e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:04/April-2025 Impact Factor- 8.187 www.irjmets.com

A NOVEL APPROACH FOR SECURING DATA USING STEGANOGRAPHY ENCRYPTION TECHNIQUE IN AN IOT NETWORK

B. Vijay[*1], D. Kavyasree[*2], V. Tejaswi[*3], T. Sai Vinay[*4], D. Bhuvaneswari[*5]

[*1,2,3,4,5]Department Of Computer Science & Engineering Aditya Institute Of Technology And Management Tekkali, India.

DOI: https://www.doi.org/10.56726/IRJMETS71882

ABSTRACT

The usage of Internet of Things (IoT) devices has led to a significant increase in data transmission, making security an important concern. Traditional encryption techniques, although effective, can be vulnerable to attacks and may not provide adequate protection for sensitive data. This project proposes a novel approach to secure data transmission in IOT networks by combining steganography and encryption techniques.

Steganography is known to image encryption technique; it provides an additional layer of security by concealing the existence of secret information. This project collects data from the IOT sensor and encrypts it in image using encryption algorithm. The receiver at the other end decrypts it to get the confidential information using a secret key.

The proposed approach uses advanced encryption standard (AES) and RSA encryption algorithm, which is widely used in data confidentiality due to its lightweight and efficient nature and uses in a network security, where in the IoT network, the sensors emitting the data and communicate over different protocols to the server. Where the proposed model takes place to encrypt transmitted data.

Keywords- Iot Data, Encryption, Steganography, Advanced Encryption Standard (AES) Algorithm, Least Significant Bit (LSB) Algorithm.

I. INTRODUCTION

The Internet of Things (IoT) has become a disruptive force in many industries, and one of the most crucial areas for its use is healthcare.The integration of IoT Technologies with healthcare is rapidly transforming the way healthcare services are being offered. In recent years, the demand for remote patient monitoring and telehealth services has escalated, driven by the need for enhanced patient engagement and accessibility to care.

The healthcare IoT market worldwide is projected to reach a revenue of US$93.28 billion by 2025. It is expected to show an annual growth rate (CAGR 2025-2029) of 9.56%, leading to a market volume of US$134.40 billion by 2029. Recently, the healthcare IoT market has been rapidly growing, with an increasing number of hospitals adopting smart medical devices and remote patient monitoring systems, which leads to efficient services and wider accessibility.

IoT data is the information collected by interconnected devices and sensors that monitor the vital signs of the patient, including temperature, blood pressure, and heart rate. This data enables real-time health monitoring, enhances patient care, and supports data analytics for improved healthcare outcomes. The healthcare industry is responsible for generating approximately 30% of the world's information, which is a substantial portion of global data. The enormous data generated also helps in the analysis of medical trends and important metrics of the patient's health.

In an era where digital information is everywhere and increasingly vulnerable to cyber threats, the need for robust data security measures has never been more important. As patients and hospitals extensively rely on digital platforms for monitoring, accessing, and data storage, the protection of sensitive information from unauthorized access and breaches has become an important parameter to the healthcare companies. Even though there are many ways to safeguard the information, the data security has become an unavoidable and serious concern.

In the rapidly evolving world, there are many technologies, procedures and methods followed to ensure data security. Among them strong Firewalls, Intrusion Detection & Prevention Systems (IDPS), Cryptography play an important role. Among all the methods used, encryption has emerged as a critical security measure to safeguard sensitive information. This technique effectively transforms readable data into an encoded format, ensuring

individuals cannot access or interpret the information.

Encryption can be done in several ways. Some include Symmetric Encryption, Asymmetric Encryption, Hash Functions, and Hybrid Encryption. Among the different encryption techniques, the Advanced Encryption Standard (AES) algorithm is widely utilized due to its exceptional security and efficiency. AES has established itself as the standard for encrypting sensitive data across analysing applications, owing to its robust design and adaptability.

Even though the AES algorithm is a very good method for protecting data, intrusions could still occur. Steganography is often used as an additional layer of protection. By hiding information inside other non-secret data, this approach enables hidden communication. Steganography is a crucial technique for secure communications in a variety of applications because, in contrast to encryption, which changes data into an unintelligible format, it aims to hide the information's existence.

There are many forms of Steganography including audio, video, text, and image. Least Significant Bit (LSB) steganography is one of the most widely used methods for embedding secret messages within digital media, particularly images. This technique utilizes the inherent characteristics of digital files to hide information in a manner that is imperceptible to the human eye, thereby facilitating covert communication. One of the important features is, it performs pixel manipulation which is hard to detect.

In this project, we aim to conceal sensitive medical information using Advanced Encryption Standard (AES) and Least Significant Bit (LSB) which acts as two-layer security. We combine the strengths of AES and LSB to achieve flexible key size and scalable encryption.

II. LITERATURE SURVEY

The Internet of Things (IoT) is emerging at a historic juncture due to the rapid expansion of computer technology, mobile communication networks, embedded technology, and the Internet. Global perception, dependable transfer, and intelligent information processing are the main characteristics of the Internet of Things. Every day, a lot of text documents are created and distributed via the Internet of Things. Modern technology makes it simple to duplicate and share these papers [1].

However, because to the vulnerabilities of its applications, widespread IoT device deployment may expose consumers to a number of security risks. Malicious users, or attackers, can manipulate the sensing tasks because the IoT's sensing layer's fundamental technology cannot directly control mobile users. Data confidentiality, dependability, integrity, authorized access, and other factors present security threats for the Internet of Medical Things (IoMT). Patients' basic diagnosis and treatment information are included in electronic medical records, the security of which must be rigorously ensured to prevent patient privacy violations [2].

The owner will no longer have total control over their data once it is stored on the cloud. Therefore, to ensure data security and privacy, a flexible access control mechanism is required [3]. Due to its many advantages, it is becoming more and more well-liked every day. However, new privacy and data security concerns are now top of mind for both service providers and customers. To achieve our goal, we first encrypt the data before concealing it in an image [4].

Published in 1977, the Advanced Encryption Standard (AES) is a symmetric-key cryptographic method. Three key lengths—128, 192, or 256 bits—as well as a block size of 128 bits are supported by AES. It replaces the symmetric key cryptography technique known as the Data Encryption Standard (DES) [5].

The cover picture can conceal private data, including voice, video, photographs, and messages. The primary goal is to use the Least Significant Bit (LSB) technique to conceal the secret message or image inside the image [6].

Reading the cover image and the confidential information is the first step. With the aid of the LSB encoder, the data is then hidden within the cover image. AES encryption is applied to the resulting stego image, causing all of the pixels to become jumbled, in order to safeguard the data from the hacker [7].

The right cover image should be chosen in order to conceal the secret message. Since there is a danger that information could be lost during communication, it is crucial to conceal information in digital images utilizing lossless compression algorithms. The LSB-based technique is a straightforward method where the least important parts of the cover picture are embedded with message bits. This method conceals the hidden message by using the least important portion of the cover image [8].

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:04/April-2025 Impact Factor- 8.187 www.irjmets.com

III. METHODOLOGY

The goal of this project is to create a robust picture steganography system that employs Least Significant Bit (LSB) steganography to conceal the data in an image after it has been encrypted using the Advanced Encryption Standard (AES) method. The sensitive data is secured using two layers of protection, namely AES encryption and steganography.

Implementation of AES Encryption Algorithm:

One popular symmetric block encryption algorithm for protecting sensitive data is the Advanced Encryption Standard (AES) algorithm. The National Institute of Standards and Technology, or NIST, released it in 2001. Because AES allows key lengths of 128, 192, or 256 bits and operates on fixed block sizes of 128 bits, it may be tailored to meet a variety of security needs.

Structure of AES:

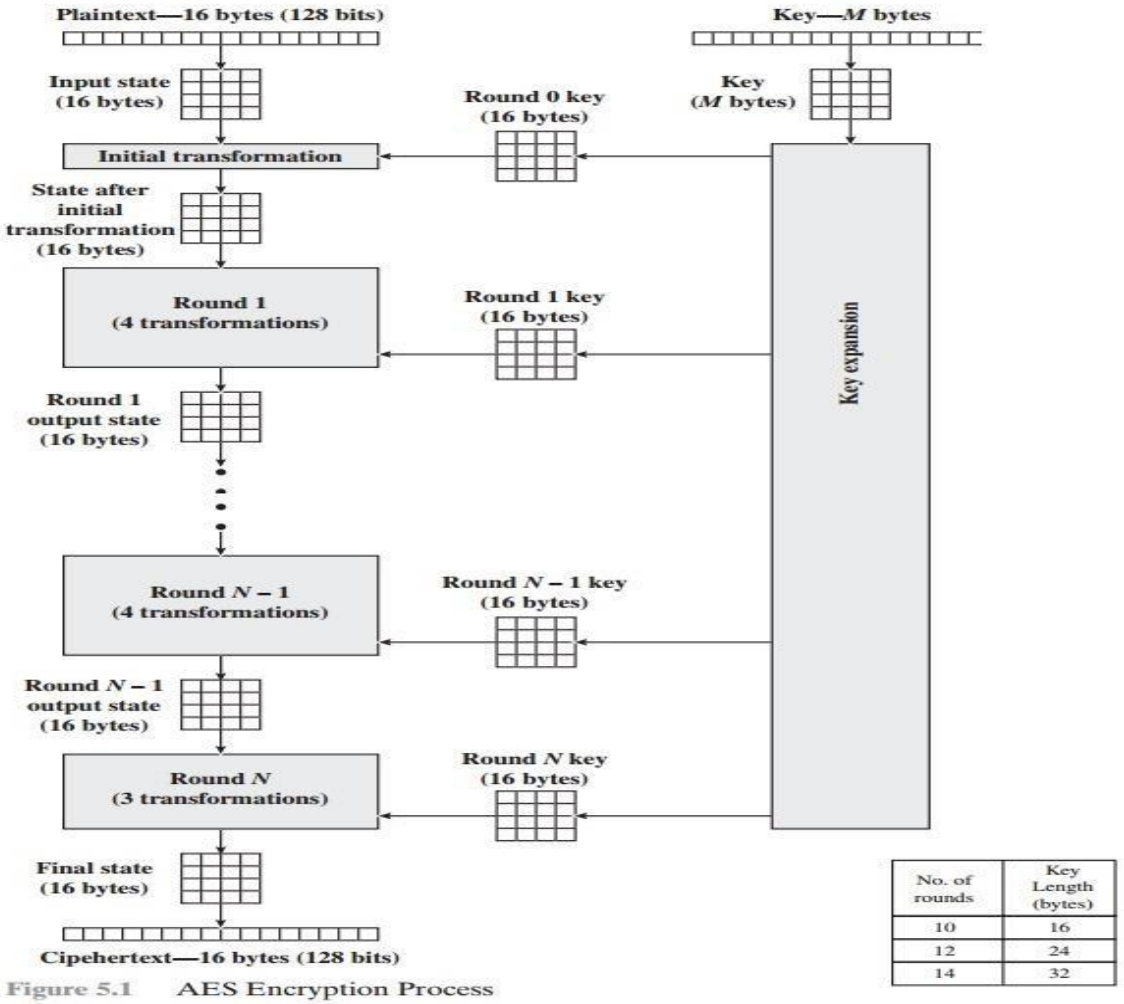


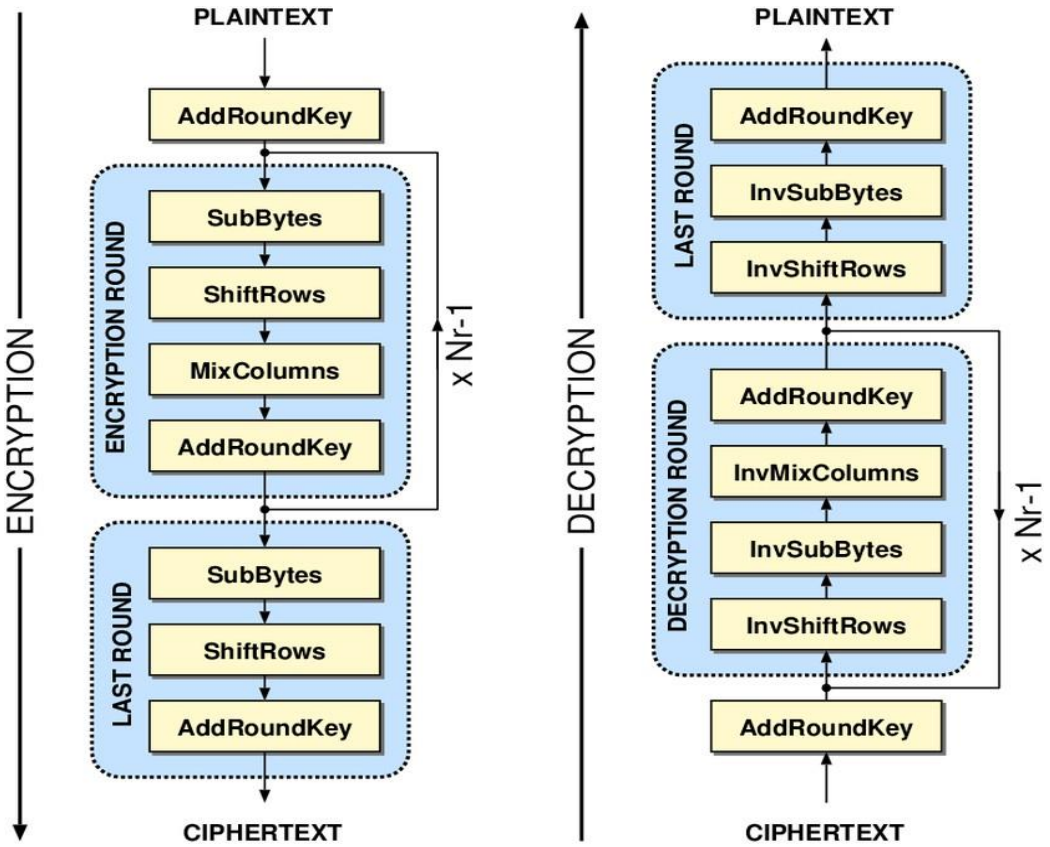
Figure 5.1 AES Encryption Process

The structure of the Advanced Encryption Standard (AES) algorithm is based on a 16-byte (128-bit) state array, which serves as the input data. This state array undergoes an initial transformation before being processed through a series of rounds. The number of rounds is determined by the key size: 10 rounds for a 128-bit key, 12 rounds for a 192-bit key, and 14 rounds for a 256-bit key.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:04/April-2025 Impact Factor- 8.187 www.irjmets.com

There are four different transformations in each round, with the exception of the last one: Add Round Key, Mix Columns, Shift Rows, and Substitute Bytes. The Mix Columns phase is skipped in the final round, which only includes three transformations. A key expansion procedure is used to create a sequence of round keys, each 128 bits in size, from the master key during the encryption process. To guarantee that the key changes from one encryption round to the next, these round keys are used in each round. Following the completion of all designated rounds, the AES algorithm generates a 16-byte ciphertext as its final result.

AES Encryption and Decryption Process:



The four Transformations in AES Encryption are:

1.Substitute Bytes 2.ShiftRows 3.MixColumns

4.Add Round Key

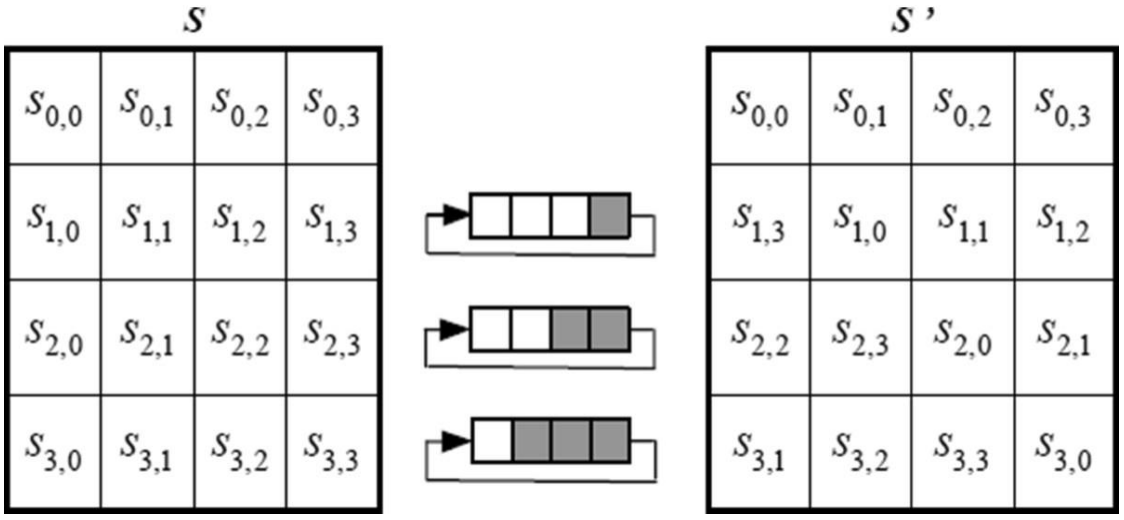
Substitute Bytes:

In this step, each individual value in the state array is replaced by a corresponding value according to a substitution table, also known as a lookup table. This lookup table is structured as a 16 x 16 matrix, which contains predefined substitution values for each possible input byte.

Shift Rows:

In this stage, the state array's rows are moved to the left by different amounts. Row 0 stays the same, Row 1 moves one byte to the left, Row 2 moves two bytes, and Row 3 moves three bytes.

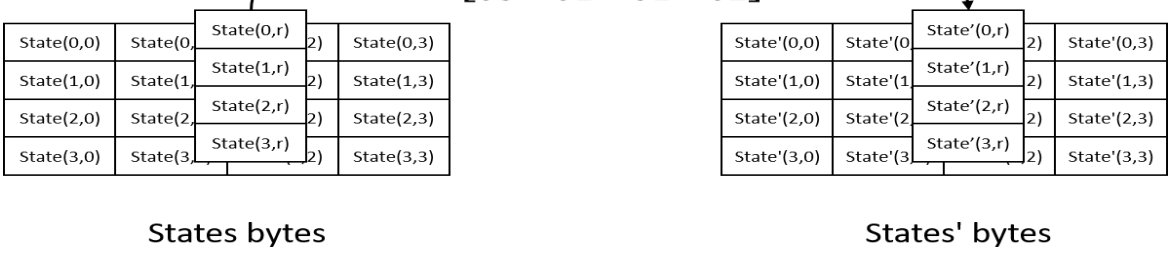
e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:04/April-2025 Impact Factor- 8.187 www.irjmets.com



Mix Columns:

In this step matrix multiplication will take place with a pre-defined matrix.

$$[S'] = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} * [S]$$



States bytes States' bytes

Add Round Key:

In this step an XOR operation takes place between the state array and the round key.

In this Encryption process all rounds consists of the above four transformations but the last round consists of only 3 rounds. Last round did not contain mix column transformation.

The four Transformations of Decryption are:

1.Add Round key 2.InvMixColumns 3.InvSubBytes 4.InvShiftRows

AddRoundKey:

The first stage in the decryption procedure is to apply the AddRoundKey transformation, which involves bitwise XORing the ciphertext with the last round key. Setting the groundwork for the ensuing changes, this phase essentially reverses the previous encryption round.

Inverse mix columns:

The MixColumns operation that was used during encryption is reversed in this stage. It restores the original byte values prior to encryption by multiplying each state array column by a predetermined matrix.

Inverse Substitute Bytes:

This step replaces every byte in the state array with the value that corresponds to it in the inverse substitution table. This change restores the original byte values by undoing the SubBytes operation that was carried out during encryption.

Inverse ShiftRows:

This step undoes the results of the encryption-related MixColumns action. Prior to encryption, the original byte values are restored by multiplying each state array column by a predetermined matrix.

Four transformations are used in the decryption process: Inverse SubBytes, Inverse ShiftRows, Inverse MixColumns, and AddRoundKey. The last round is the only one that does not use these transformations. The final

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:04/April-2025 Impact Factor- 8.187 www.irjmets.com

round only includes Inverse SubBytes, Inverse ShiftRows, and AddRoundKey; the Inverse MixColumns transformation is not used. The integrity of the original data is preserved when the encryption processes are successfully reversed during the decryption process thanks to this structure.

Implementation of LSB Steganography:

Steganography, which means "covered writing" or "hidden writing," is a term derived from two Greek words: "stegos," which means "to cover," and "grayfia," which means "writing."

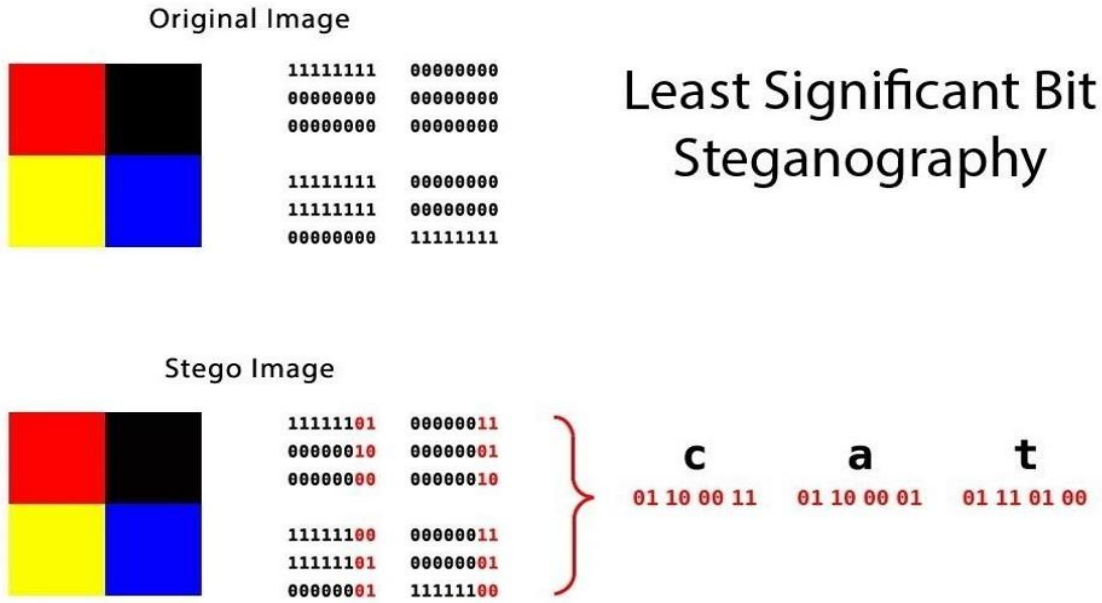
Steganography is the process of encoding hidden information into a text, audio, video, or image file in order to conceal it. This technique is used to defend sensitive or confidential information against malevolent attacks.

Pixels that make up an image typically refer to the color of that specific pixel. These pixel

values range from 0 to 255 in a grayscale (black and white) image, where 0 represents black and 255 represents white.

The acronym for Least Significant Bit is LSB. The goal of LSB embedding is to conceal our confidential information in the image's final bit value. There won't be much noticeable color change if we alter a pixel's final bit value. 0 is black, for instance. Since the value is still black, albeit a lighter hue, changing it to 1 won't have much of an impact. The encrypted image will not differ significantly from the original image.

The Encoding and Decoding process of LSB Steganography:



1. **Select the Cover Image**: The first step is to choose a suitable cover image, which will serve as the carrier for the hidden data. The image should have sufficient pixel capacity to accommodate the secret message without any distortion.

2. **Prepare the Secret Data**: The secret message should be converted to its equivalent binary representation to embed it in an image.

3. **Embed the Data**: The embedding process involves iterating through the pixels of the cover image. For each pixel, the least significant bit (LSB) of the pixel's colour value is replaced with a bit from the secret message. This is typically done in the following manner:

- For each pixel, extract the colour values (usually RGB: Red, Green, Blue).
- Replace the LSB of each colour component with a bit from the secret message, moving sequentially through the message until all bits are embedded or the image's pixel capacity is reached.

4. **Generate the Stego Image**: After embedding the secret data, a new image, known as the stego image, is created. This image contains the original pixel values with the modified LSBs, effectively hiding the secret message within the image.

5. **Data Extraction (Decoding)**: To retrieve the hidden message, the process is reversed. The recipient extracts the LSBs from the pixels of the stego image, reconstructing the binary representation of the original

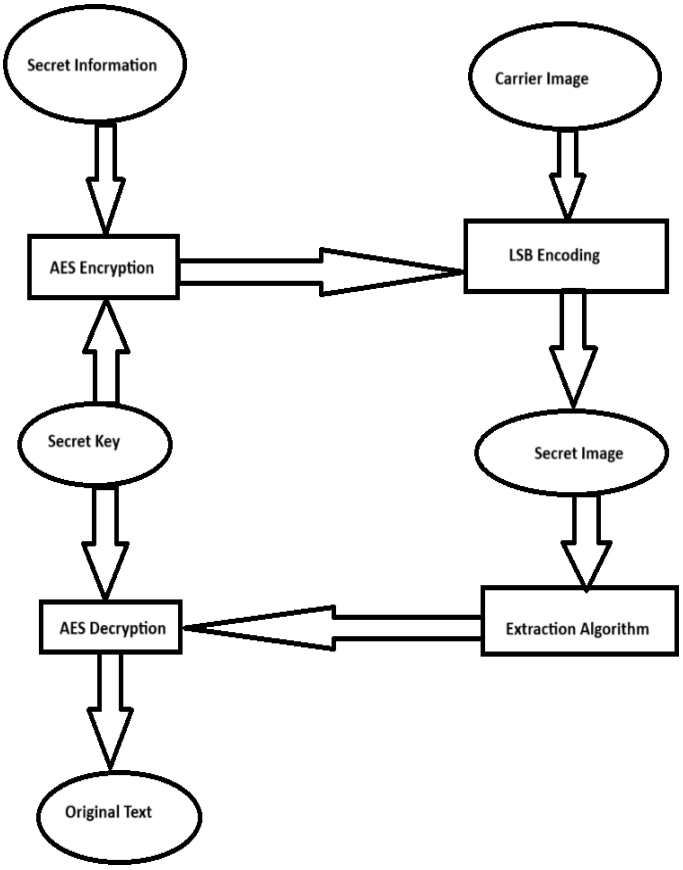
e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:04/April-2025 Impact Factor- 8.187 www.irjmets.com

secret message. This binary data is then converted back into its original format to get the secret message.

Integration of AES and LSB Steganography:

The input to the AES algorithm consists of ciphertext, which is first converted into its equivalent binary representation for embedding into a digital image. Following the encoding process, the modified image is subjected to decoding, during which the least significant bits (LSBs) of the pixel values are extracted and converted back into binary format to retrieve the original ciphertext. This ciphertext is then processed through the AES decryption algorithm to recover the original secret message.

Flow Diagram:



IV. RESULT

Collected Data from an IoT Device:

```
[8]: import pandas as pd

     # Load the Excel file
     excel_file = 'Project_Input.xlsx'

     # Read all sheets
     xls = pd.ExcelFile(excel_file)

     # Loop through each sheet and save as CSV
     for sheet_name in xls.sheet_names:
         df = pd.read_excel(xls, sheet_name)
         # Save each sheet as a separate CSV
         csv_file = f'C:\\Users\\Varanasi Tejaswi\\{sheet_name}.csv'
         df.to_csv(csv_file, index=False)
```

```
[10]: df=pd.read_csv('Sheet1.csv')
      df.head()
```

[10]:

	Name	Age	Height	Gender	Body Weight	BMI	Muscle rate	Body Water	Visceral Fat	Subcutaneous Fat	Bone Mass	BMR	Protein Mass
0	Manoj	20	170	Male	52.70	18.2	44.3	64.81	3	10.0	2.5	1369	19.6
1	Murthy	50	179	Male	63.25	19.7	45.3	50.28	4	23.0	3.1	1387	15.1
2	Bablu	14	145	Male	36.50	23.3	21.4	68.39	10	23.0	1.3	861	12.3
3	Jagannadham	54	170	Male	63.45	21.9	47.4	54.63	10	17.5	2.6	1363	17.2
4	Pavan	19	183	Male	83.70	24.9	57.5	54.57	6	20.4	3.2	1805	16.5

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:07/Issue:04/April-2025 Impact Factor- 8.187 www.irjmets.com

```
[85]: column_list = df['Age'].tolist()
      column_list

[85]: [20, 50, 14, 54, 19, 56, 77, 48, 49, 14]
```

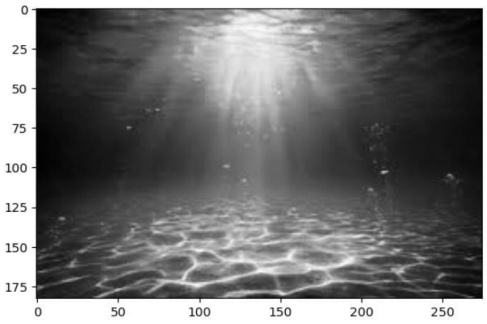
Taking input from the csv file

AES Encryption

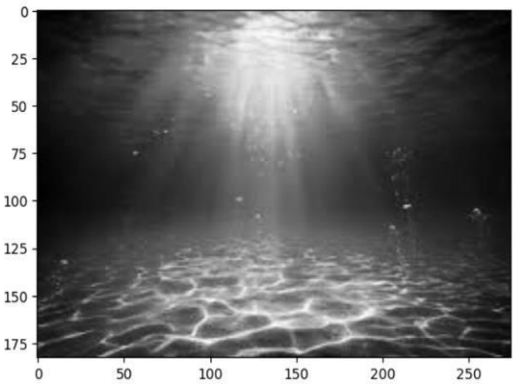
```
master_key = 0x2b7e151628aed2a6abf7158809cf4f3c  # Example 128-bit key
decrypted_list=[]
for i in column_list:
    plaintext = i
    aes = AES(master_key)
    ciphertext = aes.encrypt(plaintext)
    decrypted_list.append(ciphertext)
    print(f"Ciphertext: {ciphertext:032x}\n")

Ciphertext: 56980cee98e1b9b47a0f6d0f71a73b75

Ciphertext: ee484b274efdb836cea9b09b9039226e

Ciphertext: 861c5964e3c9dc95c6303f12bad10d9c

Ciphertext: 83d595f3fd1602ce40205165f1b824b5

Ciphertext: 3afb2a4eb995fda13ff080b05eae6cfd

Ciphertext: 38c30f133cf236f77b8f4bb6ac2964a9

Ciphertext: dfacd83f20c4fa135f6746df0b4188a2

Ciphertext: 3ae855a18daadf9a6c0508a57fb4ad50

Ciphertext: edc33dbb416269f33bf2508fcbd68bf5

Ciphertext: 861c5964e3c9dc95c6303f12bad10d9c
```

Performing steganography on the Figure 1 and the Figure 2 is the encoded image of the given input

```
[54]: import cv2
      import numpy as np
      from PIL import Image
      import matplotlib.pyplot as plt
      img=cv2.imread('img1.png',0)
      plt.imshow(img,cmap='gray')

[54]: <matplotlib.image.AxesImage at 0x1d07b88a8d0>
```




```
[64]: plt.imshow(img, cmap='gray')
      plt.savefig('Downloads\\Image.jpg', dpi=300)
```



The decryption of the decoded bits from the encoded images

```
[80]: res=[]
      if __name__ == "__main__":
          for i in decrypted_list:
              master_key = 0x2b7e151628aed2a6abf7158809cf4f3c  # Example key
              aes = AES(master_key)
              cipher=i
              # Decrypt the ciphertext
              decrypted_text = aes.decrypt(i)
              res.append(decrypted_text)
              print(f"Decrypted text: {decrypted_text:032x}")  # Print as hex

Decrypted text: 00000000000000000000000000000014
Decrypted text: 00000000000000000000000000000032
Decrypted text: 0000000000000000000000000000000e
Decrypted text: 00000000000000000000000000000036
Decrypted text: 00000000000000000000000000000013
Decrypted text: 00000000000000000000000000000038
Decrypted text: 0000000000000000000000000000004d
Decrypted text: 00000000000000000000000000000030
Decrypted text: 00000000000000000000000000000031
Decrypted text: 0000000000000000000000000000000e

[82]: print(res)

[20, 50, 14, 54, 19, 56, 77, 48, 49, 14]
```

V. CONCLUSION

In this research, we successfully combined Least Significant Bit (LSB) steganography with Advanced Encryption Standard (AES) encryption to provide a dual-layer security technique for data protection. Improving the integrity and confidentiality of sensitive data was our primary objective. The first thing we did was encrypt the

data using the powerful AES algorithm, which provides a high degree of protection against unwanted access. After encryption, we used LSB steganography to insert the data inside digital media files, so obfuscating its existence and enhancing security.

Our implementation's outcomes demonstrate that, in comparison to conventional techniques, this dual-layer strategy greatly improves data security. AES encryption makes sure that without the right decryption key, the data cannot be read, even if it is intercepted. Simultaneously, LSB steganography successfully conceals the encrypted data inside innocuous files, decreasing the likelihood that it would be discovered or viewed without permission.

While this project has yielded promising results, we must acknowledge certain limitations. The effectiveness of LSB steganography can be affected by the type of media used and the volume of data being hidden, which may impact the quality of the cover media. Additionally, further research is necessary to investigate the resilience of this method against various steganalysis techniques.

Future research could concentrate on strengthening the steganographic layer by refining the embedding process and investigating the incorporation of alternative encryption algorithms to assess how well they perform in comparison to AES. All things considered, this research advances the field of data security by providing a fresh method of data protection and highlighting the significance of multi-layered security tactics in protecting private data in a world that is becoming more and more digital.

VI. REFERENCES

[1] Text Data Security and Privacy in the Internet of Things: Threats, Challenges, and Future Directions by Umair Khadam, Muhammad Munwar Iqbal, Meshrif Alruily, Mohammed Al Ghamdi, Muhammad Ramzan, Sultan H. Almotiri

[2] W. Li, Z. Su and K. Zhang, "Security Solutions for IoT-Enabled Applications Against the Disease Pandemic," in IEEE Internet of Things Magazine, vol. 4, no. 4, pp. 100-106, December 2021, doi: 10.1109/IOTM.001.2100056. keywords: {COVID-19; Temperature sensors; Pandemics; Smart cities; Simulation; Smart homes; Sensors}

[3] S. K. S. Raja, A. Sathya and L. Priya, "A Hybrid Data Access Control Using AES and RSA for Ensuring Privacy in Electronic Healthcare Records," 2020 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS), Chennai, India, 2020, pp. 1-5, doi: 10.1109/ICPECTS49113.2020.9337051. keywords: {Access control; Cloud computing; Data privacy; Organizations; Registers; Outsourcing; Engines; Attributes; Authorized Cloud; Semi-Trusted Authority; Public Cloud; Designation; Organization; Audit request; Data Owners; Rule Based Engine},

[4] An Approach for Enhancing Security of Cloud Data using Cryptography and Steganography with E-LSB Encoding Technique Mohammad Obaidur Rahman, Muhammad Kamal Hossen†, Md. Golam Mo

[5] P. P. Bandekar and G. C. Suguna, "LSB Based Text and Image Steganography Using AES Algorithm," 2018 3rd International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2018, pp. 782-788, doi: 10.1109/CESYS.2018.8724069. keywords: {Image steganography; LSB; Advanced Encryption Standard; PSNR Ratio; MSE; OTP},

[6] M. R. Islam, A. Siddiqa, Md. Palash Uddin, A. K. Mandal and M. D. Hossain, "An efficient filtering based approach improving LSB image steganography using status bit along with AES cryptography," 2014 International Conference on Informatics, Electronics & Vision (ICIEV), Dhaka, Bangladesh, 2014, pp. 1-6, doi: 10.1109/ICIEV.2014.6850714.

[7] keywords: {Ciphers; Image color analysis; PSNR; Encryption; Histograms; Image Steganography; Filtering Algorithm; AES Cryptography; Conceal of Message; LSB Image Steganography} LSB BASED TEXT AND IMAGE STEGANOGRAPHY USING AES ALGORITHM.

[8] Priya Paresh Bandekar1 and Suguna G C2 An analysis of LSB Based Image Steganography Techniques K.Thangadurai and G.Sudha Devi,