ABSTRACT

The Internet of Things (IoT) is a network of physical objects that are interconnected through sensors and software to exchange data with other devices and systems via the Internet. The Internet of Medical Things (IoMT) technologies is one of the IoT branches, where IoMT devices transmit a vast amount of medical data in real time. Healthcare data is highly susceptible to cyber attackers. In this project, we have designed a hybrid framework combining cryptographic and steganographic methods.

Cryptography is the technique of transforming plain text or image to ciphertext or cipher image using mathematical techniques. We have implemented Advanced Encryption Standard (AES) 256 as cryptographic technique. It is a secure symmetric encryption algorithm using a single 256-bit key to encrypt and decrypt data. Steganography is the process of concealing information in another medium (audio, video, image, etc.). It hides the transmitted data inside a cover object (carrier object). Through the process, a stego-object is created, which carries the hidden information object but appears identical to the cover object. The encoder that we have implemented is a combination of AES-256 cryptographic and the Least Significant Bit (LSB) steganographic technique. On the encoder side, sensitive medical data is first encrypted with a symmetric key (k) utilizing the AES-256 algorithm. This encrypted data is transformed into a bitstream. The bitstream is embedded into a cover image using the LSB method, resulting in a stego image. The stego image is then transmitted over the Internet. On the decoder side, the steps are performed in the reverse sequence. The LSB-embedded data is extracted first. Then the extracted data is decrypted using the symmetric key (k) of AES-256. For performance evaluation of the use case, we have analyzed the histograms of the original image and the stego image. Identical histograms prove that both images are alike and undetectable to hackers.

The objective of the project is to ensure communication security in IoMT devices through a hybrid cryptographic and steganographic technique in medical data transmission. Securing healthcare data will increase the effectiveness of IoMT devices benefiting the patients and developing a smart healthcare system.

1. INTRODUCTION

The Internet of Things (IoT) is a network of physical objects that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the internet. In, recent years, IoT has increasingly been implemented by many applications affecting our daily life. According to Fabio Duarte, there are over 15 billion connected IoT devices worldwide and the number of IoT devices is expected to double by 2030 [1]. These devices are compact, can communicate in various ways, and have low power consumption which makes them exceptionally efficient in real-life applications.

The Internet of Medical Things (IoMT) is an IoT branch dedicated to the healthcare industry. The IoMT is the collection of medical devices and applications that connect to healthcare IT systems through online computer networks. Medical devices equipped with Wi-Fi allow the machine-to-machine communication that is the basis of IoMT. These IoMT devices link to cloud platforms, on which captured data can be stored and analyzed. IoMT is also known as healthcare IoT. According to market watch, the global IoMT market is poised to value over 187.60 billion by 2028 end [2]. Beside this, the global IoMT market is projected to grow significantly in terms of revenue in the near future, as the industry is moving towards the recovery period from the pandemic.

1.1 Project Overview

IoMT is a network that has being used by wireless sensor networks (WSN) and radio frequency identification (RFID) through wireless network and technology to achieve perception of information, reliable transmission, and intelligent processing. Hence, ensuring secure data transmission by protecting privacy and safety are essential features of IoMT. This security is related to tag information (RFID), wireless communication information security, network transmission of information security, privacy, and information processing security. Therefore, it is important to have a thorough study and research on IoMT device, potential threats, and security requirements. The main purpose of the network security and information protection is to achieve

confidentiality and integrity. Security issues are of great importance in enlarging the scale of network and devices. There are some security risks in both consumers and business in IoMT. So, secure data encryption and decryption techniques can be used to reduce security risks. A suitable encryption decryption algorithm can play a vital role in reducing the security risks.

A hybrid encryption and decryption algorithm has been implemented as a use case in this project. Cryptography and steganography are combinedly used as encryption and decryption technique in this use case. The cryptography converts the plain text into an incomprehensible cipher text whereas steganography conceals the traces of the data. When hackers want to access a system, they will aim for the weakest point, which is not the encryption, but the key. Advance Encryption Standard (AES)-256 is the most secure encryption technique and till now there is no report of its cracking. But the key is the weakest point. The key can be secured by using steganography or secure hash algorithm (SHA) which is irreversible.

The summary of the contributions that we have achieved are as follows-

1. A structured systematic review of IoMT device advantages in the healthcare system.
2. Security requirements that are necessary for IoMT systems as well as different types of techniques to provide secure data collection, transmission, and storage are discussed.
3. We have designed a hybrid encryption-decryption method combining cryptography and steganography and evaluated the performance of the implemented method.

2. BACKGROUND

In the background analysis of this project, we will briefly discuss the context and structure of IoMT devices, communication protocols, IoMT related technologies, security requirements and techniques. We will also discuss IoMT device pros cons, and latest IoMT device trends for 2023.

2.1 Architecture of IoMT

The IoMT is an incorporation of medical devices and applications that can connect to healthcare information technology systems using networking technologies. It can reduce unnecessary hospital visits and the burden on healthcare systems by connecting patients to their physicians and allowing the transfer of medical data over a secure network. Each technology has its own set of guidelines. IoMT devices are mainly divided into four layers. They are the sensor layer, gateway layer, cloud layer, and visualization layer. All layers are illustrated in Figure 1 below.

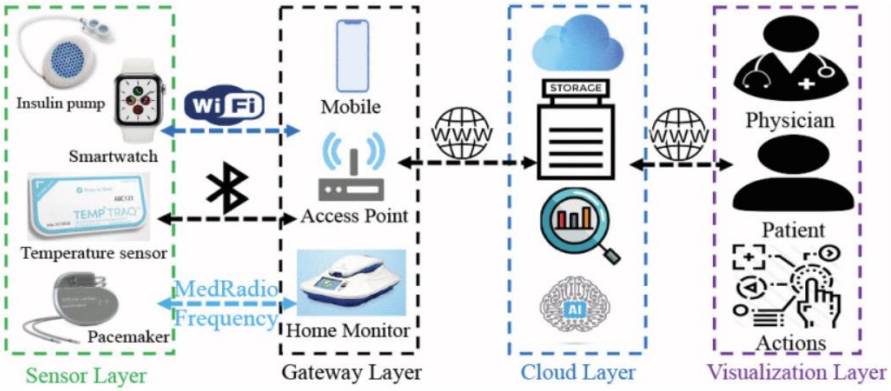

Figure 1 IoMT architecture [3]

Sensor layer: This layer consists of biometric sensors which are placed in the patient's body. They transfer data to the gateway layer through wireless protocols like Wi-Fi and Bluetooth.

Gateway layer: Due to the processing and storage limitations of IoMT sensors, the data is transferred without processing to the second layer, i.e., the gateway layer. In this layer, data can

be stored for a short time and perform some preprocessing. These devices can be patients' smartphones, access points, or home monitoring systems.

Cloud layer: The cloud layer is responsible for getting the data from the gateway for storage, analysis, and secure access. The analysis includes data processing and finding any changes in the patient's health. Then it is presented to the physicians or patients for further action. The access to the sensors can be remotely managed and controlled from this layer.

Visualization layer: In this layer, the data are presented to the physicians and the patients to track their health. This layer also includes the actions recommended by the physician based on the patient's health conditions.

2.2 IoMT Communication Protocols

Communication protocols play a crucial role on the IoMT architecture as they define how devices and systems communicate with each other. In IoMT, communication protocols enable the seamless transfer of medical data between various devices, sensors, gateways, and cloud-based platforms, ensuring secure and reliable data transmission [13-15].

One of the most widely used communication protocols in IoMT is the Bluetooth Low Energy (BLE) protocol, which is often used in wearable medical devices such as fitness trackers and blood glucose monitors. BLE provides low power consumption, reliable data transmission, and low latency, making it ideal for medical devices that require continuous monitoring.

Another important communication protocol used in IoMT is the Zigbee protocol, which is used in home healthcare systems and hospital environments. Zigbee is a low-power, wireless mesh network protocol that allows devices to communicate with each other through a centralized hub or gateway. Zigbee provides reliable and secure communication between devices, ensuring data privacy and confidentiality.

The Hypertext Transfer Protocol (HTTP) is another widely used communication protocol in IoMT, which is used for web-based communication between medical devices and cloud-based servers. HTTP enables secure communication and is used for transferring patient data, images, and medical records securely and efficiently.

The Health Level 7 (HL7) protocol is another communication standard that is widely used in healthcare settings. HL7 enables the exchange of healthcare information between different healthcare systems and devices. It provides a framework for standardizing medical data exchange, ensuring interoperability between different medical systems.

In summary, communication protocols are an essential component of the IoMT architecture, enabling secure and reliable communication between different devices, systems, and platforms. The choice of protocol depends on the application requirements, device type, and data transfer needs, and understanding the various protocols used in IoMT is crucial for developing and deploying effective and efficient IoMT systems.

2.3 IoMT Related Technologies

Wireless Sensor Networks (WSN) and Radio-Frequency Identification (RFID) systems are crucial technologies on the IoMT. The integration of WSN and RFID with smart objects has provided novel communication capabilities in recent years. Given their advantages and widespread use in IoMT, we provide a brief overview of these technologies in this section.

Wireless Sensor Networks (WSN): WSNs in IoMT can be used for various purposes, such as monitoring vital signs, tracking patients' movements, detecting falls, and alerting caregivers in case of an emergency. For example, a WSN can be used to monitor a patient's blood pressure, heart rate, and body temperature continuously and wirelessly transmit the data to a healthcare provider for analysis. In case of abnormal readings, the provider can be alerted, and necessary action can be taken promptly. A WSN usually consists of a microprocessor, memory unit, communication interface, and a battery shown in Figure 2. These sensors can be attached to objects/persons according to their specific needs [14,16,17]. WSNs in IoMT can also be used for remote patient monitoring, telemedicine, and clinical research. With the help of WSNs, healthcare providers can monitor patients' health status in real-time, regardless of their location. This has significant implications for elderly patients, those with chronic conditions, and those who live in remote areas with limited access to medical facilities. In conclusion, WSNs in IoMT have the potential to transform healthcare by enabling remote monitoring, early detection of health problems, and

personalized care. However, careful consideration must be given to the design and implementation of WSNs to address the challenges associated with their use in healthcare.

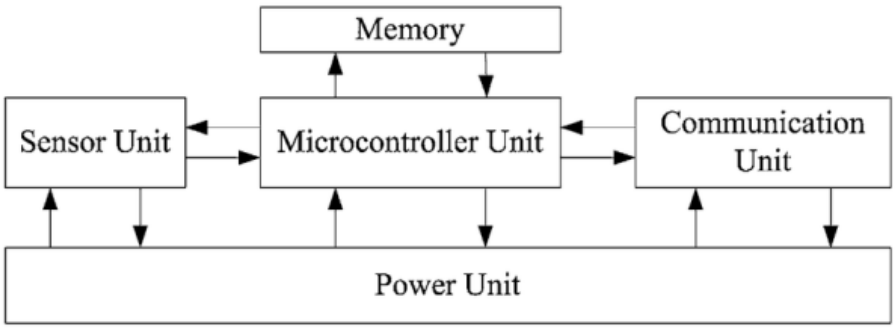

Figure 2 WSN architecture [4]

RFID: RFID technology has found significant applications on the IoMT due to its ability to track and monitor medical equipment, supplies, and patients' movements. RFID technology uses radio waves to communicate between a reader and a tag, which contains information about the item or person being tracked. In IoMT, RFID technology can be used to track medical equipment, such as surgical tools, to ensure they are properly sterilized and accounted for. It can also be used to track medical supplies, such as medication, to prevent waste and ensure proper inventory management. RFID technology can also be used to track patients' movements within a healthcare facility, enabling healthcare providers to monitor patients' activities, ensure their safety, and optimize care delivery. For example, RFID tags can be placed on patients' wristbands, which can be used to monitor their location and activities in real-time [18,19]. However, the deployment of RFID in IoMT poses challenges, such as data privacy and security concerns. Healthcare providers must ensure that the data collected by RFID tags are secure and comply with data protection regulations. Overall, the use of RFID technology in IoMT has the potential to improve patient safety, optimize care delivery, and streamline medical inventory management.

2.4 IoMT Security Requirements

The patient’s medical data is very sensitive. A set of requirements that can ensure IoMT systems’ security at all layers is needed. The set has been derived from CIANA (Confidentiality,

Integrity, Availability, Non-Repudiation, and Authentication) considerations and consists of the following security requirements [20], [21].

1. **Authentication and authorization**: Ensuring that only authorized users and devices can access and modify medical data.
2. **Data privacy and confidentiality**: Protecting sensitive medical data from unauthorized access or disclosure.
3. **Integrity and non-repudiation**: Ensuring that medical data is not tampered with and that there is a clear audit trail of all data transactions.
4. **Availability and reliability**: Ensuring that medical data and devices are always available and functional.
5. **Compliance**: Ensuring that IoMT systems comply with relevant regulatory requirements, such as HIPAA and GDPR.
6. **Risk management**: Identifying and mitigating potential security risks and vulnerabilities.
7. **Physical security**: Ensuring that medical devices and data are physically secure and protected from theft or damage.
8. **Incident response**: Developing and implementing a plan to detect, respond to, and recover from security incidents or breaches.

2.5 Attacks on IoMT

The IoMT applications rely on a variety of technologies with their own set of security vulnerabilities. Due to the lack of fundamental security procedures, IoMT devices are susceptible to various types of cyber-attacks, which not only compromise patients' privacy but also cause financial and reputational damage [22]. For example, the healthcare industry has lost more than $160 million since 2016 due to cyber-attacks, including ransomware attacks, which have increased by 94% between 2021 and 2022 [23]. In addition, attacks on brain implants have resulted in fatalities. Table 1 shows some of the potential IoMT attacks and their influence on the system’s security requirements.

Table 1 Description of attacks and their effect on IoMT security requirements [4]

Attack	Brief Description	Effects
Side-channel attack	The information is obtained from the side channels of the encryption device.	Confidentiality, Integrity
Tampering devices	The IoMT device is physically accessed to modify the data (modification in a device using RFID or communication link).	Confidentiality, Integrity
Tag cloning	An attacker might exploit data obtained through a successful side-channel attack or replicate data from a previously used tag. The cloned tag, for example, might be used to gain access to an unlawful facility or data, such as medical data (Using simple technologies, attackers may clone RFIDs).	Confidentiality, Authorization, Integrity
Sensor tracking	This form of attack invades patients' privacy. Attackers might obtain access to patients' whereabouts or fake GPS data by using unsecured equipment. Other sensors, such as those used in fall detection, wheelchair management, and remote monitoring systems, can also be utilized to divulge sensitive data about patients.	Confidentiality, Authorization, Integrity, Privacy
Eavesdropping	An attacker intercepts and tracks the necessary hardware and communication to capture data. Data obtained in this manner (unlawfully) can be utilized in a variety of ways.	Confidentiality, Non-repudiation, Privacy
Replay	An attacker can use an authentication message that was previously transmitted between two legitimate users. In this situation, an attacker can intercept a signed packet and send it back to target multiple times.	Authorization
Man-in-the-middle	It's a cyber-attack that targets two IoMT devices' communication and gains access to their private data. The attacker can listen in on or monitor the communication between the two devices in this attack. The attacker can alter the intercepted data before they are transmitted to their intended destination.	Confidentiality, Authorization

Rogue access	A fake gateway is placed inside the wireless network range in this attack to give genuine users access and intercept traffic.	
DoS/DDoS	Unlike DoS attacks, which are carried out by a single node, a DDoS attack is carried out by several sources, flooding a specified target with messages or connection requests with the purpose of rendering the service inaccessible to legitimate users.	Availability
Sinkhole	A malicious node attracts traffic in this attack by offering a better connection quality. Once the attack is successful, other attacks (such as eavesdropping or selective forwarding) can be launched, in which the malicious node isolates specific nodes by discarding packets that pass through them.	
Sniffing	Data transferred between two nodes is passively intercepted by sniffing attacks. Since the attacker can observe the data passed between the system's layers.	Confidentiality
Selective Forwarding	A malicious node may simply change, drop, or selectively forward some messages to other nodes in the network. As a result, the information received by the destination is incomplete.	All
Brute Force	The attackers usually use automated tools to create multiple password combinations until they succeed. The dictionary attack is an example of a serious vulnerability for IoMT devices.	Confidentiality, Integrity
SQL injection	An SQL injection attack involves introducing a faulty SQL statement into the application's backend database. A successful SQL injection attack can compromise or change sensitive patient data.	All

Account hijacking	At the network level, many IoT devices communicate in transparent text or with insecure encryption. Intercepting the packet when an end user is authenticating allows an attacker to undertake account hijacking.	Confidentiality, Integrity
Ransomware	Ransomware encrypts important information and demands a large fee to unlock it. In return for money, attackers can encrypt sensitive data such as health information and keep the decryption key.	Integrity, Availability

2.6 IoMT Device Strengths and Weaknesses

IoMT devices can transfer data within seconds via internet. These data can be electronic medical records, physical check results, electronic prescriptions, diagnostic reports, medical reference invitations, personal daily health information, etc. This is the biggest advantage of IoMT device. But every device has its own advantages and disadvantages. Some of the advantages of IoMT devices are-

1. **Portability:** Most of the IoMT devices are small and portable. These devices are cell phone, smart watch, or any other wearable devices. People can travel with those devices and checks heart rate, oxygen saturation, BMI, number of steps taken etc. while travelling.
2. **Many applications:** IoMT devices have many applications like patients' making online appointments and consulting with doctors, paperless medical records, preserving whole medical history in the server etc. But the most import application now a days is data mining. Through data mining, we can obtain deeper cognition of the reason why disease happens to someone. For example, by using association rules, researchers can know the potential relationship among genes of disease.
3. **New job for experts in medical industry:** The development of Internet Medical Things would create new positions. With the Internet of Medical Things, the number of outpatient doctors will decrease, while special data analysis experts will be increasingly needed.

Some of the disadvantages of IoMT devices are-

1. **Battery Life:** The portable facility relies on a battery as a power source. Therefore, the weakness of the battery turns into one of the weaknesses of the Internet of Medical Things. With the development of technology, the battery can work under a secure circuit so that the battery can continuously output a stable current until the voltage drops to the minimum even there is some unstable current.
2. **Monitoring Liability:** Data taken by the IoMT devices are not monitored by any healthcare professional. So, the liability of the accuracy of those data are compromised.
3. **Data Security:** IoMT system transfers a vast amount of medical data. Ensuring the data privacy and security of those data is the biggest challenge and disadvantage of IoMT device.

2.7 IoMT Device Trends 2023

Every year IoMT devices used in healthcare organizations are revolutionizing medical care in unique ways. Improving healthcare outcomes and the evolution of high-speed networking technologies are some of the promising areas of market advancement in IoMT devices. IoMT devices evolve with time.

- **Consumer Health Wearables:** Consumer health wearables are consumer-grade devices for personal fitness, such as activity trackers, bands, wristbands, sports watches, and smart garments. Most of these devices are not regulated by health authorities but may be suggested by experts for specific health applications. Companies operating in this space include Misfit (Fossil Group), Fitbit, and Samsung Medical.
- **In-clinic Segment:** The in-clinic segment includes IoMT devices that are used for administrative or clinical functions. Instead of the care provider physically using a device, the provider can be located remotely while a device is used by qualified staff. Examples include Rijuven's Clinic in a Bag, which is a cloud-based examination platform for clinicians to assess patients at any point of care; ThinkLabs' digital stethoscope; and Tytocare's comprehensive telehealth patient examination device for the heart, lungs, ears, skin, throat, and abdomen, which also can measure temperature.

- **Personal Emergency Response System (PERS):** PERS is a medical alert system that integrates wearable device/relay units and a live medical call center service to increase self-reliance for homebound or limited-mobility seniors. A PERS has three components: a small radio transmitter, a console connected to the telephone, and an emergency response center that monitors calls. The package allows users to quickly communicate and receive emergency medical care [5].
- **Remote Patient Monitoring (RPM) Device:** RPM comprises all home monitoring devices and sensors used for chronic disease management, which involves continuous monitoring of physiological parameters to support long-term care in a patient's home. For continuous observation of discharged patients to accelerate recovery time and prevent re-hospitalization; and medication management acute home monitoring is very important. It can also provide users with medication reminders and dosing information [6].
- **Innovative Devices:** Some of the innovative devices in 2023 are Zoll's wearable defibrillator, which continuously monitors patients at risk of ventricular tachycardia or fibrillation; Stanley Healthcare's hand hygiene compliance system, which incorporates an occupancy sensor and a real-time location system receiver to track the identity of employees using the dispenser and uses analytics to determine whether employees are following hygiene protocol; and Boston Children's Hospital's GPS-based MyWay app, which guides visitors to their destination using the quickest route [7].

2.8 IoMT System Security Techniques

There are several different techniques to secure IoMT systems. These techniques can be divided into three main categories: 1) symmetric; 2) asymmetric; and 3) keyless, as shown in Figure 3. Symmetric and asymmetric techniques rely on cryptographic algorithms, while keyless techniques are noncryptographic [3].

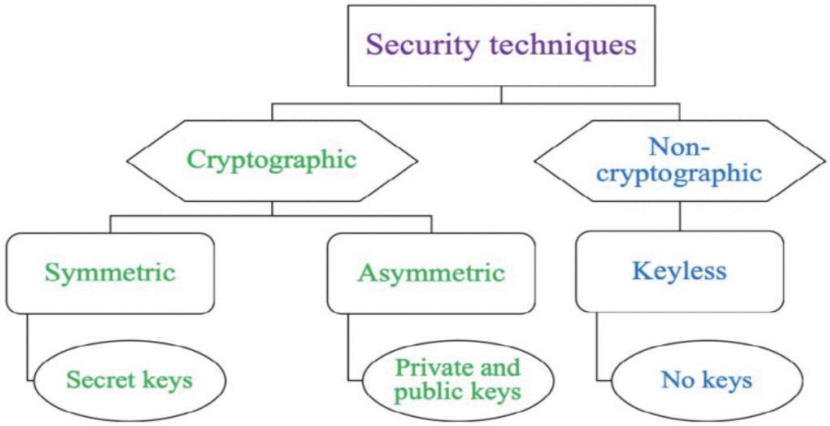

Figure 3 IoMT security techniques [3]

2.8.1 Cryptography

Cryptography is the art of keeping information secret and safe by transforming it into a form that unintended recipients cannot understand. Three basic cryptographic techniques are symmetric key, asymmetric key, and hash function. Symmetric key cryptographic algorithm is based on a single key to encrypt and decrypt data between two nodes trying to communicate. The key is to be generated and distributed prior to using the algorithm. Symmetric key encryption technique is illustrated in figure 4.

Figure 4 Symmetric key Encryption [3]

The public and private key pairs are used in the asymmetric key technique. Key distribution is not required in this algorithm. Public key available to everyone. But the private key is secured

to the receiver only. So, any data encrypted with the public can only be decrypted by the actual receiver. The illustration of asymmetric encryption is presented in figure 5. On the other hand, the hash function algorithm generates a hash value with a fixed length based on the transmitted data.

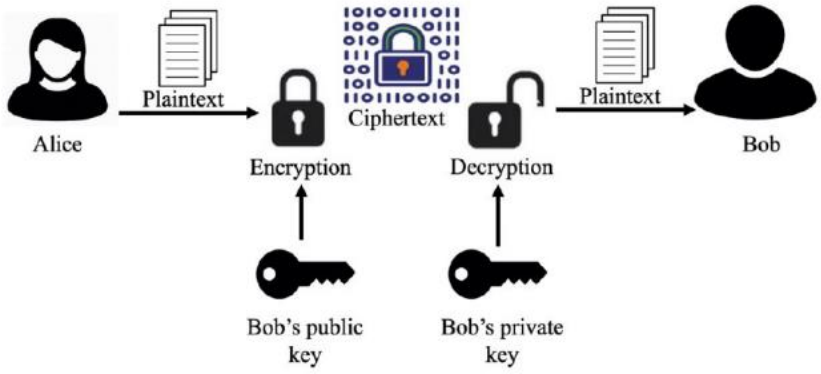

Figure 5 Symmetric key Encryption [3]

2.8.2 Advanced Encryption Standard (AES) 256

The AES 256 Encryption algorithm (also known as the Rijndael algorithm) is a secure symmetric encryption algorithm using a 256-bit key. It transforms plain text into a cipher through multiple rounds including round key addition, byte substitution, row shifting, and column mixing. AES 256 reduces the risk of a data breach in case of a security breach, making the encrypted data unreadable without the proper key [8]. The National Institute of Standards and Technology selected three "flavors" of AES: 128-bit, 192-bit, and 256-bit. Each type uses 128-bit blocks. The difference lies in the length of the key. As the longest, the 256-bit key provides the strongest level of encryption. With a 256-bit key, a hacker would need to try 2^{256} different combinations to ensure the right one is included [8]. AES has several steps through which the data is encrypted. Figure 6 shows all the steps sequentially.

The steps in AES are-

- **Divide Information into Blocks**: The first step of AES 256 encryption is dividing the information into blocks. Assuming AES has a 128- bits block size, it divides the information into 4x4 columns of 16 bytes.

- **Key Expansion**: The next step of AES 256 encryption involves the AES algorithm recreating multiple round keys from the first key using Rijndael's key schedule.
- **Adding the Round Key**: In round key addition, the AES algorithm adds the initial round key to the data that has been subdivided into 4x4 blocks.
- **Bite Substitution**: In this step, each byte of data is substituted with another byte of data.
- **Shifting Rows**: The AES algorithm then proceeds to shift rows of the 4x4 arrays. Bytes on the 2nd row are shifted one space to the left, those on the third are shifted two spaces, and so on.
- **Mixing Columns**: The AES algorithm uses a pre-established matrix to mix the 4x4 columns of the data array.
- **Another Round Key Addition**: The AES algorithm then repeats the second step, adding around key once again, then does this process all over again.

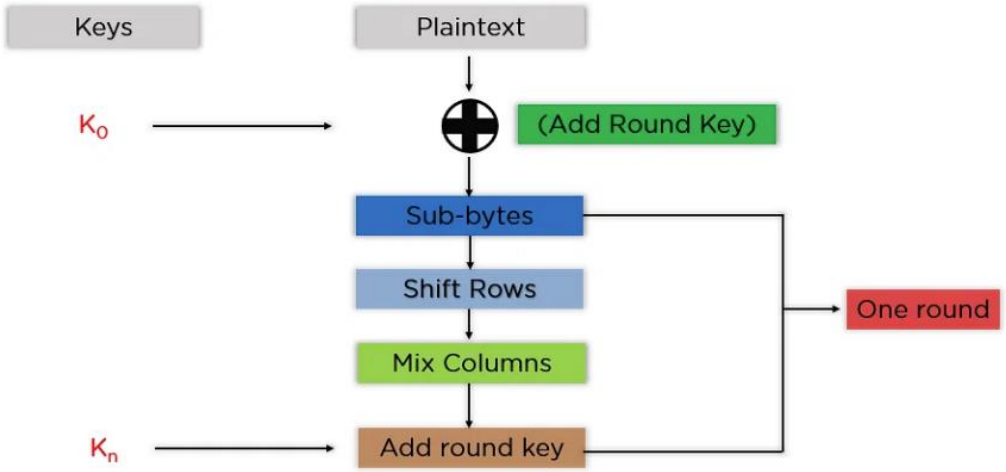

Figure 6 Steps of AES algorithm [9]

The main features of AES are [9]:

- **Key Expansion**: A single key is taken during the first stage, which is later expanded to multiple keys used in individual rounds
- **Byte Data**: The AES encryption algorithm does operations on byte data instead of bit data.

- **Key Length**: The number of rounds to be carried out depends on the length of the key being used to encrypt data. The 128-bit key size has ten rounds, the 192-bit key size has 12 rounds, and the 256-bit key size has 14 rounds.

2.8.3 Steganography

Steganography is the process of concealing information in another medium (audio, video, image, etc.).It is generally known as invisible communication. It hides the transmitted data inside a cover object (carrier object). In this process cover object and stego-object (carrying hidden information object) looks indifferent[10]. In this project we have used image as a cover object. So, we are discussing about image steganography.

Image steganographic techniques can be divided into two domains. Spatial domain and transform domain. There are many versions of spatial steganography. All directly change some bits in the image pixel values in hiding data. Spatial steganographic techniques can be classified into-

- Least significant bit (LSB)
- Pixel value differencing (PVD)
- Edges-based data embedding method (EBE)
- Random pixel embedding method (RPE)
- Mapping pixel to hidden data method
- Labeling or connectivity method
- Pixel intensity-based method
- Texture based method
- Histogram shifting method

Transform domain is a more complex way of hiding information in an image. Various algorithms and transformations are used on the image to hide information in it. Transform domain techniques are classified into-

- Discrete Fourier transformation technique (DFT).
- Discrete cosine transformation technique (DCT).
- Discrete Wavelet transformation technique (DWT).

- Lossless or reversible method (DCT)
- Embedding in coefficient bits

2.8.4 Least Significant Bit (LSB)

LSB based steganography is one of the simplest techniques that hides a secret message in the least significant bit of pixel values without introducing many perceptible distortions. Change in the value of the LSB are imperceptible for human eyes. In LSB method the change in byte is 0.000002%. An illustration of LSB method is shown in Figure 7.

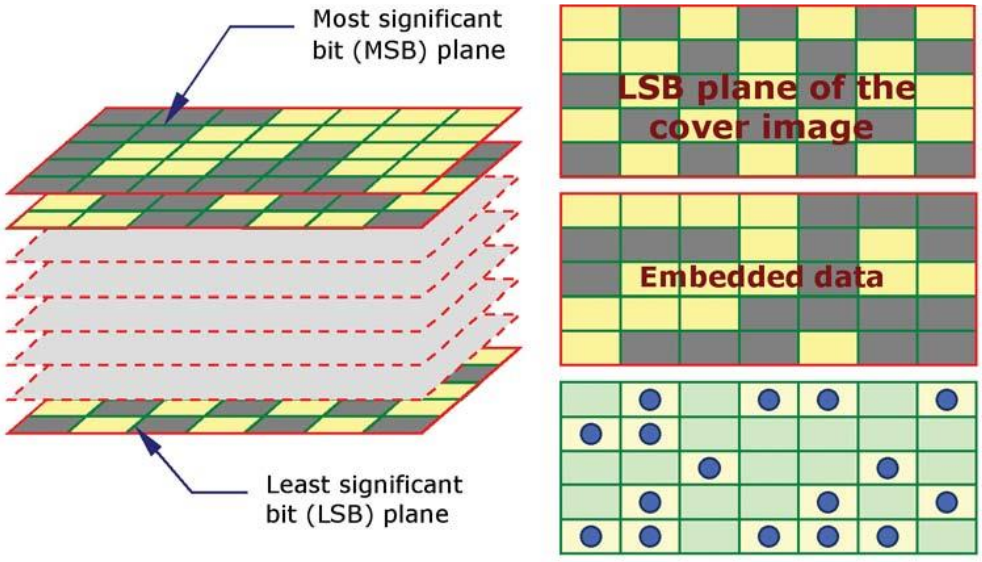

Figure 7 LSB method

- Advantages of spatial domain LSB technique are:
  - There is less chance for degradation of the original image.
  - More information can be stored in an image.
- Disadvantages of the LSB technique are:
  - Less robust, the hidden data can be lost with image manipulation.

3. RELATED WORK

We have designed our use case inspired from two papers. Their work is closely related to our work. The first paper named “Image Steganography embedded with Advance Encryption Standard (AES) securing with SHA-256” combines image steganography, AES, and SHA 256. The proposed method of this paper secures the weaker section which is the key in Advance Encryption Standard using hashing technique. They enhanced the level of concealment of information from unauthorized access and for covert information exchange by encrypting the data and hiding it into a multimedia file known as image. The Secure Hash Algorithm 256 generates a hash key of 256 bits which is an irreversible hashing technique, after that the key is used in the process of encrypting the text with Advance Encryption Standard 256. The cipher text is embedded into a target image using LSB method [11]. The flow chart of the proposed system taken from the paper is shown in Figure 8 and the summary of the paper is presented in Table 2 below.

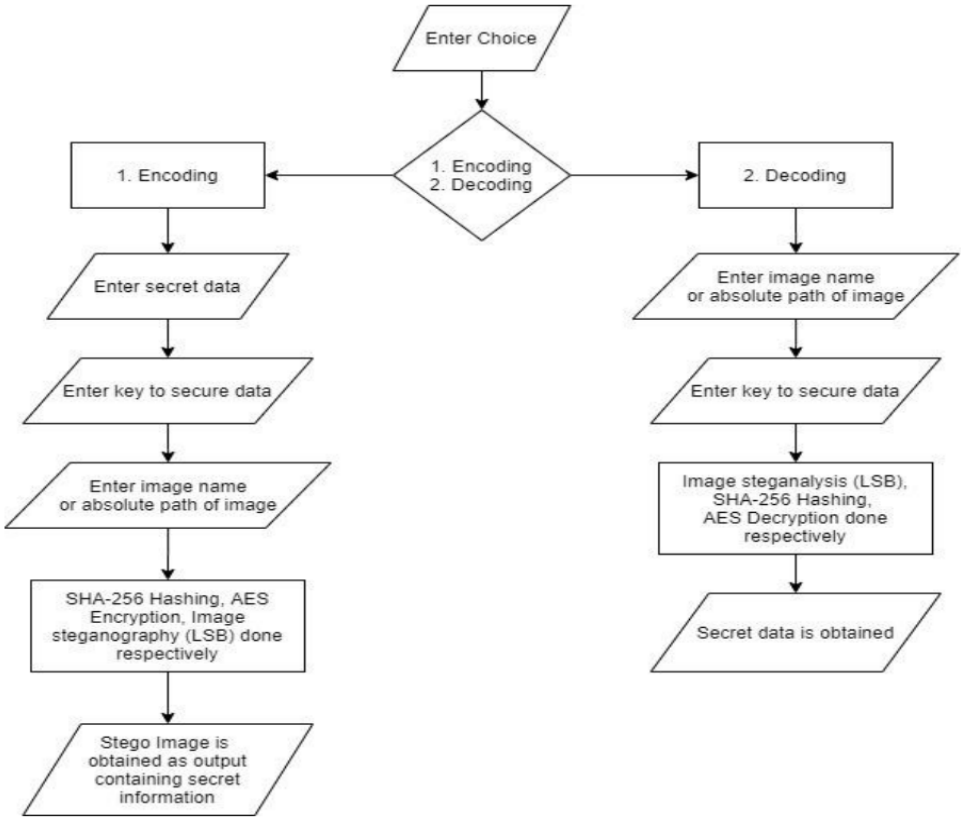

Figure 8 Flow chart of the proposed system from the paper named “Image Steganography embedded with Advance Encryption Standard (AES) securing with SHA-256” [11]

Table 2 Summary of the paper named “Image Steganography embedded with Advance Encryption Standard (AES) securing with SHA-256”

Authors	Method	Observation
Vikas Singhal, Yash Kumar Shukla, Navin Prakash	SHA-256, AES-256, LSB	The proposed system successfully hides the data using image steganography via LSB and surplus the protection of data using cryptographic technique AES-256.

The second paper named “IoMT Security: SHA3-512, AES-256, RSA and LSB Steganography” combines image steganography, AES, RSA, and SHA 256. This paper proposes an information security scheme for IoMT that utilizes AES-256, RSA, SHA3-512 and LSB embedding in medical scans or images [12]. The summary of the paper is presented in Table 3 and the result comparison of the proposed method is illustrated from the paper in Figure 9.

Table 3 Summary of the paper named “IoMT Security: SHA3-512, AES-256, RSA and LSB Steganography”

Authors	Method	Observation
Wassim Alexan, Ahmed Ashraf, Eyad Mamdouh, Sarah Mohamed, Mohamed Moustafa,	SHA-512, AES-256, RSA LSB	The proposed scheme guarantees the secure transmission of medical data through insecure channels or networks.

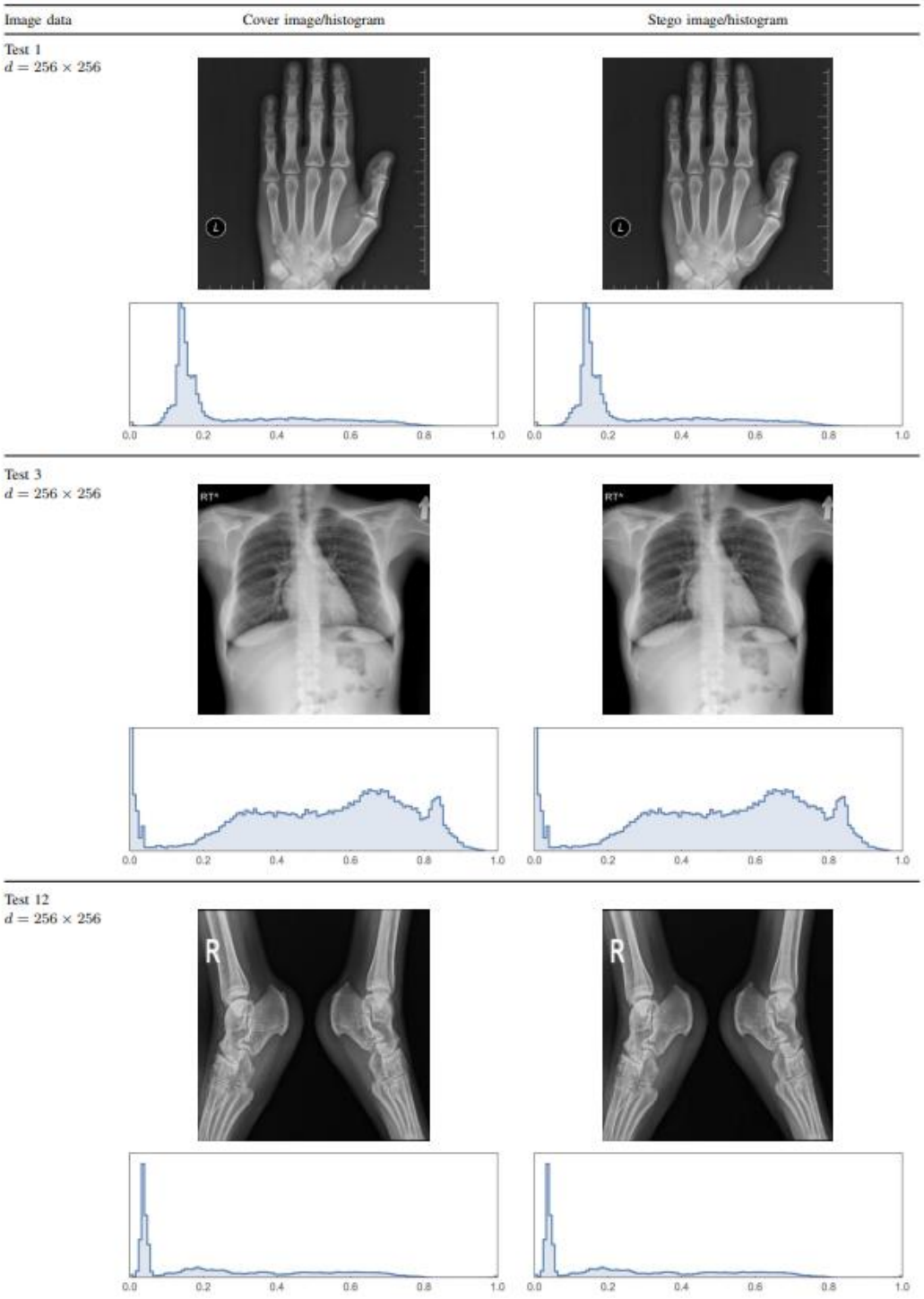

Figure 9 The histograms to compare results of the proposed method taken from the paper named “IoMT Security: SHA3-512, AES-256, RSA and LSB Steganography” [12]

4. CASE STUDY

In the case study, we have designed a hybrid encryption & decryption technique for the transmitter and receiver respectively. It is a combination of Advanced Encryption Standard (AES) 256 and Least Significant Bit (LSB). We have downloaded the dataset from Kaggle [5].

The transmitter and receiver sides have separate implementation steps. On the transmitter side, first sensitive medical data is then encrypted with a symmetric key (k) utilizing the AES-256 algorithm. This encrypted data is transformed into a bitstream. Then bitstream is embedded into a cover medical image using the Least Significant Bit (LSB) method which results in a stego-image. On the receiver side, all the steps are performed in reverse sequence. First, the extraction of the LSB-embedded data is done. Then the extracted data is decrypted using the symmetric key (k). Figure 10 depicts the flow diagram of the implemented case study.

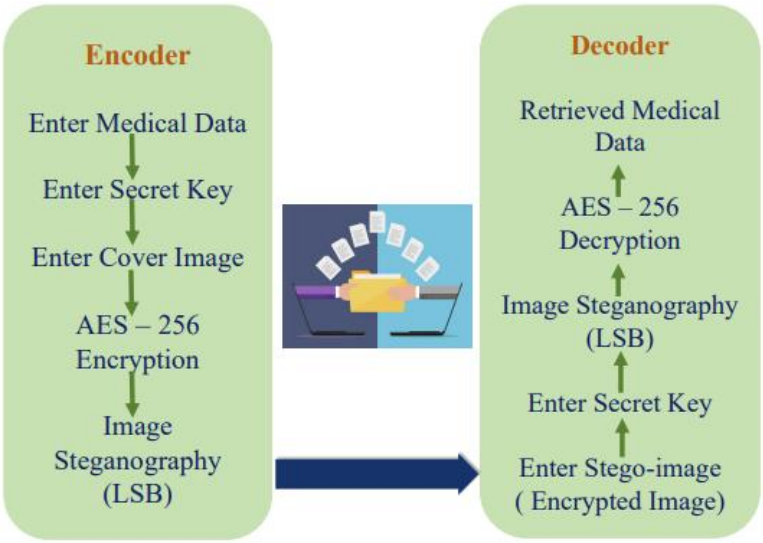

Figure 10 Case study process diagram

4.1 Case Study Implementation

Python script is designed for steganography, the practice of concealing information within other non-secret data, specifically within image files. The code is divided into two primary components: an Encryption Module and a Decryption Module.

The Encryption Module performs the following tasks:

I. Imports essential libraries, such as os, sys, hashlib, Image (from PIL), base64, AES (from Cryptodome.Cipher), and pandas, providing a range of functionalities required for the encryption and encoding process.

II. The AES_encryptor function is responsible for encrypting the input message using the Advanced Encryption Standard (AES) algorithm in Galois/Counter Mode (GCM). It generates a random salt and derives a private_key from the password, employing the Scrypt Key Derivation Function (KDF). The function outputs a dictionary containing the encrypted cipher_text, salt, nonce, and tag, all encoded in base64 format.

III. Several auxiliary functions are defined to facilitate the encoding process:
    a) str_to_bin: Transforms a string into its binary representation.
    b) genData: Generates binary data from the encrypted dictionary with a length prefix for each value.
    c) modPixEncode: Alters the image's pixel values based on the binary data, embedding the encrypted information within the pixels.
    d) encode_image: Accepts an image and an encrypted dictionary as inputs, processes the image data, encodes the binary data into the image, and creates a new image with modified pixel values.

IV. The image_encode function reads an input image file, obtains a secret key (password) from the user, encrypts the input data using the AES_encryptor function, and calls the encode_image function to embed the encrypted message within the image. The encoded image is saved in the "Encoded Image" folder with an '_encoded' suffix.

V. The script extracts data from an Excel file, converts it into a string, and calls the image_encode function to encode the message into an image.

The Decryption Module performs the following tasks:

I. Imports essential libraries, similar to the Encryption Module.
II. The AES_decryptor function is responsible for decrypting the encrypted data using the AES algorithm in GCM. It derives a private_key from the password, employing the Scrypt KDF. The function outputs the decrypted plain_text.
III. Several auxiliary functions are defined to facilitate the decoding process:
    a) binary_to_str: Transforms binary data back into a string.
    b) modPixDecode: Extracts the binary data concealed within the image's pixel values.
    c) decode_image: Accepts an image file path as input, processes the image, extracts the binary data, and reconstructs the encrypted dictionary.
IV. The image_decode function reads an encoded image file, obtains a secret key (password) from the user, calls the decode_image function to extract the encrypted dictionary from the image, and then calls the AES_decryptor function to decrypt the message. It returns the decrypted data.
V. In the __main__ block, the script prompts the user to input the encoded image file name and the secret key. It then calls the image_decode function to extract and decrypt the hidden message from the image, subsequently printing the decoded data to the console.

The Python script of use case demonstrates an effective steganographic method to securely conceal information within image files. This implementation employs a combination of Advanced Encryption Standard (AES) encryption in Galois/Counter Mode (GCM) and image manipulation techniques, ensuring a high level of security for the hidden data. The script is divided into two primary modules—Encryption and Decryption—each containing a series of functions designed to handle the encryption, encoding, decryption, and decoding processes. This approach offers a practical and secure solution for embedding sensitive information within images, which can be useful in a wide range of applications, such as secure communication, data storage, and digital watermarking.

4.2 Results

In order to assess the effectiveness of the steganographic technique employed, we have conducted a thorough analysis comparing the histograms of the original image and the stego image. The histograms serve as a visual representation of the distribution of pixel values in an image, and their identical nature demonstrates that both images are remarkably similar, thereby ensuring that any potential hackers would be unable to distinguish between the two. This evaluation was performed by superimposing one histogram on top of the other, allowing for a direct comparison between the two sets of data.

A wide variety of cover images were utilized for this experiment, encompassing grayscale images, color images, images of differing resolutions, and images of various types. The results consistently showed that the steganographic technique performed exceptionally well, with the histograms of the encrypted images appearing virtually indistinguishable from those of the original images.

In Figure 11, we observe a comparison between grayscale images and their corresponding histograms. It is evident from this comparison that both histograms are virtually identical, which serves as a strong indication of the effectiveness of the steganographic technique in preserving the original image's characteristics. Similarly, in Figure 12, we examine a comparison between color images and their respective histograms. Here, we can also observe that the RGB histograms closely resemble one another, further reinforcing the notion that the employed steganographic method is both robust and reliable in ensuring that the encrypted images remain undetectable to potential hackers.

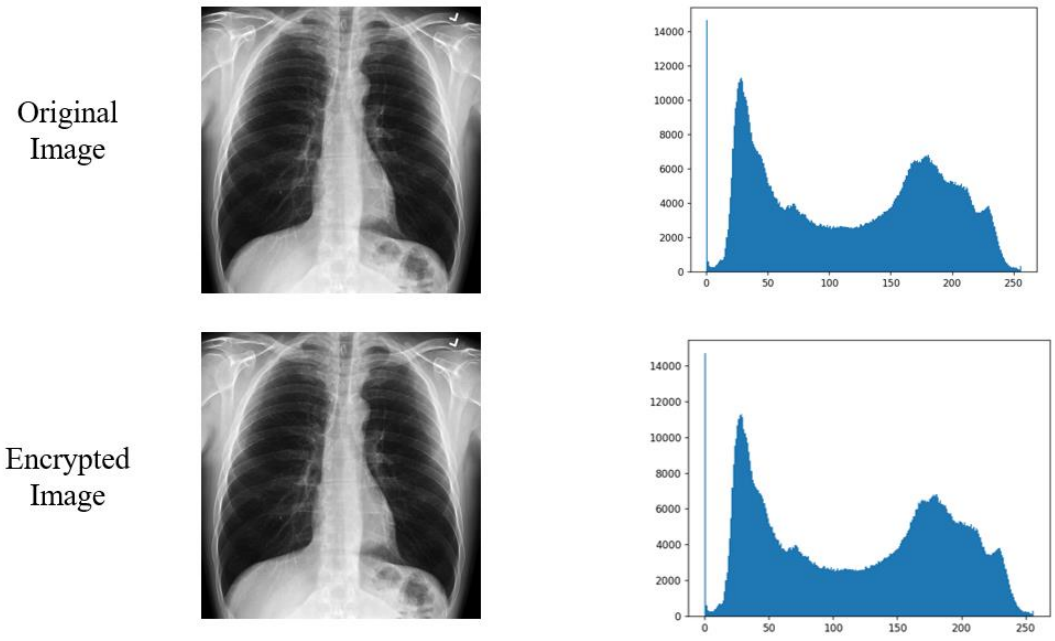

Figure 11 Cover grayscale image, encrypted stego image and their corresponding histograms

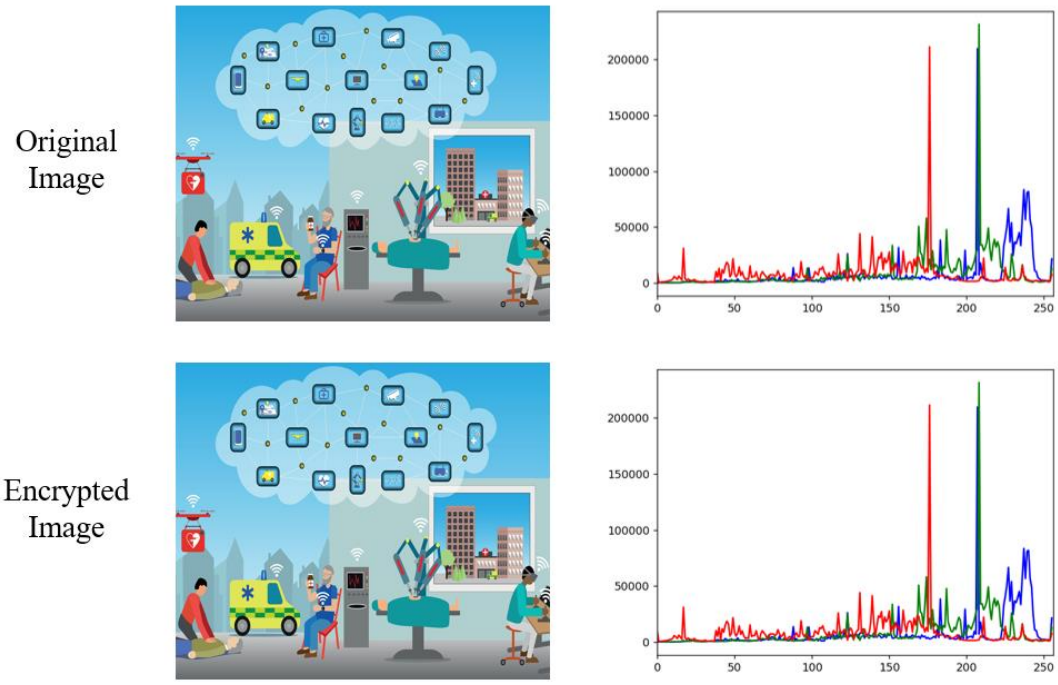

Figure 12 Cover color image, encrypted stego image and their corresponding histogram

5. DISCUSSION & FUTURE WORK

This project aims to explore IoMT device structure and security protocols. Also designs a hybrid cryptographic and steganographic technique to ensure communication security in IoMT devices for medical data transmission. There are lots of techniques of secure data transmission. But AES is the most popular one now a days. As AES is unbreakable until today, it's used in all means of secure data communications. So, in today's perspective it might look unnecessary to combine any other encryption technique with AES. But the computational power is increasing rapidly. So, in near future in the era of quantum computing it might not be impossible to break AES. Lots of research are going on to design more secure algorithms. Our project was a small part of it.

In this design we have combined two techniques. The constraints of this design are-

- Separate cover image is required, which will increase the data size that needs to be transmitted.
- If the secret key that is used in AES needs to be shared with someone, in what encryption technique that will be transmitted?

So, to solve the transmission problem of the secret key of AES, we will design an asymmetric encryption technique like Rivest-Shamir-Adleman (RSA) public key private key system in future and combine with our process. In that process, secret key will be shared encrypting with the public key so that only authorized users can unlock it with private key.

6. CONCLUSION

The use of IoMT is a reality of today's world. Many hospital systems are adopting it or in the process of adopting it. The majority of current research activities focuses on how medical and health-monitoring technologies can help reduce healthcare costs while improving patient health. This is also an objective of many developed hospitals and medical facilities. As a result, protecting this technology has become critical, as the IoMT is vulnerable to a variety of attacks due to its reliance on wireless communications. These attacks have the potential to compromise the system and breach patient's privacy, as well as compromise the confidentiality, integrity, and availability of medical services. We have discussed IoMT security requirements and the techniques of securing this domain and their related assets. Securing healthcare data will increase the effectiveness of IoMT devices benefiting the patients and developing a smart healthcare system.

APPENDIX B. CODE

Encoder code

```
 # import the library module
import os
import sys
import hashlib
from PIL import Image
from base64 import b64encode, b64decode
import hashlib
from Cryptodome.Cipher import AES
import os
from Cryptodome.Random import get_random_bytes
import pandas as pd


def AES_encryptor(plain_text, password):
    # generate a random salt
    salt = get_random_bytes(AES.block_size)
    # use the Scrypt KDF to get a private key from the password
    private_key = hashlib.scrypt(password.encode(), salt=salt, n=2**14, r=8, p=1,
dklen=32)
    #print(private_key)
    # create cipher config
    cipher_config = AES.new(private_key, AES.MODE_GCM)

    # return a dictionary with the encrypted text
    cipher_text, tag = cipher_config.encrypt_and_digest(bytes(plain_text, 'utf-
8'))
    #print(b64encode(cipher_text).decode('utf-8'))
    print(b64encode(cipher_config.nonce).decode('utf-8'))
    return {
        'cipher_text': b64encode(cipher_text).decode('utf-8'),
        'salt': b64encode(salt).decode('utf-8'),
        'nonce': b64encode(cipher_config.nonce).decode('utf-8'),
        'tag': b64encode(tag).decode('utf-8')
    }



# Encode data into image
```

```
def genData(modValue):
    newd = []
    for value in modValue.values():
        for char in value:
            newd.append(format(ord(char), '08b'))
    return newd



# Pixels are modified according to the
# 8-bit binary data and finally returned
def modPix(pix, value):

    datalist = genData(value)
    lendata = len(datalist)
    imdata = iter(pix)

    for i in range(lendata):

        # Extracting 3 pixels at a time
        pix = [value for value in imdata.__next__()[:3] +
                                imdata.__next__()[:3] +
                                imdata.__next__()[:3]]

        # Pixel value should be made
        # odd for 1 and even for 0
        for j in range(0, 8):
            if (datalist[i][j] == '0' and pix[j]% 2 != 0):
                pix[j] -= 1

            elif (datalist[i][j] == '1' and pix[j] % 2 == 0):
                if(pix[j] != 0):
                    pix[j] -= 1
                else:
                    pix[j] += 1
                # pix[j] -= 1


        # Eighth pixel of every set tells
        # whether to stop ot read further.
        # 0 means keep reading; 1 means thec
        # message is over.
        if (i == lendata - 1):
            if (pix[-1] % 2 == 0):
```

```
                if(pix[-1] != 0):
                    pix[-1] -= 1
                else:
                    pix[-1] += 1

        else:
            if (pix[-1] % 2 != 0):
                pix[-1] -= 1

        pix = tuple(pix)
        yield pix[0:3]
        yield pix[3:6]
        yield pix[6:9]



def encode_enc(newimg, data_dict):
    w = newimg.size[0]
    (x, y) = (0, 0)

    for pixel in modPix(newimg.getdata(), data_dict):
        newimg.putpixel((x, y), pixel)
        if (x == w - 1):
            x = 0
            y += 1
        else:
            x += 1


def image_encode(data):
    import re

    # remove non-printable characters from input
    #data = re.sub('[^\x20-\x7E]', '', data)

    img_path = os.path.join("Original Image", input("Enter image file name: "))
    image = Image.open(img_path, 'r')
    secret_key = input("Enter Secret Key: ")

    AES_data = AES_encryptor(data, secret_key)
    salt = AES_data['salt']
    cipher_text = AES_data['cipher_text']
    nonce = AES_data['nonce']
    tag = AES_data['tag']
```

```
    value = {'salt': salt, 'cipher_text': cipher_text, 'nonce': nonce, 'tag':
tag}

    newimg = image.copy()
    encode_enc(newimg, value)

    # Save the encoded image in the same format as the original image
    format = image.format
    if format == 'JPEG':
        extension = 'jpg'
    else:
        extension = format.lower()
    new_img_path = os.path.join("Encoded Image",
os.path.splitext(os.path.basename(img_path))[0] + '_encoded.' + extension)
    newimg.save(new_img_path, format)


excel_file = "data set.xlsx"
sheet_name = "Dataset"
df = pd.read_excel(excel_file, sheet_name=sheet_name, header=None)
arr = df.iloc[5].values
data = "".join(map(str, arr))
print (data)
#secret_key = input("Enter Secret Key: ")
image_encode(data)
```

Decoder code

```
# import the library module
import os
import sys
import hashlib
from PIL import Image
from base64 import b64encode, b64decode
import hashlib
from Cryptodome.Cipher import AES

def AES_decryptor(cipher_text, password, salt, nonce, tag):
    # use the Scrypt KDF to get a private key from the password
    private_key = hashlib.scrypt(password.encode(), salt=b64decode(salt),
n=2**14, r=8, p=1, dklen=32)
    # create cipher config
    cipher_config = AES.new(private_key, AES.MODE_GCM, nonce=b64decode(nonce))
    # return the decrypted text
    decrypted_text = cipher_config.decrypt_and_verify(b64decode(cipher_text),
b64decode(tag))
    return decrypted_text.decode('utf-8')


def decode_data(imdata):
    binary_data = ''
    for pixel in imdata:
        for value in pixel:
            binary_data += str(value % 2)
    # split by 8-bit chunks
    chunks = [binary_data[i:i+8] for i in range(0, len(binary_data), 8)]
    # convert each chunk to its ASCII equivalent
    message = ''.join([chr(int(chunk, 2)) for chunk in chunks])
    # remove padding at end of message
    end_index = message.find('\x00')
    if end_index != -1:
        message = message[:end_index]
    return message


def image_decode(secret_key):
    img_path = os.path.join("Encoded Image", input("Enter encoded image file
name: "))
    image = Image.open(img_path, 'r')
```

```
    imdata = list(image.getdata())

    salt = ''
    cipher_text = ''
    nonce = ''
    tag = ''

    # read the hidden values from the image
    for i in range(4):
        value = ''
        for j in range(8):
            value += str(imdata[i*3+j][0] % 2)
        if i == 0:
            salt = value
        elif i == 1:
            cipher_text = value
        elif i == 2:
            nonce = value
        else:
            tag = value

    decrypted_text = AES_decryptor(cipher_text, secret_key, salt, nonce, tag)
    print('Decrypted message:', decrypted_text)


secret_key = input("Enter Secret Key: ")
image_decode(secret_key)
```

REFERENCES

[1] Fabio Duarte, "Number of IoT Devices (2023)," *explodingtopics.com/*, 2023. https://explodingtopics.com/blog/number-of-iot-devices (accessed Apr. 18, 2023).

[2] MarketWatch, "Internet of Medical Things (IoMT) Market Global Trends, Market Share, Industry Size, Growth, Opportunities and Market Forecast 2028," *marketwatch.com*, 2023. https://www.marketwatch.com/press-release/internet-of-medical-things-iomt-market-global-trends-market-share-industry-size-growth-opportunities-and-market-forecast-2028-2023-03-29 (accessed Apr. 28, 2023).

[3] A. Ghubaish, T. Salman, M. Zolanvari, D. Unal, A. Al-Ali, and R. Jain, "Recent Advances in the Internet-of-Medical-Things (IoMT) Systems Security," *IEEE Internet Things J.*, vol. 8, no. 11, pp. 8707–8718, 2021, doi: 10.1109/JIOT.2020.3045653.

[4] R. Hireche, H. Mansouri, and A.-S. K. Pathan, "Security and Privacy Management in Internet of Medical Things (IoMT): A Synthesis," *J. Cybersecurity Priv.*, vol. 2, no. 3, pp. 640–661, 2022, doi: 10.3390/jcp2030033.

[5] Mayor Muriel Bowser, "Personal Emergency Response System (PERS)," *odr.dc.gov/book/path/PERS*. https://odr.dc.gov/book/path/PERS (accessed Apr. 17, 2023).

[6] L. P. Malasinghe, N. Ramzan, and K. Dahal, "Remote patient monitoring: a comprehensive study," *J. Ambient Intell. Humaniz. Comput.*, vol. 10, no. 1, pp. 57–76, 2019, doi: 10.1007/s12652-017-0598-x.

[7] Satavisa Pati, "Top 10 IoMT Trends and Use Cases in Healthcare for 2023," *www.analyticsinsight.net*, 2022. https://www.analyticsinsight.net/top-10-iomt-trends-and-use-cases-in-healthcare-for-2023/ (accessed Apr. 17, 2023).

[8] N-able, "Understanding AES 256 Encryption," *www.n-able.com*, 2019. https://www.n-able.com/blog/aes-256-encryption-algorithm (accessed Feb. 23, 2023).

[9] Baivab Kumar Jena, "What Is AES Encryption and How Does It Work?," *www.simplilearn.com*, 2023. https://www.simplilearn.com/tutorials/cryptography-tutorial/aes-encryption (accessed Mar. 02, 2023).

[10] M. Hussain, "A Survey of Image Steganography Techniques A Survey of Image Steganography Techniques Mehdi Hussain and Mureed Hussain," no. February, 2015.

[11] M. V. Singhal*, M. Y. K. Shukla, and D. N. Prakash, "Image Steganography embedded

with Advance Encryption Standard (AES) securing with SHA-256," *Int. J. Innov. Technol. Explor. Eng.*, vol. 9, no. 8, pp. 641–648, 2020, doi: 10.35940/ijitee.h6442.069820.

[12] W. Alexan, A. Ashraf, E. Mamdouh, S. Mohamed, and M. Moustafa, "IoMT Security: SHA3-512, AES-256, RSA and LSB Steganography," *Proc. - 2021 8th NAFOSTED Conf. Inf. Comput. Sci. NICS 2021*, no. December, pp. 177–181, 2021, doi: 10.1109/NICS54270.2021.9701567.

[13] Ferguson, J.E.; Redish, A.D. Wireless communication with implanted medical devices using the conductive properties of the body. *Expert Rev. Med. Devices* 2011, *8*, 427–433.

[14] Kos, A.; Milutinovi´c, V.; Umek, A. Challenges in wireless communication for connected sensors and wearable devices used in sport biofeedback applications. *Future Gener. Comput. Syst.* 2019, *92*, 582–592.

[15] Lone, T.A.; Rashid, A.; Gupta, S.; Gupta, S.K.; Rao, D.S.; Najim, M.; Srivastava, A.; Kumar, A.; Umrao, L.S.; Singhal, A. Securing communication by attribute-based authentication in hetnet used for medical applications. *EURASIP J. Wirel. Commun. Netw.* 2020, *146*, 146.

[16] Toscano, E.; Bello, L.L. Comparative assessments of IEEE 802.15. 4/ZigBee and 6LoWPAN for low-power industrial WSNs in realistic scenarios. In Proceedings of the 9th IEEE International Workshop on Factory Communication Systems, Lemgo, Germany, 21–24 May 2012.

[17] Navya, V.; Deepalakshmi, P. Threshold-based energy-efficient routing for transmission of critical physiological parameters in a wireless body area network under emergency scenarios. *Int. J. Comput. Appl.* **2021**, *43*, 367–376.

[18] Sundaresan, S.; Doss, R.; Zhou, W. RFID in healthcare–current trends and the future. In *Springer Series in Bio-/Neuroinformatics*; Kasabov, N., Ed.; Springer: Berlin/Heidelberg, Germany, 2015; Volume 5, pp. 839–870.

[19] Chen, X.; Zhu, H.; Geng, D.; Liu, W.; Yang, R.; Li, S. Merging RFID and blockchain technologies to accelerate big data medical research based on physiological signals. *J. Healthc. Eng.* **2020**, *2020*, 2452683.

[20] P. Kasyoka, M. Kimwele and S. Mbandu Angolo, "Certificateless pairing-free authentication scheme for wireless body area network in healthcare management system", *J. Med. Eng. Technol.*, vol. 44, no. 1, pp. 12-19, 2020.

[21] T. Belkhouja, S. Sorour and M. S. Hefeida, "Role-based hierarchical medical data encryption for implantable medical devices", Proc. IEEE Global Commun. Conf. (GLOBECOM), pp. 1-6, Dec. 2019.

[22] Sun, Y.; Lo, F.P.-W.; Lo, B. Security and privacy for the internet of medical things enabled healthcare systems: A survey. IEEE Access 2019, 7, 183339–183355.

[23] Davis, J. Ransomware Attacks Cost Healthcare Sector at Least $160M Since 2016. Health IT Security. Available online: https: //healthitsecurity.com/

POSTER PRESENTATION

This work is presented in 2023 Student Research and Creative Endeavor Symposium on 24 March 2023 at Purdue University Fort Wayne.