Journal of Information Systems Engineering and Management
2025, 10(16s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Shielded conveyance of Gut Medical Data in IoMT Networks Using Dual Mode Cryptography and Image Steganography Techniques

Shravani Amar[1], Shaik Hussain Shaik Ibrahim[2], P. Anitha[3]

[1]*Research scholar, Department of Information Technology, Malla Reddy University, Hyderabad, India, 2232CS020005@mallareddyuniversity.ac.in.*

[2] *Associate Professor, Department of Computer Science & Engineering, Malla Reddy University, Hyderabad, India, shaikhussain2207@gmail.com.*

[3] *Associate Professor, Department of Computer Science & Engineering (Data Science), Malla Reddy University, Hyderabad, India, dranitha.p@mallareddyuniversity.ac.in.*

ARTICLE INFO

Received: 08 Dec 2024

Revised: 28 Jan 2025

Accepted: 09 Feb 2025

ABSTRACT

Gut diseases impact millions of individuals worldwide, underscoring the urgent need for effective security measures in healthcare systems to protect vulnerable medical information. With the increasing adoption of Internet of Medical Things (IoMT) devices for monitoring and managing GI conditions, ensuring the confidentiality and integrity of patient data is paramount. Current security protocols often struggle with issues such as high computational demands and susceptibility to advanced cyber-attacks, highlighting the need for more robust and efficient solutions. To address these challenges, Two-Level Encryption with Gastrointestinal Image Steganography (TLE-GIS) is proposed that enhances data security through a multi-layered approach. The process begins with Data partitioning, into even and odd parts. Then, the even part of the data is encrypted using Networking and Cryptography library (NaCL), known for its high-level encryption functions and ease of use. The odd part is encrypted using Optimal Asymmetric Encryption Padding (OAEP), a padding scheme that adds randomness to encryption, improving security by mitigating specific attack vectors. After encryption, the even and odd parts are combined into a single encrypted message. The combined encrypted message is then hidden within a medical image using a technique called Least Significant Bit - Iterative Wavelet Transform (LSB-IWT), which offers improved quality and reduced distortion. The simulation results shows that the proposed TLE-GIS approach resulted in improved security standards compared to traditional methods.

Keywords: Gut diseases, Internet of Medical Thing, Data security, Networking and Cryptography library, Optimal Asymmetric Encryption Padding, Least Significant Bit, Iterative Wavelet Transform, Data hiding

1.INTRODUCTION

The GI diseases are prevalent worldwide, impacting a significant portion of the global population. According to the World Health Organization (WHO), GI disorders account for a substantial number of hospital visits and health care expenditures [1]. Common GI diseases include irritable bowel syndrome, inflammatory bowel disease, peptic ulcer disease, and gastrointestinal cancers. In the United States alone, it is estimated that around 60 to 70 million people are affected by various GI conditions [2]. Furthermore, GI cancers such as colorectal cancer are among the leading causes of cancer-related deaths globally, with the American Cancer Society estimating over 150,000 new cases of colorectal cancer in the U.S. annually. The high prevalence of these diseases underscores the need for effective management and innovative approaches to treatment and patient data security.

With the rise of IoMT devices in monitoring and managing GI diseases, ensuring the security and privacy of patient data has become crucial. IoMT devices, such as wearable sensors and smart pills, continuously collect sensitive health

information. This data, when transmitted over networks [3], is vulnerable to interception and unauthorized access. To safeguard patient privacy and maintain data integrity, it is essential to implement robust security measures. Steganography, the practice of concealing information within other data, offers a promising solution. By embedding patient data within innocuous digital content, such as medical images or text files, steganography can provide an additional layer of security, making it harder for unauthorized parties to detect and exploit the sensitive information.

Current security methods for IoMT data often face several challenges [5]. Traditional encryption techniques, while effective in protecting data during transmission, can be computationally intensive and may introduce delays in real-time monitoring systems. Additionally, the increasing sophistication of cyber-attacks [6] poses a constant threat, as hackers continuously develop new methods to breach security protocols. Many existing solutions [7] also suffer from issues related to scalability and interoperability, making them less effective in diverse and evolving healthcare environments. Furthermore, standard encryption methods alone may not be sufficient to address the specific security needs of IoMT devices, which require a multi-layered approach to safeguard both data in transit and data at rest.

To enhance the security of IoMT systems [8], a combined approach utilizing both data encryption and image steganography methods is highly beneficial. Joint data encryption [9] and steganography [10] can provide a comprehensive security solution by not only encrypting the data but also embedding it within seemingly innocuous digital images or other files. This dual-layered approach can obscure the presence of sensitive information and protect against unauthorized access. For instance, encrypted GI data can be hidden within medical images or other non-sensitive files, making it less detectable and less likely to be targeted by cybercriminals. By integrating these methods, healthcare providers can better safeguard patient data, ensure privacy, and maintain the integrity of IoMT systems, ultimately enhancing the overall security of GI disease management technologies. The novel contributions of the proposed methodology for IoMT-based GI healthcare security are as follows:

- To develop dedicated security model tailored specifically for IoMT-based GI healthcare systems, addressing a significant gap in the literature where no existing solutions currently focus on securing sensitive GI-related data.
- A unique hybrid encryption approach is introduced, combining NaCL with OAEP provides a dual layer of protection that effectively mitigates various attack vectors.
- The methodology incorporates data-hiding strategy that utilizes LSB-IWT, which enhances data security while preserving the quality and integrity of medical images.

The rest of the paper is organized as follows: Section 2 provides a comprehensive survey of existing security methods for IoMT-based healthcare systems. Section 3 details the proposed methodology, while Section 4 presents the experimental results. Finally, Section 5 concludes the paper with a summary of findings and future research directions.

2. LITERATURE SURVEY

M. Elhoseny et al.[11] proposed a Guarded Medical Data Transmission Model utilizing Rivest-Shamir-Adleman (RSA) encryption, Advanced Encryption Standard (AES) encryption, and Discrete Wavelet Transform (DWT). The methodology combined RSA for secure key exchange, AES for robust data encryption, and DWT for data transformation to enhance security and efficiency. This approach aimed to provide a multi-layered security model to protect sensitive medical data transmitted over IoT networks. However, the model encountered drawbacks such as increased computational complexity due to the combined use of multiple encryption and transformation techniques. Hameed et al. [12] introduced medical steganography technique based on Adversarial Neural Cryptography with SHA-256 (ANC-SHA-256) and digital signatures using LSB replacement. While this approach faced challenges including computational complexity and potential degradation in image quality due to the steganographic process. Ramyashree et al. [13] presented CrypticCare, a strategic approach to telemedicine security incorporating LSB and Discrete Cosine Transform (DCT) steganography. The method combined LSB replacement with DCT to enhance patient data protection by embedding secret information in medical images. Despite its advantages, the approach had limitations such as susceptibility to attacks targeting the steganographic techniques and potential difficulties in ensuring data integrity and robustness against sophisticated extraction methods.

Latif et al. [14] proposed fragmented approach for securing medical health records in multimodal medical images. This methodology involved segmenting medical images into fragments and applying encryption techniques to each

fragment to ensure data security. While this approach enhanced data protection, it encountered challenges such as increased processing time and complexity in managing and reconstructing fragmented data. Zhang et al. [15] developed a reversibly selective encryption method for medical images based on coupled chaotic maps and steganography. However, this method faced drawbacks such as potential vulnerability to attacks targeting the chaotic maps and challenges in balancing encryption strength with image quality preservation. Ch et al. [16] proposed steganographic framework for securing sensitive healthcare data in telemedicine using convolutional neural networks (CNNs). The framework employed CNNs to enhance the security and robustness of steganographic techniques for protecting healthcare data. Mohammed et al. [17] introduced SecMISS, a secured medical image secret sharing mechanism for smart health applications. The primary drawback was the potential increase in data management complexity and the need for effective reconstruction methods to ensure the usability of the shared medical images.

Tiwari et al. [18] employed Redundant Discrete Wavelet Transform (RDWT)-DCT-Schur decomposition-based watermarking for enhancing the security of medical images, with authentication using Binary Robust Independent Elementary Features (BRIEF) features. However, it faced challenges related to computational overhead and potential impact on image quality, which could affect the practical application of the method. Nadhan et al. [19] proposed enhancing healthcare security in the digital era by safeguarding medical images with lightweight cryptographic techniques in IoT healthcare applications. The primary drawback was the potential trade-off between encryption strength and computational efficiency, which could impact the overall performance of IoT healthcare systems. Sankaranarayanan et al. [20] introduced a color secret sharing protocol for secure transmission of medical images. While the approach enhanced data security, it faced challenges such as increased complexity in managing color information and potential limitations in scalability for large-scale medical image datasets.

Priya et al. [21] proposed a super-resolution deep neural network (SRDNN) based multi-image steganography technique for highly secure lossless image transmission. The primary drawback was the computational intensity and potential complexity in training and deploying SRDNN models for practical use. Huo et al. [22] developed a high-capacity and high-security image steganography network based on chaotic mapping and generative adversarial networks (GANs). The network combined chaotic mapping with GANs to enhance the capacity and security of image steganography. Rezaei et al. [23] explored bio-inspired algorithms for secure image steganography, focusing on enhancing data security and quality in data transmission. Their approach applied bio-inspired algorithms to improve the security and effectiveness of steganographic techniques. Shi et al. [24] proposed a separable privacy-preserving technique based on reversible medical data hiding in plaintext encrypted images using neural networks. The technique aimed to combine reversible data hiding with neural networks to ensure privacy and maintain data integrity. El-Shafai et al. [25] introduced a 3D chaos-based medical image cryptosystem for secure cloud-IoT eHealth communication services. The primary drawback was the complexity of implementing 3D chaotic encryption and potential performance issues in real-time communication scenarios.

3. PROPOSED METHODOLOGY

The novelty of this approach lies in its unique combination of multiple advanced techniques to address the security needs of IoMT-based GI healthcare systems, which has not been previously presented in the literature. This methodology integrates a Dual encryption scheme, using NaCL for high-security encryption and OAEP for enhanced RSA encryption, Linked with a sophisticated data-hiding technique that combines LSB and IWT. The integration of these methods offers a novel solution that not only ensures robust encryption and secure data transmission but also maintains the quality and integrity of medical images, setting it apart from existing security protocols and highlighting its innovative contribution to the field. Figure 1 shows the proposed TLE-GIS operational framework. The detailed operation is given as follows

Step 1: Data Splitting: The input message form patient is split into even and odd parts. This splitting allows for the application of different encryption algorithms to each part of the data.

Step 2: Even Data Encryption: The even data is encrypted using NaCL, which is a modern cryptographic library that provides high-level encryption functions and is known for its ease of use and strong security.

Step 3: Odd Data Encryption: The odd data is encrypted using OAEP, which is a padding scheme used with RSA to provide additional security by adding randomness and preventing certain types of attacks.

Step 4:Combined Encrypted Data: The encrypted even and odd parts are combined into a single encrypted message. This combined encrypted data leverages the strengths of both NaCL and OAEP encryption methods.

Step 5:Data Hiding using LSB-IWT: The combined encrypted message is hidden in a medical image using LSB-IWT. It provides better quality, and less distortion compared to DWT. The encrypted data is embedded into the IWT coefficients, which helps in maintaining the integrity and quality of the medical image while securely hiding the data.

Step 5: Stego-Image Storage: Usually in IoMT applications, this stego-image stored into cloud servers, which securely contains the encrypted data, is then transmitted to the receiver over a secure channel, ensuring that the sensitive patient information remains hidden and protected from unauthorized access.

Step 6: Data Extraction using Inverse IWT-LSB: Upon receiving the stego-image, the receiver applies the inverse IWT-LSB extraction techniques to retrieve the combined encrypted message hidden within the image. This process reverses the data hiding procedure, allowing for the accurate recovery of the encrypted information while preserving the integrity of the medical image.

Step 7: Encrypted Message Splitting: Once the encrypted data is extracted, the receiver separates it back into its even and odd parts. This splitting is necessary to apply the appropriate decryption algorithms to each segment, corresponding to the encryption methods initially used on the sender's side.

Step 8: Data Decryption: The receiver decrypts the even part using the NaCL cryptographic library and the odd part using OAEP decryption procedures. This decryption process involves reversing the encryption algorithms to restore the original data while ensuring that the information remains secure and intact throughout the transmission.

Step 8: Recovered Patient Data: After successfully decrypting both parts, the receiver combines them to reconstruct the original patient data. This recovered data can then be used for medical analysis or decision-making, ensuring that all sensitive information remains confidential and secure from potential cyber threats.

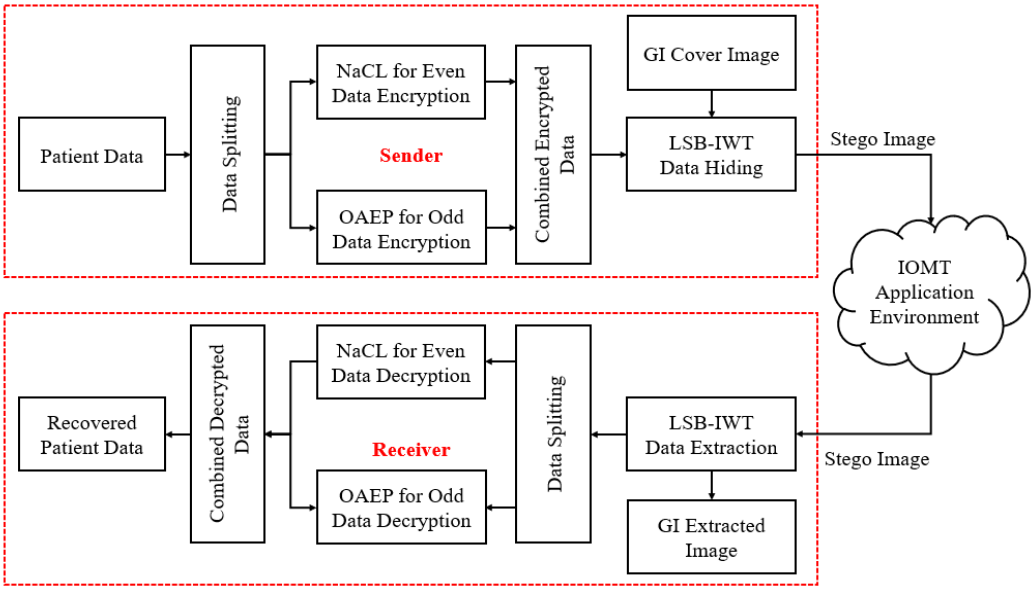

Figure 1. Proposed TLE-GIS Flow Diagram.

3.1 NaCL Cryptography

The NaCL's symmetric encryption method combines the strengths of the XSalsa20 stream cipher for fast, secure encryption and the Poly1305 Message Authentication Code (MAC) for robust message authentication. This combination ensures that encrypted messages remain confidential, intact, and authentic, providing high security and efficiency in cryptographic communications. The use of a unique nonce for each encryption process ensures that the same plaintext results in different ciphertexts, adding an additional layer of security against various types of cryptanalytic attacks.

3.1.1 Key and Nonce Generation

The process begins with the generation of a secret key (k) and a nonce (n). They both serves a critical role in ensuring that the same plaintext encrypted with the same key will produce different ciphertexts, thereby preventing replay attacks and ensuring semantic security.

Key (k): The secret key is a 256-bit (32-byte) value that is shared between the sender and the receiver. The secret key used in the encryption process. It is a randomly generated value:

$$k = \text{random}(32) \quad (1)$$

Nonce (n): The nonce is a unique, randomly generated 192-bit (24-byte) value for each message. It ensures that the encryption process produces unique ciphertexts for the same plaintext:

$$n = \text{random}(24) \quad (2)$$

The nonce is expanded internally in the encryption process to fit the requirements of the XSalsa20 stream cipher, which uses a 512-bit nonce internally derived from the provided 192-bit nonce.

3.1.2 XSalsa20 Stream Cipher

XSalsa20 is a stream cipher derived from Salsa20; a cryptographic algorithm designed by Daniel J. Bernstein. The XSalsa20 cipher extends the nonce length to 192 bits, providing a higher level of security by allowing more unique nonces and thereby more unique ciphertexts for the same plaintext.

Nonce Expansion: The nonce provided (192 bits) is expanded to 512 bits for use within the XSalsa20 cipher. This expanded nonce is used to create the initial state for the XSalsa20 cipher, which is then used to generate the key stream for encryption.

Key Stream Generation: The XSalsa20 generates a pseudorandom key stream S of the same length as the plaintext m. The generation of the key stream depends on both the k and the expanded n, making it unique for each encryption session. Mathematically, the key stream S is generated as follows:

$$S = \text{XSalsa20}(k, n, L) \quad (3)$$

Here, L is the length of the plaintext m.

3.1.3 Encryption Process

Once the key stream S is generated, the encryption process involves performing a bitwise XOR operation between the plaintext m and the key stream S to produce the ciphertext c. This process ensures that the ciphertext is a scrambled version of the plaintext, which cannot be deciphered without the key and the nonce.

$$c = m \oplus S \quad (4)$$

Here, c is the ciphertext, m is the plaintext, S is the key stream generated by XSalsa20. The XOR operation ensures that even small changes in the plaintext or key stream result in significantly different ciphertexts, providing strong security.

3.1.4 Poly1305-MAC

After encrypting the plaintext, NaCL ensures data integrity and authenticity by appending a Poly1305 MAC to the ciphertext. Poly1305 is a high-speed MAC designed to work efficiently with XSalsa20. According to MAC generation, the Poly1305 takes the ciphertext c and a one-time key derived from the shared secret key k and nonce n to produce a 128-bit authentication tag t. This tag provides a cryptographic checksum of the ciphertext, ensuring that any unauthorized modification of the ciphertext can be detected.

$$t = \text{Poly1305}(k, c) \quad (5)$$

Here, t is the MAC, k is the secret key, c is the ciphertext.

3.1.5 Final Encryption and Decryption

Encryption: The final encrypted message consists of the concatenation of the MAC with t and the ciphertext c:

$$\text{encrypted\_message} = t \parallel c \quad (6)$$

This concatenated output is what is transmitted to the receiver. The inclusion of the MAC ensures that any tampering or alteration of the ciphertext during transmission can be detected by the receiver. The decryption process in NaCL involves two main steps: verifying the MAC and decrypting the ciphertext.

MAC Verification: Upon receiving the encrypted message, the receiver splits the MAC with t and the ciphertext c. The receiver then computes the MAC of the received ciphertext using the shared key k and the nonce n and compares it with the transmitted MAC with t. If the MACs match, the ciphertext is considered authentic and has not been tampered with. If the MAC verification fails, the message is rejected, and the receiver assumes the message has been altered or is fraudulent.

Decryption: Once the MAC is verified, the receiver proceeds to decrypt the ciphertext. The decryption involves generating the same key stream S using the XSalsa20 cipher with the same nonce n and key k. The plaintext m is then recovered by performing a bitwise XOR operation between the ciphertext c and the key stream S.

$$m = c \oplus S \tag{7}$$

This operation reverses the encryption process, yielding the original plaintext m.

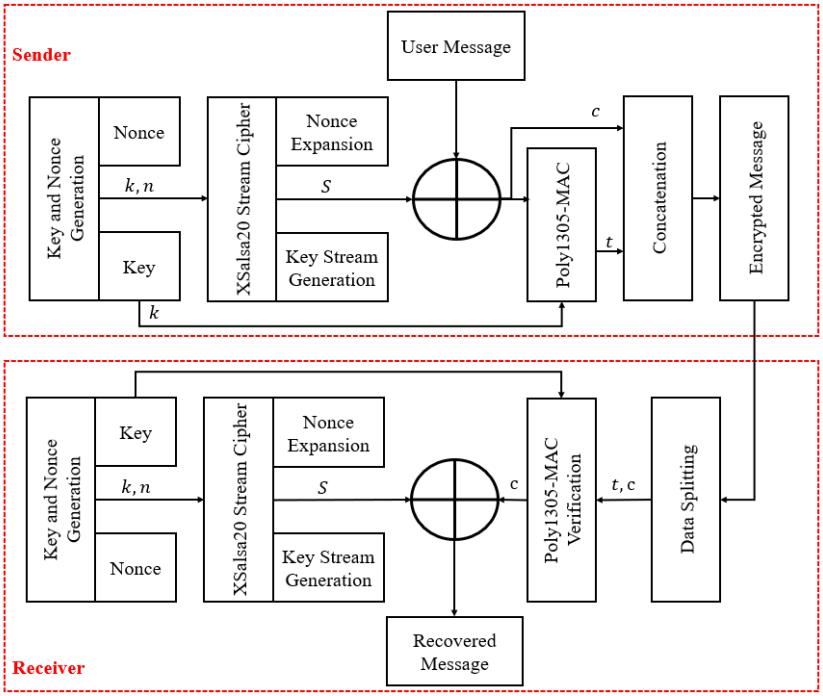

Figure 2. Proposed NACL Flow Diagram.

3.2 OAEP Cryptography

The OAEP padding scheme significantly enhances the security of RSA encryption by introducing randomness and padding, making it resistant to various cryptographic attacks. The combination of RSA and OAEP provides a secure method for encrypting sensitive data, ensuring both confidentiality and integrity.

3.2.1 RSA Keys Generation

Before applying OAEP, RSA keys must be generated. RSA keys are required to perform the encryption and decryption operations securely. RSA key generation involves creating a pair of keys (public and private) that are used for encrypting and decrypting messages. The RSA key pair consists of a public key and a private key. Here, Public key is composed of the modulus n and the public exponent e. The private key is composed of the modulus n and the private exponent d. These keys are generated as follows:

Modulus ($\boldsymbol{n}$): The product of two large prime numbers p and q.

$$n = p \times q \tag{8}$$

Public Exponent (e) : A value typically chosen to be 65537 for its properties that balance security and computational efficiency.

Private Exponent (d): The multiplicative inverse of e modulo $\phi(n)$, where $\phi(n) = (p-1)(q-1)$ is Euler's totient function.

$$e \times d \equiv 1 \, (mod \, \phi(n)) \tag{9}$$

Using these components, the public key is (n, e) and the private key is (n, d).

3.2.2 OAEP Padding

The OAEP scheme combines the plaintext with randomness to produce a padded message suitable for RSA encryption. This scheme consists of several steps to securely pad the plaintext M before encryption. Figure 3 shows the proposed OAEP flow diagram, which considers message from user (pliantext) as input and generates encoded message.

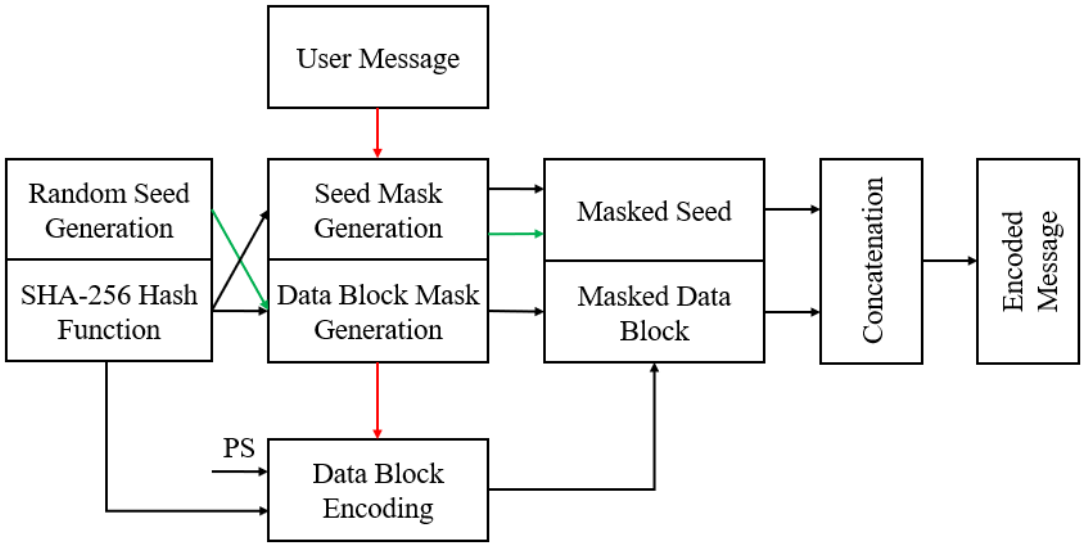

Figure 3. Proposed OAEP Flow Diagram.

Encoding the Message: To ensure security, plaintext M must be padded and randomized. This prevents certain cryptographic attacks and ensures that each encryption operation produces a unique ciphertext, even for identical plaintexts. According to message representation, the plaintext message M is first converted into an octet string m of a specified length. Then perform random seed generation, where seed r of a length based on the length of the output of the chosen hash function, such as SHA-256. The seed introduces randomness to ensure that the same message results in different ciphertexts when encrypted multiple times.

$$r = \text{random}(h) \tag{10}$$

Here, h is the length of the output of the SHA-256 hash function.

Mask Generation Function (MGF): The OAEP uses two MGFs, which ensures that the randomness in the padding process is unpredictable and consistent. This unpredictability prevents attackers from deducing patterns in the encrypted message. The MGF1 is a deterministic algorithm that takes an input and a desired output length and produces a pseudo-random output of the specified length.

$$seedMask = \text{MGF1}(m, h) \tag{11}$$

$$dataBlockMask = \text{MGF1}(r, k - h - 1) \tag{12}$$

Here, m is the message, h is the hash length, k is the RSA modulus size in bytes.

Data Block Encoding: Concatenate the empty data such as padding string (PS), the message m, and a hash of a label to form a data block DB. The PS is used to ensure the data block has the required length and structure.

$$DB = PS \parallel Hash(label) \parallel m \tag{13}$$

Here ‖ denotes concatenation.

Masked DB and Seed: Masking ensures that the padding and randomness are securely combined with the message, making it more resistant to attacks. The use of masks hides the actual data and padding patterns from potential attackers. XOR the data block DB with the data block mask to create a masked data block (maskedDB). Similarly, XOR the random seed r with the seed mask to create a masked seed (maskedSeed).

$$maskedDB = DB \oplus dataBlockMask \quad (14)$$

$$maskedSeed = r \oplus seedMask \quad (15)$$

Concatenation to Form Encoded Message: The encoded message (EM) ensures that the padding and randomness are incorporated into the message before encryption. This step finalizes the preparation of the message for secure RSA encryption. Concatenate the masked seed and the masked data block to form the encoded message EM.

$$EM = maskedSeed \parallel maskedDB \quad (16)$$

3.2.3 RSA Cryptography

Figure 4 illustrates the flow of the proposed RSA cryptography system, starting with an OAEP-encoded input message. This encoded message undergoes RSA encryption to produce a secure ciphertext, ensuring confidentiality during transmission. Upon receiving the ciphertext, RSA decryption is performed to retrieve the OAEP-encoded message. The process guarantees that the encoded message remains protected from unauthorized access. Finally, the decoded message is extracted, completing the secure communication cycle.

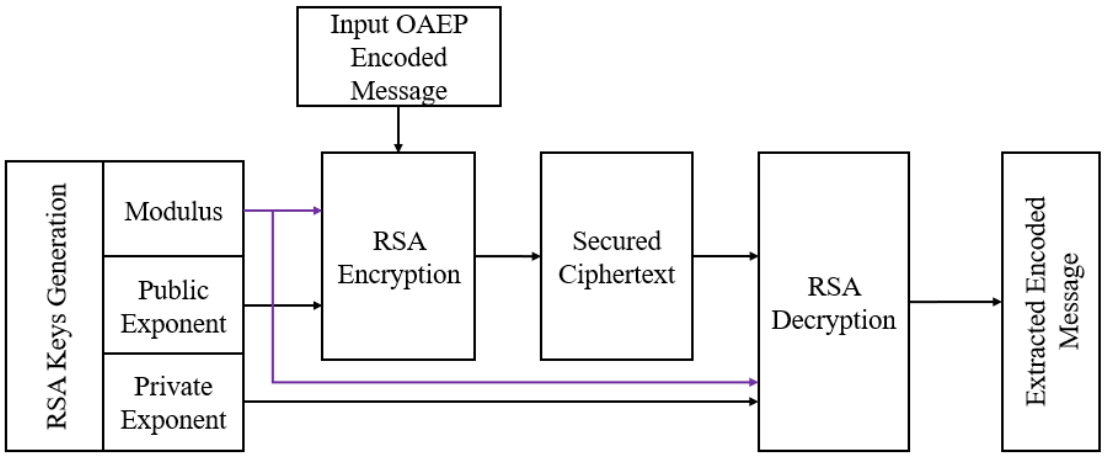

Figure 4. Proposed RSA Cryptography Flow Diagram.

Encryption: RSA encryption with the public key encrypts the encoded message, transforming it into ciphertext. This ensures that only the intended recipient with the corresponding private key can decrypt and access the original message. With the encoded message EM prepared using OAEP, RSA encryption is applied. RSA encryption uses the public key (n, e) to encrypt EM.

$$C = EM^e \bmod (n) \quad (17)$$

Here, C is the ciphertext, EM is the encoded message, e is the public exponent, n is the RSA modulus.

Decryption: RSA decryption uses the private key to retrieve the encoded message from the ciphertext. This step is necessary to recover the original encoded message, which will then be decoded using OAEP. The ciphertext C is decrypted using the private key (n, d). The receiver uses their private exponent d to compute the encoded message EM.

$$EM = C^d \bmod (n) \quad (18)$$

Here, EM is the decoded message, C is the ciphertext, d is the private exponent, n is the RSA modulus.

3.2.4 OAEP Decoding

Figure 5 presents the OAEP decoding flow diagram, beginning with the RSA-decrypted encoded message as input. The OAEP decoding process removes the padding and restores the original user message. This method ensures that any additional security layers applied during encryption are reversed, yielding the extracted user message as output.

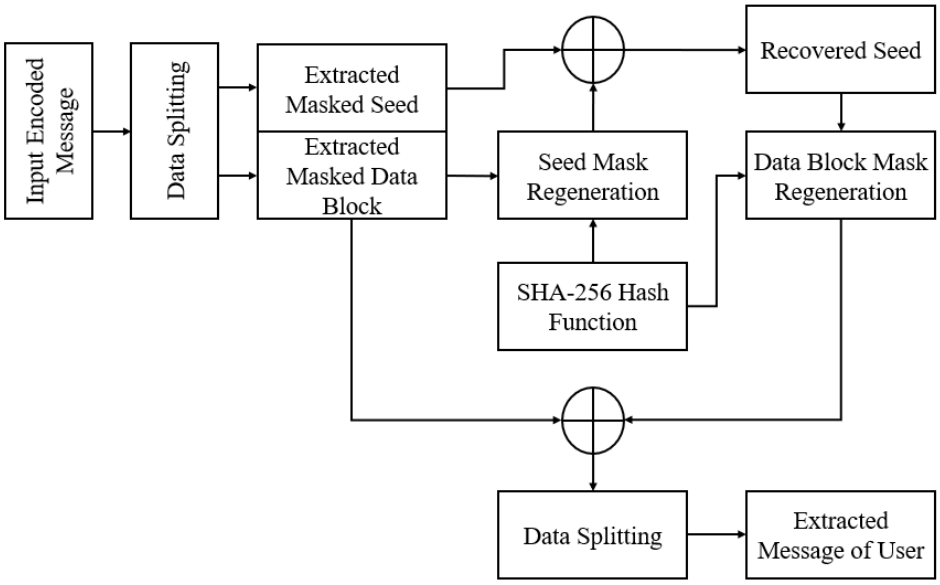

Figure 5. Proposed OAEP Decoding Flow Diagram.

Extract the Masked Seed and Masked Data Block: This step separates the components of the encoded message to allow for individual processing of the seed and data block. Split the encoded message EM back into the masked seed $maskedSeed$ and the masked data block $maskedDB$.

$$\{maskedSeed, maskedDB\} = \text{split(EM)} \tag{19}$$

Generate the Seed Mask: Regenerating the seed mask ensures that the original randomness can be recovered accurately, which is crucial for decoding the data block. Use MGF1 to regenerate the seed mask from the masked data block $maskedDB$.

$$seedMask = MGF1(maskedDB, h) \tag{20}$$

Recover the Seed: Recovering the seed restores the original random value used in the encoding process, which is necessary for generating the data block mask. XOR the masked seed with the seed mask to recover the original seed r.

$$r = maskedSeed \oplus seedMask \tag{21}$$

Generate the Data Block Mask: Regenerating the data block mask ensures that the original data block can be accurately recovered by reversing the masking process. Use MGF1 again to regenerate the data block mask using the recovered seed r.

$$dataBlockMask = MGF1(r, k - h - 1) \tag{22}$$

Recover the Data Block: Recovering the data block allows for the extraction of the original message and padding. XOR the masked data block with the data block mask to recover the original data block DB.

$$DB = maskedDB \oplus dataBlockMask \tag{23}$$

Extract the Message: Extracting the message ensures that the original plaintext is retrieved correctly after decoding and verifying the padding. Finally, split the data block DB to recover the original plaintext message m.

$$\{PS, Hash(label), m\} = \text{split}(DB) \tag{24}$$

The receiver verifies the integrity of the data by checking the padding string PS and the hash value. If the checks are successful, the plaintext message m is recovered.

3.2 LSB-IWT

The LSB-IWT data hiding, and extraction procedure involves several key steps to ensure secure and effective embedding and retrieval of data. The use of IWT enhances the robustness of the data hiding process by minimizing distortion and preserving the quality of the image. The combination of wavelet-based transformation and LSB modification ensures that the hidden data is well-protected and resilient to various forms of data tampering. Figure 6 illustrates the proposed LSB-IWT operational flow for data hiding and extraction. In the data hiding phase, the original GI cover image and the NaCl-OAEP encrypted message serve as inputs. The LSB-IWT technique is then applied to hide the encrypted message within the cover image, producing a secured stego-image as the output. In the data extraction phase, the process begins with the stego-image as input. The LSB-IWT method is used to extract the hidden data, resulting in the output of the extracted NaCl-OAEP encrypted message. This flow ensures both secure embedding and retrieval of sensitive information in medical images.

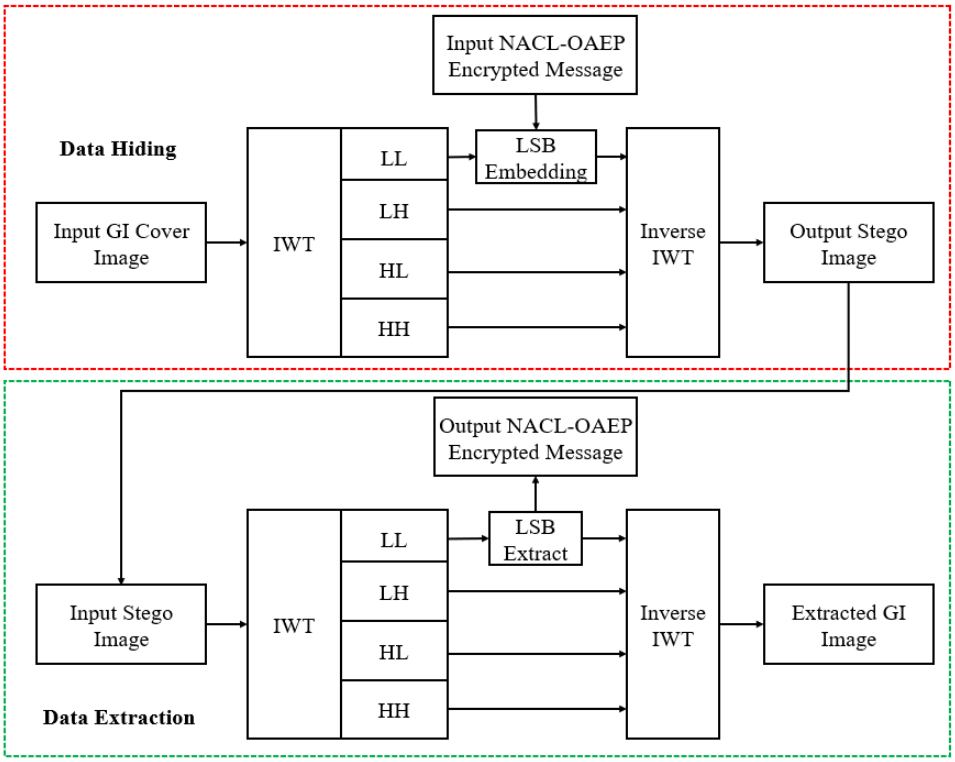

Figure 6. Proposed LSB-IWT Operational Flow.

3.3.1 Data Hiding

Step 1: Input Data: The original GI image and the data to be hidden must be prepared for embedding. The image will be used as the carrier, and the data will be embedded into it. Choose the cover image I with dimensions $M \times N$. Convert the NACL-OAEP encrypted data D to be hidden into a binary format. Ensure that the length of DDD fits within the image's capacity for embedding.

Step 2: Perform IWT: The IWT is applied to transform the image into the wavelet domain, where data embedding can be performed with minimal impact on image quality. Apply a multi-level wavelet transform to the image I. The transform decomposes the image into approximation and detail coefficients.

$$W(L) = \{LL_L, LH_L, HL_L, HH_L\} \tag{25}$$

Here, L denotes the level of decomposition, and $\{LL_L, LH_L, HL_L, HH_L\}$ represent the approximation, horizontal, vertical, and diagonal details, respectively. Repeat the wavelet decomposition for multiple levels to produce more robust components.

Step 3: Data Embedding: The NACL-OAPE encrypted data must be embedded into the image coefficients with minimal distortion. This involves modifying the least significant bits of the coefficients. Convert the binary data D into a format suitable for embedding. Typically, the data is organized as a sequence of bits. Embed the binary data D into the LSB positions of the wavelet coefficients. For example, if C_{ij} is a coefficient of LL_L and d_i is the bit to be embedded:

$$\mathrm{C}'_{\mathrm{ij}} = \text{embedded}\left(C_{ij}, (d_i \; in \; LSB)\right) \tag{26}$$

Here, $\mathrm{C}'_{\mathrm{ij}}$ is the modified LL coefficient.

Step 4: Generate the Stego-Image: The stego-image, which contains the hidden data, must be generated and saved. This image will be used for transmission or storage. Combine the modified LL coefficients into a single matrix. Then, perform the inverse IWT on the combined $\mathrm{C}'_{\mathrm{ij}}$ coefficients to reconstruct the stego-image I' using LH, HL, HH coefficients.

$$I' = \mathrm{IWT}^{-1}(C'_{ij}, LH_L, HL_L, HH_L) \tag{27}$$

3.3.2 Data Extraction

Step 1: Input Stego-Image: The process involves retrieving and decoding the embedded data. Obtain the stego-image I' from which data was extracted.

Step 2: Perform IWT: Apply the IWT to the stego-image to retrieve the coefficients where data was embedded. Decompose the watermarked image I' into its wavelet coefficients.

$$W_{LL}{}' = \mathrm{IWT}(I') \tag{28}$$

Here, $W_{LL}{}'$ represents extracted LL coefficients.

Step 3: Data Extraction: Retrieve the bits from the LSB positions of the wavelet coefficients. For each coefficient W_{LL}:

$$\mathrm{C}'_{\mathrm{ij_{new}}} = \text{extract}\left(W_{LL}{}', \; \mathrm{LSB_{length}}\right) \tag{29}$$

$$d_i = LSB(C'_{ij_{new}}) \tag{30}$$

Here, $\mathrm{LSB_{length}}$ is the length of LSB used same in data hiding process, $C'_{ij_{new}}$ represents extracted data coefficients, and d_i represents resultant data coefficients. Combine the extracted bits (d_i) to reconstruct the original binary data D.

4. RESULTS AND DISCUSSION

This section evaluates different methods and metrics by applying them to the same dataset to ensure a fair comparison. It measures and compares performance using metrics providing insights into each method's effectiveness. By maintaining consistent data, the analysis highlights strengths and weaknesses across various approaches.

4.1 Dataset

The Kvasir dataset, sourced from the Vestre Viken Health Trust (VV) in Norway, serves as the foundation for cover images in the proposed system. This dataset, meticulously annotated by experienced medical professionals from the Cancer Registry of Norway and the VV, includes high-resolution endoscopic images that depict various anatomical landmarks and pathological conditions in the GI system. These images, ranging from 720x576 to 1920x1072 pixels, offer valuable visual data for training and validating algorithms in the proposed system, ensuring comprehensive and reliable performance in GI image analysis. Figure 7 shows the sample images from dataset with various, which acts as cover images in proposed TLE-GIS process. With images categorized into distinct classes such as esophagitis, polyps, and ulcerative colitis, as well as procedural images showing lesion removal, the dataset's diverse and detailed coverage supports the development of robust and accurate image-based applications.

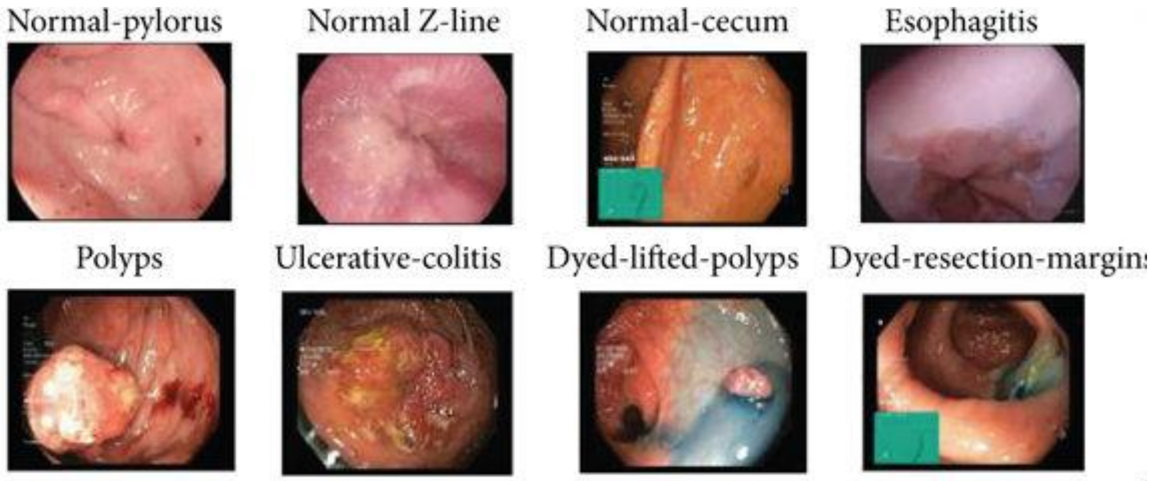

Figure 7. GI-Image Dataset.

4.2 Subjective Analysis

Figure 8 illustrates the outcome of the proposed NACL-OAEP encryption process. It displays a patient's details, including fields such as Name, ID, Age, Message, Blood type, Mail, and Body Mass Index (BMI), which are combined into a single message string separated by hash symbols (#). This combined message is subject to encryption using both NaCL and OAEP methods. It showcases the encrypted results for the even, and odd parts of the message individually, followed by their respective encrypted outcomes. Additionally, it presents the final combined encrypted message, which integrates both the NaCL-encrypted even part and the OAEP-encrypted odd part, demonstrating the layered encryption approach. Notably, the patient details are customizable and can be updated or expanded by the user within the application.

```
Combined Message : Vamshi#Appolo0505#30#gastric pain, and issues#B+#vamshi.namani@gmail.com#23
Even Message : Vamshi#Appolo0505#30#gastric pain, an
ODD Message : d issues#B+#vamshi.namani@gmail.com#23
NACL Even Encrypted Message : b"\x05w\x13\xa3\xa5\xd0B\x99!yKh\xf0\xc8\x1f\x00&Z\xec\xdb\\cc?\xeb\xd7\xb7v'C\xdd\xb0\xdf\xdf\x94q\xf0\x9b\
x9cC5\x10\xe3\x7f\x13!D\x83&\xe0J\x19\xc5d*\xf3\xf8\xb7+B\xf1\xf8{\xc5\xc3\xe8\x8bq\x86r\r]\x007\xe6\xa1`"

OAEP ODD  Encrypted Message : b"w\x05\xefy\xf4\xa09\x0c\xf5&\xe5s\x84J\x19g-\xfd\x0e\x85\xbaYT\xb3\xc6\xba\xfd\xc3\xa7QP\x0fK'4sLU\x1a \x
a4\xcb\xb7\xadU\x9f<\n\xb6\x7f\x18\x9f\xaf\x15\xbb\xe8\xa2\xe2\x1c\x0c8\x0fT\xe3\x80\xe0\x00\xa0R\xf4\xe7\x01r\xba\xa2\xd1N\x1e\xbf\x88jqZ ss\x
ae:\xad\xde\x9a\xf2\xb5\xb5\xcb\xa8\xac\xd6w\xb8Jsa|\x97J\xe5]\ns\x87\xf1\xa2\xdeok\x9aY\x92J\xcb\x8fqaWQh\xa0t\xce[v\x87\x18\xe7\xca\x12{\x1
4\xec\x01\xfc\xdd\xd2+E\xbeT\x84\x01\xdc=\xaf\x0e\xff /}\r\xac\xbeqf\xa6\xf6\xfc\xb5(\xadQ\xba\x02\x0b\x8f\xc8\xd5\x043j5\x1c\x05E\xa1\xdf\xe8m\
xf0hx\r9\x0cb\xe7\xbbr\xcf\x071\x8cy/\x9c2\xda\x93\x9a\x1a\xfd}\x01m2\xa6\x83\x9e\xa1\x9b^\x8d\xae\x1e\x14\xa4\x92%nU/)\xdd\xa9(L.{\x89\xf2\x
fc\xb2\x8d\x1f\xd2\xd4p\x82\xf6!}\xe9\xeas\x9b\xdaP"

Combined  Encrypted Message : b"\x05w\x13\xa3\xa5\xd0B\x99!yKh\xf0\xc8\x1f\x00&Z\xec\xdb\\cc?\xeb\xd7\xb7v'C\xdd\xb0\xdf\xdf\x94q\xf0\x9b\x
9cC5\x10\xe3\x7f\x13!D\x83&\xe0J\x19\xc5d*\xf3\xf8\xb7+B\xf1\xf8{\xc5\xc3\xe8\x8bq\x86r\r]\x007\xe6\xa1`w\x05\xefy\xf4\xa09\x0c\xf5&\xe5s\x8
```

Figure 8. Proposed NACL-OAEP encrypted outcome.

Figure 9 depicts the results of decrypting the combined encrypted message from a stego-GI image. It illustrates the extraction of the encrypted message from the stego-image and its subsequent decryption process. It shows the decrypted message, revealing the original patient details that were hidden within the image. Furthermore, it includes the separated data from the decrypted message, which allows for verification and utilization of the extracted patient information, highlighting the effectiveness of the data-hiding and retrieval process in the proposed system.

```
x0fT\xe3\x80\xe0\x00\xa0R\xf4\xe7\x01r\xba\xa2\xd1N\x1e\xbf\x88jqZ ss\xae:\xad\xde\x9a\xf2\xb5\xb5\xcb\xa8\xac\xd6w\xb8Jsa|\x97J\xe5]\ns\x87\xf
1\xa2\xdeok\x9aY\x92J\xcb\x8fqaWQh\xa0t\xce[v\x87\x18\xe7\xca\x12{\x14\xec\x01\xfc\xdd\xd2+E\xbeT\x84\x01\xdc=\xaf\x0e\xff /}\r\xac\xbeqf\xa6\
xf6\xfc\xb5(\xadQ\xba\x02\x0b\x8f\xc8\xd5\x043j5\x1c\x05E\xa1\xdf\xe8m\xf0hx\r9\x0cb\xe7\xbbr\xcf\x071\x8cy/\x9c2\xda\x93\x9a\x1a\xfd}\x01m2\x
a6\x83\x9e\xa1\x9b^\x8d\xae\x1e\x14\xa4\x92%nU/)\xdd\xa9(L.{\x89\xf2\xfc\xb2\x8d\x1f\xd2\xd4p\x82\xf6!}\xe9\xeas\x9b\xdaP"

Extracted Decrypted Message : Vamshi#Appolo0505#30#gastric pain, and issues#B+#vamshi.namani@gmail.com#23

Extracted Name: Vamshi
Extracted ID  : Appolo0505
Extracted Age : 30
Extracted Message   : gastric pain, and issues
Extracted Bloog Group: B+
Extracted Mail ID   : vamshi.namani@gmail.com
Extracted BMI       : 23
```

Figure 9. Proposed NACL-OAEP decrypted outcome.

Figure 10 compares the steganography outcomes of different methods applied to GI cover image. Figure 10 (a) shows the original image used for embedding secret data. Figure 10 (b) illustrates the RSA-AES-DWT [11] method, where

the stego-image suffers from low contrast issues, making the embedded data less discernible. Figure 10 (c) displays the LSB-ANC-SHA-256 [12] method outcome, which results in color degradation, impacting the visual quality of the stego-image. Figure 10 (d) represents the CrypticCare method outcome, where the stego-image exhibits both color degradation and high sharpness, which can distort the image appearance. Finally, Figure 10 (e) demonstrates the proposed TLE-GIS method, which achieves a balance between effective data embedding and minimal visual distortion, showcasing improved performance in preserving image quality while ensuring secure data hiding.

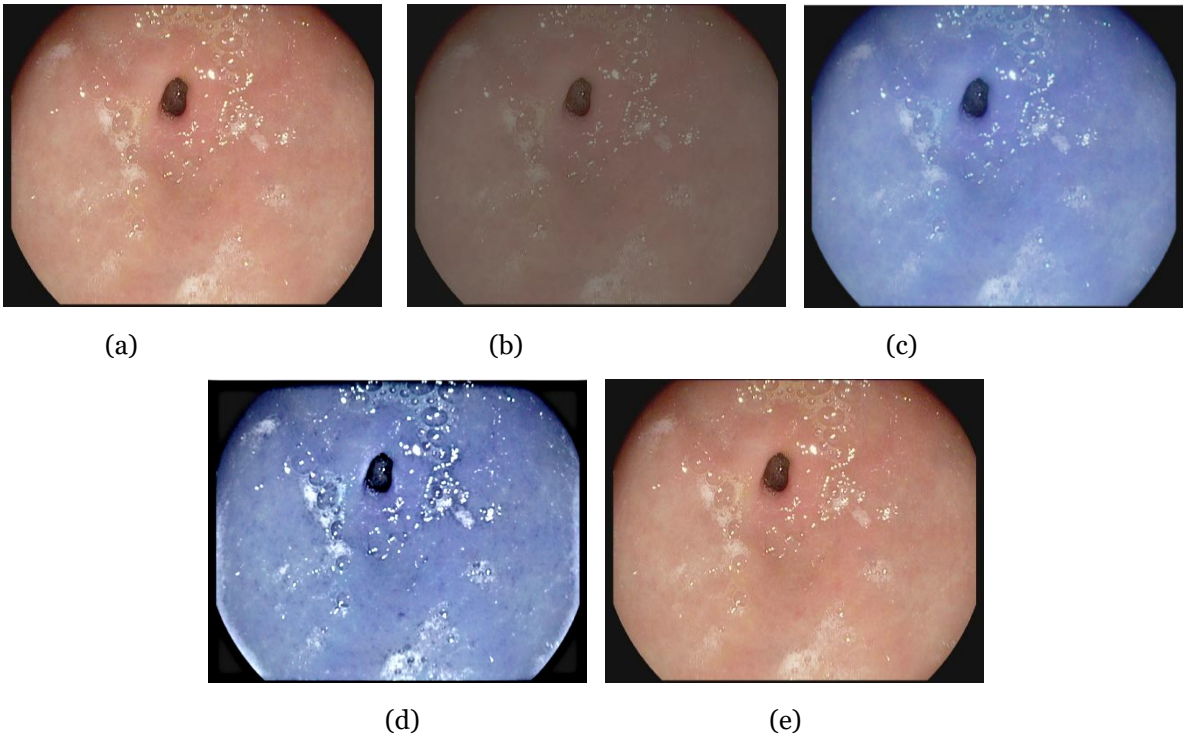

Figure 10. Steganography outcomes of various methods. (a) GI Cover image. (b) RSA-AES-DWT [11]. (c) LSB-ANC-SHA-256 [12]. (d) CrypticCare [13]. (e) Proposed TLE-GIS.

Table 1 compares the performance of different steganography methods in terms of Peak Signal-to-Noise Ratio (PSNR) and Normalized Cross-Correlation (NCC). The PSNR is a measure of the quality of the stego image compared to the original image, reflecting how much distortion is introduced by the steganography process. Higher PSNR values indicate better image quality with less distortion. Here, the proposed TLE-GIS method achieves the highest PSNR of 84.6734, significantly outperforming other methods, which have PSNR values ranging from 40.6206 to 67.55. The NCC measures the similarity between the stego-image and the original image, with higher values indicating better preservation of the image's integrity. The proposed TLE-GIS also has the highest NCC value of 0.9999999, showing excellent similarity to the original image compared to the other methods, which range from 0.9959 to 0.9986.

Table 1: Image quality metrics comparison of various steganography methods.

Method	PSNR	NCC
AES-RSA-DWT [11]	40.6206	0.9986
CrypticCare [13]	40.6282	0.9986
LSB-ANC-SHA-256 [12]	67.55	0.9959
Proposed TLE-GIS	84.6734	0.9999999

Table 2 presents Structural Similarity Index Measure (SSIM) and Feature Similarity Index (FSIM) metrics to evaluate the similarity between the stego-image and the original image. The SSIM measures the perceived quality of the image by comparing luminance, contrast, and structure. Higher SSIM values signify better image quality. The FSIM evaluates the similarity based on feature extraction, focusing on the naturalness and visual quality of the image. Both

metrics are critical for assessing how well the steganography method preserves the essential visual characteristics of the image. The proposed TLE-GIS method shows the highest SSIM (0.9999999) and FSIM (0.9999985), indicating superior performance in maintaining image similarity compared to other methods, which have lower scores (SSIM from 0.9911 to 0.9887 and FSIM from 0.9877 to 0.9859).

Table 2: Image similarity metrics comparison of various steganography methods.

Method	SSIM	FSIM
AES-RSA-DWT [11]	0.9911	0.9877
CrypticCare [13]	0.9911	0.9877
LSB-ANC-SHA-256 [12]	0.9887	0.9859
Proposed TLE-GIS	0.9999999	0.9999985

Table 3 provides Mean Squared Error (MSE), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE) metrics to evaluate the accuracy and error characteristics of various steganography methods. The MSE measures the average squared difference between the original and stego-image pixel values, with lower values indicating less distortion. The MAE provides the average magnitude of errors between the original and stego-image pixel values, focusing on absolute differences. The RMSE represents the square root of the average squared differences and is sensitive to large errors. The proposed TLE-GIS method excels in all these metrics, demonstrating the lowest error values, including the smallest MSE (3.4093e-09), MAE (1.2079e-06), and RMSE (5.8389e-05), compared to other methods.

Table 3: Image error metrics comparison of various steganography methods.

Method	MSE	MAE	RMSE
AES-RSA-DWT [11]	8.6685e-05	0.0049	0.0093
CrypticCare [13]	8.6532e-05	0.0049	0.0093
LSB-ANC-SHA-256 [12]	1.14e-08	1.268e-05	1.068e-04
Proposed TLE-GIS	3.4093e-09	1.2079e-06	5.8389e-05

5. CONCLUSION

In conclusion, the proposed TLE-GIS methodology offers a robust solution for securing sensitive medical data by combining advanced encryption techniques with sophisticated data hiding methods. By utilizing NaCL and OAEP for encryption and integrating LSB-IWT for data embedding, the approach ensures high-level security while preserving medical image quality. This multi-layered strategy effectively addresses the limitations of existing protocols, providing a comprehensive framework for protecting GI disease data in IoMT systems. Future research could explore the application of this security framework in the classification of GI diseases using advanced machine learning algorithms. Enhancements in encryption and data hiding methods could further optimize the balance between security and classification accuracy, facilitating more effective and secure disease diagnosis and management.

REFERENCES

[1] Zhang, K., Wang, H., Cheng, Y., Liu, H., Gong, Q., Zeng, Q., ... & Chen, D. (2024). Early gastric cancer detection and lesion segmentation based on deep learning and gastroscopic images. *Scientific Reports*, *14*(1), 7847.

[2] Hossain, Tanzim, FM Javed Mehedi Shamrat, Xujuan Zhou, Imran Mahmud, Md Sakib Ali Mazumder, Sharmin Sharmin, and Raj Gururajan. "Development of a multi-fusion convolutional neural network (MF-CNN) for enhanced gastrointestinal disease diagnosis in endoscopy image analysis." *PeerJ Computer Science* 10 (2024): e1950.

[3] P. Ping, P. Wei, D. Fu, B. Guo, O. T. Bloh and F. Xu, "IMIH: Imperceptible Medical Image Hiding for Secure Healthcare," in *IEEE Transactions on Dependable and Secure Computing*, doi: 10.1109/TDSC.2024.3355165

[4] Esmaeelzadeh Rostam, Habib, Homayun Motameni, and Rasul Enayatifar. "Privacy-preserving in the smart healthcare system using steganography and chaotic functions based on DNA." *Security and Privacy* 7, no. 3 (2024): e363.
[5] Alenizi, Abdullah, Mohammad Sajid Mohammadi, Ahmad A. Al-Hajji, and Arshiya Sajid Ansari. "A Review of Image Steganography Based on Multiple Hashing Algorithm." *Computers, Materials and Continua* 80, no. 2 (2024): 2463-2494.
[6] Singh, Dilbag, Sharanpreet Kaur, Mandeep Kaur, Surender Singh, Manjit Kaur, and Heung-No Lee. "A systematic literature review on chaotic maps-based image security techniques." *Computer Science Review* 54 (2024): 100659.
[7] K. N. Singh, N. Baranwal, A. K. Singh, A. K. Agrawal and H. Zhou, "Using Multimodal Biometrics, Data Hiding and Encryption for Secure Healthcare Imaging System," in *IEEE Transactions on Consumer Electronics*, doi: 10.1109/TCE.2024.3438356.
[8] Afzal, Ifrah, Shabir A. Parah, Nasir N. Hurrah, and O. Y. Song. "Secure patient data transmission on resource constrained platform." *Multimedia Tools and Applications* (2024): 1-26.
[9] Charoghchi, Sara, and Samaneh Mashhadi. "A secure secret image sharing with steganography and authentication by Hamming code (15, 11) for compressed images." *Multimedia Tools and Applications* 83, no. 11 (2024): 31933-31955.
[10] Balas, Michael, Chris Rudnisky, and Edsel B. Ing. "Hidden in Plain Sight: AI-Driven Steganography and Watermarking for Secure Transmission of Ophthalmic Data." *AJO International* (2024): 100043.
[11] M. Elhoseny, G. Ramírez-González, O. M. Abu-Elnasr, S. A. Shawkat, N. Arunkumar and A. Farouk, "Secure Medical Data Transmission Model for IoT-Based Healthcare Systems," in *IEEE Access*, vol. 6, pp. 20596-20608, 2018, doi: 10.1109/ACCESS.2018.2817615.
[12] Hameed, Mohamed Abdel, M. Hassaballah, Riem Abdelazim, and Aditya Kumar Sahu. "A novel medical steganography technique based on Adversarial Neural Cryptography and digital signature using least significant bit replacement." *International Journal of Cognitive Computing in Engineering* (2024).
[13] Ramyashree, P. S. Venugopala, S. Raghavendra and B. Ashwini, "CrypticCare: A Strategic Approach to Telemedicine Security Using LSB and DCT Steganography for Enhancing the Patient Data Protection," in *IEEE Access*, vol. 12, pp. 101166-101183, 2024, doi: 10.1109/ACCESS.2024.3430546.
[14] Latif, Ghazanfar, Jaafar Alghazo, Nazeeruddin Mohammad, Sherif E. Abdelhamid, Ghassen Ben Brahim, and Kashif Amjad. "A Novel Fragmented Approach for Securing Medical Health Records in Multimodal Medical Images." *Applied Sciences* 14, no. 14 (2024): 6293.
[15] Zhang, Lina, Xianhua Song, Ahmed A. Abd El-Latif, Yanfeng Zhao, and Bassem Abd-El-Atty. "Reversibly selective encryption for medical images based on coupled chaotic maps and steganography." *Complex & Intelligent Systems* 10, no. 2 (2024): 2187-2213.
[16] Ch, Rupa, Venkayya Yadlapalli, Suhana Sulthana Sk, G. Thippa Reddy, and Sandeep Kautish. "Robust steganographic framework for securing sensitive healthcare data of telemedicine using convolutional neural network." *CAAI Transactions on Intelligence Technology* (2024).
[17] Mohammed, Ajmal, and P. Samundiswary. "SecMISS: Secured Medical Image Secret Sharing mechanism for smart health applications." *The Visual Computer* 40, no. 6 (2024): 4251-4271.
[18] Tiwari, Anurag, Divyanshu Awasthi, and Vinay Kumar Srivastava. "Image security enhancement to medical images by RDWT-DCT-Schur decomposition-based watermarking and its authentication using BRISK features." *Multimedia Tools and Applications* 83, no. 22 (2024): 61883-61912.
[19] Nadhan, Archana S., and I. Jeena Jacob. "Enhancing healthcare security in the digital era: Safeguarding medical images with lightweight cryptographic techniques in IoT healthcare applications." *Biomedical Signal Processing and Control* 88 (2024): 105511.
[20] S. Sankaranarayanan *et al.*, "Enhancing Healthcare Imaging Security: Color Secret Sharing Protocol for the Secure Transmission of Medical Images," in *IEEE Access*, vol. 12, pp. 100200-100216, 2024
[21] Priya, S., S. P. Abirami, B. Arunkumar, and B. Mishachandar. "Super-resolution deep neural network (SRDNN) based multi-image steganography for highly secured lossless image transmission." *Scientific Reports* 14, no. 1 (2024): 6104.
[22] Huo, Lin, Ruipei Chen, Jie Wei, and Lang Huang. "A high-capacity and high-security image steganography network based on chaotic mapping and generative adversarial networks." *Applied Sciences* 14, no. 3 (2024): 1225.

[23] Rezaei, Samira, and Amir Javadpour. "Bio-Inspired algorithms for secure image steganography: enhancing data security and quality in data transmission." *Multimedia Tools and Applications* (2024): 1-34.

[24] Shi, Hui, Ziyi Zhou, Jianhao Qin, Hao Sun, and Yonggong Ren. "A separable privacy-preserving technique based on reversible medical data hiding in plaintext encrypted images using neural network." *Multimedia Tools and Applications* (2024): 1-26.

[25] El-Shafai, Walid, Fatma Khallaf, El-Sayed M. El-Rabaie, and Fathi E. Abd El-Samie. "Proposed 3D chaos-based medical image cryptosystem for secure cloud-IoMT eHealth communication services." *Journal of Ambient Intelligence and Humanized Computing* 15, no. 1 (2024): 1-28.