-Khamy, S., Korany, N., & Gamal, A. (2020). A new Fuzzy-DNA Image Encryption and Steganography Technique. *IEEE Access*, PP, 1-1. <https://doi.org/10.1109/ACCESS.2020.3015687>

[7] Zhang, Q., Guo, L., & Wei, X. (2010). Image encryption using DNA addition combining with chaotic maps. *Mathematical and Computer Modelling*, 52(11-12), 2028-2035. <https://doi.org/10.1016/j.mcm.2010.06.005>

[8] Zhao, J., Wang, S., & Zhang, L. (2023). Block Image Encryption Algorithm Based on Novel Chaos and DNA Encoding. *Information*, 14(3), 150. <https://doi.org/10.3390/info14030150>

[9] Wang, X., & Su, Y. (2021). Image encryption based on compressed sensing and DNA encoding. *Signal*  *Processing: Image Communication*, 95, 116246. <https://doi.org/10.1016/j.image.2021.116246>

[10] Novamizanti, L., Budiman, G., & Iwut, I. (2015). Designing secured data using a combination of LZW compression, RSA encryption, and DCT steganography. In *Proceedings of the International Conference on*  *Wireless Technology (ICWT)*, pp. 1-6. <https://doi.org/10.1109/ICWT.2015.7449245>

[11] Vijayakumar, P., & Rajashree, R. (2015). A Hybrid Secure Architecture for Passport Verification System Using Visual Cryptography and Steganography. *Journal of Chemical and Pharmaceutical Sciences*, Special Issue 10: July 2015, pp. 74-78.

[12] Vijayakumar, P., Vijayalakshmi, V., & Zayaraz, G. (2015). Hybrid Secure GSM Architecture using DNA Computing based Hyperelliptic Curve Cryptography. *International Journal of Electronic Security and Digital*  *Forensics*, 7(2), 105-118. doi:10.1504/IJESDF.2015.069598.

[13] P. Vijayakumar, S. Indupriya and R. Rajashree, A Hybrid Multilevel Security Scheme using DNA Computing based Color Code and Elliptic Curve Cryptography, ISSN (Print) : 0974-6846 ISSN (Online) : 0974-5645, *Indian*  *Journal of Science and Technology*, Vol 9(10), DOI: 10.17485/ijst/2016/v9i10/88987, March 2016**.**

[14] P. Vijayakumar, V. Vijayalakshmi, R. Rajashree, Increased Level of Security Using DNA Steganography, Inderscience- *International Journal of Advanced Intelligence Paradigms (IJAIP)*, Vol. 10, Issue 1/2, pp. 74-82, January 2018.

[15] M. G.C and V. Perumal, "Encryption with Automatic key Generation and Compression," 2022 *Third International*  *Conference on Intelligent Computing Instrumentation and Control Technologies (ICICICT)*, 2022, pp. 295-301, doi: 10.1109/ICICICT54557.2022.9917718.