Systems and Methods for Image Encryption Using Steganography and XOR Operations

1st Shailaja Uke
Vishwakarma Institute of Technology,
Pune, India
shailaja.uke@vit.edu

2nd Soham Nimale
Vishwakarma Institute of Technology,
line 4: Pune, India
soham.nimale22@vit.edu

3rd Sneha Jain
Vishwakarma Institute of Technology,
Pune, India
sneha.jain221@vit.edu

4th Sneha Bhat
Vishwakarma Institute of Technology,
Pune, India
bhat.sneha22@vit.edu

5thst Somnath Ghadge
Vishwakarma Institute of Technology,
Pune, India
nitin.ghadge22@vit.edu

6th Soham Gargote
Vishwakarma Institute of Technology,
Pune, India
balasaheb.soham221@vit.edu

7th Tanuj Somani
Vishwakarma Institute of Technology,
Pune, India
tanuj.somani22@vit.edu

Abstract— **Cybersecurity is a significant part of safe data transmission, particularly through networks such as the Internet. Confidential medical or military images and personal images of individuals and organizations are a big part of such data. These images need a very high level of security to prevent access by unauthorized users. Unauthorized access to such images can lead to significant financial and life losses. To prevent this, we incorporate various encryption-decryption algorithms and concepts like steganography into a system for safe, reliable, efficient, and secure image storage and transfer. We tune each algorithm implemented in the system individually to produce the best results upon integration with other algorithms. Further, we provide users with Graphic User Interface that helps them interact with this system conveniently and efficiently. Finally, we test the system using various medical images and present the results.**

Keywords— Cybersecurity, Image Processing, Optical Encryption, RSA, Steganography.

I. INTRODUCTION

N this digital age, people share billions of images using networks like the Internet. Ideally, only the intended senders and receivers should have access to these images. However, weak security systems often lead to unauthorized access by third parties or breaches of privacy for individuals and organizations sharing private or confidential data [1]. A robust and secure encryption-decryption system is necessary to prevent this. This system should also be fast, reliable, efficient, and secure. However, improper implementation of algorithms in such a system can cause permanent data loss or unauthorized access to sensitive data. Plagiarism is also a significant issue when images are shared using the Internet. Implementation of Optical encryption to prevent it and also provide security can be done [2]. Steganography is another technique often used to hide original data under another block of data. Hiding text under an image using steganography is a common practice. The encryption of properties of images in text form, which we use in our system for decryption purposes, can be done using RSA (Rivest-Shamir-Adleman) encryption algorithm. Thus, in this paper, we suggest a secure encryption-decryption system that integrates multiple such encryption and data-hiding techniques to protect private images while sharing without any data loss.

II. LITERATURE REVIEW

Encryption-Decryption systems use various algorithms. These Algorithms used in the encryption and decryption of data are called Cipher (data can be in the form of images or text) [3]. We perform encryption to modify given data into an unreadable format by implementing Cipher to protect data from unauthorized users [3]. Meanwhile, in the decryption, we convert the modified unreadable data back to the original form when authorized users access it [3]. This conversion from modified data to un-modified data is possible only if the users use the associated "Key" [4]. The "Key" serves as a password to get the original data back [4]. Thus, correct decryption is possible only if the appropriate key is used [4]. Only authorized users have the key for decryption [4]. The longer the key, the better the encryption since it would take significant time for unauthorized users to find the correct key. But longer keys would lead to increased encryption and decryption time complexities compared to shorter keys [5, 6]. Many algorithms can be implemented for image encryption [3]. The main difference between the encryption of image and text is that some error in the decrypted image is acceptable since the human eye cannot detect such small changes [3]. But for text encryption-decryption, the original text must always remain the same after the decryption to avoid errors [7]. Thus, this gives the freedom to use complex algorithms where some data loss in the images is acceptable. For such image-based encryption algorithms, the modified image must be super sensitive to the value of the key; little change in the value of the key must correspond to a high magnitude of disturbance in the image so that it is harder to find the correct key [3].

Another security layer can be affixed using Steganography. Steganography is a crucial technique generally used for safe information transfer by hiding the original information in some other form [8]. Steganography is often implemented to hide data inside an image [9]. Using Cryptography, we manipulate/encrypt the original data to protect it from unauthorized users [4]. But, in Steganography, we hide the very fact that data is being secretly exchanged [8]. Thus, Steganography combined with Cryptography can result in a powerful Encryption-Decryption system that is hard to recognize and decrypt using brute force algorithms. Implementation of Steganography is possible by using various algorithms [9, 10]. Using Steganography, it is necessary to hide data such that only minor apparent changes are visible when the original image is compared to the output image with the hidden messages by any unauthorized user [11]. Thus, a good Steganography algorithm must be capable of hiding a large amount of information without giving out visual hints to users/algorithms [10]. Because of this, the commonly used simple LSB-based algorithms are not suitable for Steganography on their own [11]. Also, using the two Least Significant Bits to increase the quantity of data hidden in the output image is not a good practice since it reduces the overall security by increasing the chances of getting detected [11]. But, using 24-Bit RGB pixels, it is possible to hide a large quantity of data even with one Least Significant Bit based algorithm.

After implementing Steganography, RSA can be applied to encrypt the image properties that we use in the system for decryption. The LSB-based Steganography algorithms are not robust on their own [12]. Thus, using the RSA algorithm helps make the encryption highly secure. We use RSA for generating public and private keys [6, 13]. The Use of two different keys for encryption and decryption is called Asymmetric cryptography [13]. Both the keys are mathematically related to each other [13, 17]. The use of asymmetric keys is generally very time-consuming compared to symmetric keys [5] but preferred because only the receiver has the private key capable of decrypting the encrypted image [13]. Thus, even if a user with the public key leaks the key, unauthorized users cannot use it to access the original images since only the private key owner can do so. Combining optical data encryption with RSA key encryption gives a highly secure system. To further increase security, we implement Steganography in the proposed system and use RSA to encrypt the image properties instead of implementing RSA for encrypting the actual pixel values. Using these methods together, a fast, secure, and efficient system for image encryption would be possible.

III. Methodology

A. *Flowcharts*

Separate flowcharts for the encryption and decryption system are depicted in Fig. 1. And Fig. 2. respectively. Each flowchart consists of several steps based on various encryption-decryption techniques.

1) *Encryption System Flowchart and Description*

a) Input Parameters

The image to be encrypted is the Input Image. The sender side of the system is employed to encrypt this image. The Top Layer refers to the image under which the users wish to hide their encrypted image. The size of the Top layer must be larger than the size of the Input Image to make Steganography possible [14]. The user has to enter a Public Key value for the RSA encryption step. “Save Optically Encrypted” and “Save Steganography Analysis Image” are two Boolean variables used to determine whether the user wants to save semi-encrypted images produced during encryption or continue without doing so. The Output Location parameter is needed to store the encrypted image produced in the appropriate location.

b) Optical Encryption

Optical encryption based on XOR operations is done in this step.

c) Steganography

Optically Encrypted Image is injected pixel by pixel into the Top Layer using the steganography algorithm of this system. Garbage values are then injected into the Top Layer using the same algorithm. Boolean “Save Steganography Analysis Image” is used to decide whether the user wants to perform a Steganography analysis and save the results produced or not. The modified Top Layer image and the un-modified Top Layer are compared in the Steganography analysis.

d) RSA Encryption

It is used to encrypt the image properties. The cipher text produced as a result of this step is necessary for decryption at the receiver’s side.

2) *Decryption System Flowchart and Description*

a) Input Parameters

The users on the receiving side must provide the Input Encrypted Image to the system for decryption along with the correct Cipher text and Private key. Users must also enter the Output Image Path parameter to save the decrypted image generated in the appropriate location.

b) RSA Decryption

RSA decryption algorithm is applied to the input cipher text using the private key to produce decrypted image properties.

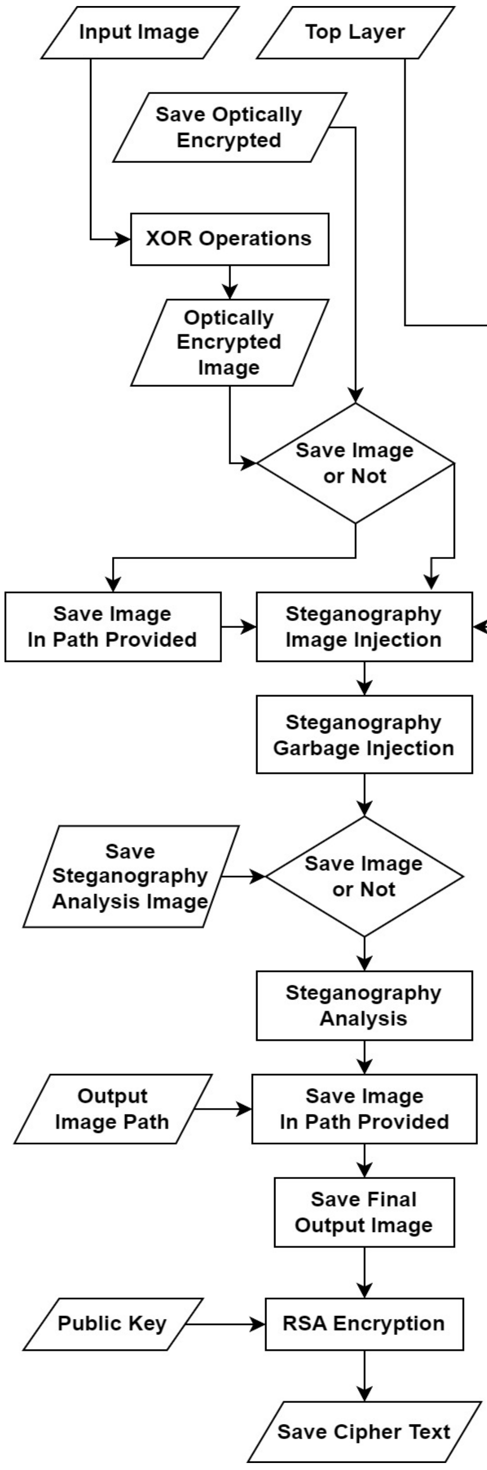

Fig. 1. Flowchart for encryption algorithm implemented in the system.

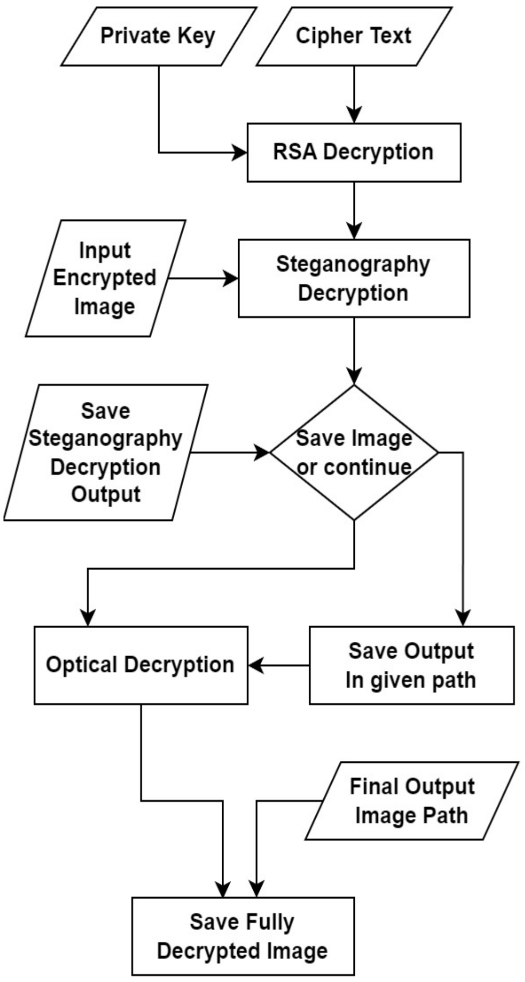

Fig. 2. Flowchart for decryption algorithm implemented in the system.

c) Steganography Decryption

The image properties produced in the RSA decryption are used to extract the concealed image from the altered Top Layer without being affected by the garbage values injected during the encryption.

d) Optical Decryption

The Optical decryption process employs XOR operations to decrypt the image. The system saves the resulting image in the designated path specified in the input parameters.

B. Method

The proposed system utilizes the Python programming language, leveraging its object-oriented nature to divide algorithms and their associated code into various steps. Integrating these discrete steps, we created two user-friendly Graphic User Interfaces (GUI) for encryption and decryption. A separate GUI was created because the input parameters required for each of them to function were different. The GUI for encryption is displayed in Fig. 3. The white frame present in the GUI window in the Fig. 3 acts as a log. This log frame is an invaluable tool for communicating with users. If the users do not provide the system with any essential input parameter, the log frame will display a message to warn users about the same.

1) Optical Encryption and Decryption

Optical encryption and decryption are both implemented using Exclusive-OR (XOR) operations. For encryption, the XOR operation of 8-bit RGB values of each pixel is done with the corresponding 8-bit RGB values of the next pixel. The entire image is optically encrypted by iterating through each pixel and performing this operation. Thus, a Raster scan is used for optical encryption [3]. Also, the first pixel from the Input Image is saved at the end of the optically encrypted image because it is necessary for the decryption. Thus, the first pixel acts like a pseudo key during the decryption. Fig. 4. depicts an example of optical encryption using this method. For decryption, the XOR operation of 8-bit RGB values of each pixel is done with 8-bit RGB values of the previous pixel. This procedure is the same as the Raster scan used for encryption but in the reverse direction from bottom to top. On comparing the result produced using this decryption process with the original image, one can observe that the resulting image is shifted one pixel in the backward direction. That is, the first pixel is now present at the end of the image, the second pixel is now at the previous position of the first pixel, and the third pixel is at the old location of the second pixel, and so on. To fix this issue, we swap and move the pixels in the forward direction by one unit.

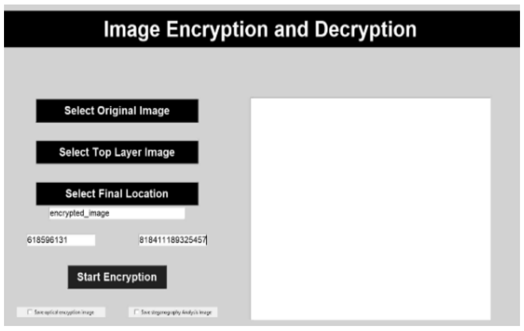
Image Encryption and Decryption
Select Original Image
Select Top Layer Image
Select Final Location
Start Encryption
Save optical encryption image
Save steganography Analysis Image

encrypted_image

618596131

818411189325457

Fig. 3. Use of Graphics User Interface with multiple input parameters for Encryption.

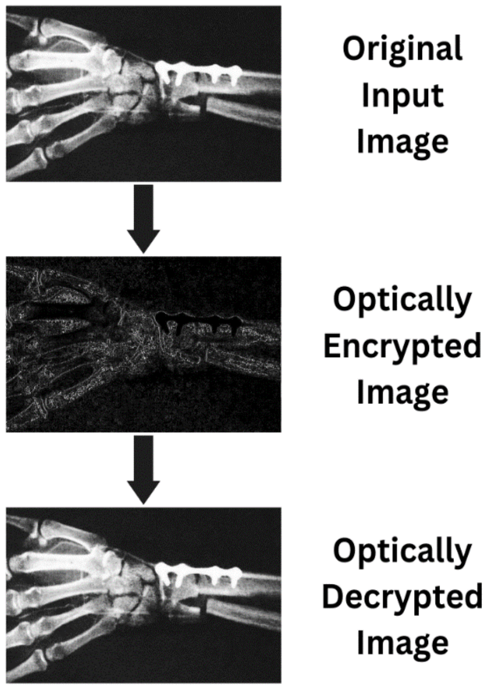

Fig. 4. Fig. 4. Picture depicting a medical image before optical encryption, after optical encryption, and after optical decryption. No loss in the image is seen.

2) *Steganography: Encryption, Decryption, and Analysis*

The steganography implementation in our system involves the following steps:

a) Assignment of Indices

Each pixel in the Top Layer image is assigned a unique pixel index. The pixel at the top left corner of the image is assigned an index of 0. The next pixel corresponds to the next pixel in the Raster scan of the Top Layer image starting from the top left corner. The index assigned to the next pixel is the index assigned to the current pixel plus one.

b) Calculation of Skip Pixels Variable

The Skip Pixels variable is equal to the area of the Top Layer divided by the area of the Input Image rounded to the greatest integer less than or equal to the resulting value.

c) Iteration over pixels of the Top Layer

We iterate over each pixel of the Top Layer image to inject the pixels of the Optically Encrypted Image into its least significant red-green-blue bit together with some garbage bits.

d) Valuable and Garbage bits Injection.

We inject all the information about the input image necessary to reconstruct it using three bit-sequences corresponding to the bit-wise concatenation of each 8-bit red-green-blue channel value of each pixel present in the input image. After each injection in the Top Layer image, we skip the number of pixel indices equal to the value of the Skip Pixels variable. Later, the LSBs of the pixels present at these indices are filled with garbage values to decrease the chances of the steganography getting detected. We used the Raster scan method for iterating through each pixel of the Top Layer image to inject the bit sequence of the input image in it along with the garbage values at appropriate indices.

e) Filling garbage bits in empty regions

After injecting all the bits of the optically encrypted image with the garbage bits into the Top Layer, any empty areas remaining in the Top Layer image are filled with random garbage LSB to avoid detection.

During encryption, the pixels from the optically encrypted image are concealed in the Top Layer image provided by the user along with some garbage values using this algorithm. These random 0 or 1 garbage bits are put into the Top Layer by replacing the LSBs of each pixel present in the Top Layer at the pixel indices other than the ones containing information about the original input image. Bit injection is performed in all the RGB channels of the image separately for each of the three bit-sequences obtained from the input image. From the algorithm described for steganography, we can conclude that the length and breadth of the original image are of extreme value for encryption and decryption. The decryption of a modified Top Layer image is very difficult without knowing the actual length and breadth variables of the associated Input Image. Thus, to protect these variables, we make use of the RSA algorithm. We have referred to these variables as the "properties" of the input image. A similar algorithm can be used to decrypt the modified Top Layer image, that is, by extracting the pixel indices that are divisible by the value of the variable Skip Pixels. Again, iteration in the Raster scan method is necessary for correct decryption. Steganography analysis is done in parallel while encrypting the image. We put grey pixels in the analysis image if we spot any difference in the corresponding pixels of the modified and un-modified Top Layer image. If no difference exists, we put a white pixel instead.

3) *RSA Encryption and Decryption*

The image properties (the length and breadth) of the original image are converted into a cipher number using the public key from input parameters and the RSA algorithm. The cipher number is re-formatted into a String to obtain the cipher text. For decryption using the private key, the cipher text needs to be re-formatted again into a number (Integer) format. RSA algorithm needs a high time complexity. Because of this drawback, we cannot use RSA for the encryption of all the pixels of the input image individually. This process is unfeasible and computationally expensive. Hence, only valuable information, that is, the necessary properties of the input image are encrypted using the RSA algorithm. These properties are sufficient to get the correctly decrypted image back from the Top Layer.

IV. Results and Discussions

This system underwent extensive testing by encrypting and decrypting medical images multiple times. During the testing, we came across many errors and problems. We tuned our encryption-decryption system several times to solve these issues. First, as depicted in Fig. 5., the simple LSB-based Steganography implementation is easily detected during the analysis. The Steganography analysis step provided in the system is helpful for re-selecting and replacing the Top Layer if it does not perform well. It also helps check if the implemented steganography is secure or easily detectable. To fix the issue found during analysis, we changed the steganography algorithm to put the input image's bit-sequence in the Top Layer image at a random constant distance rather than one after the other. The result is depicted in Fig. 6. But, when the user changes the Top Layer to a bigger or smaller resolution image, the algorithm seems to fail. We see this issue in Fig. 7., where we used a larger resolution of the Top Layer image. To tackle this, we calculate the perfect maximum distance that can be allowed between any two pixels of the Top Layer, such that the entire input image can be hidden into the Top Layer image when injected bit-wise in those pixels. A detailed explanation of this algorithm is in section 3.2.2. The results after implementing this solution are depicted in Fig. 8. In this way, we improved the Steganography layer of our system to a point where as long as the Top Layer image had more resolution than the input image, the steganography would be successful and hard to detect.

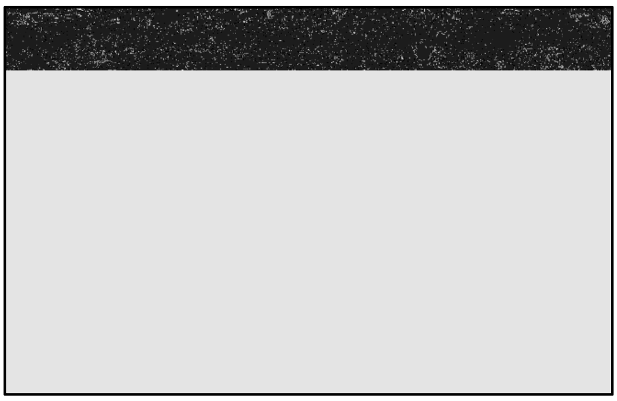

Fig. 5. Steganography analysis output using simple LSB-based steganography. The black pixels at the top of the figure represent the detected bit sequence of the input image. The grey pixels represent the unused pixels of the Top Layer Image.

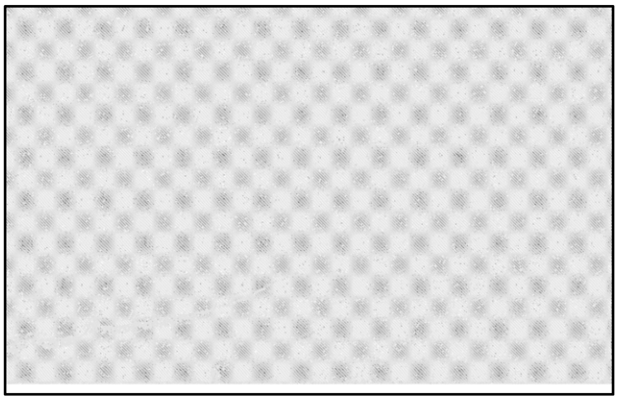

Fig. 6. Steganography analysis output for random constant distance between bit-sequence of the input image while injecting into the Top Layer. The black pixels at the top of the figure represent the detected bit sequence of the input image. The small layer of white pixels at the bottom of the analysis image represents the unused pixels of the Top Layer Image.

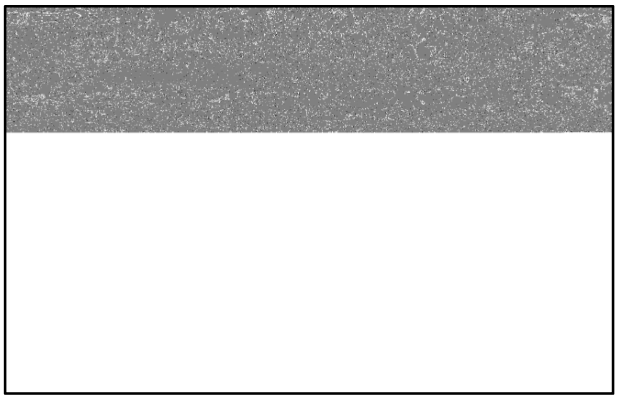

Fig. 7. Steganography analysis output for the same algorithm used in Fig. 6., but for a different Top Layer image. The black pixels at the top of the figure represent the detected bit sequence of the input image. The white pixels at the bottom of the figure represent the unused pixels of the Top Layer Image.

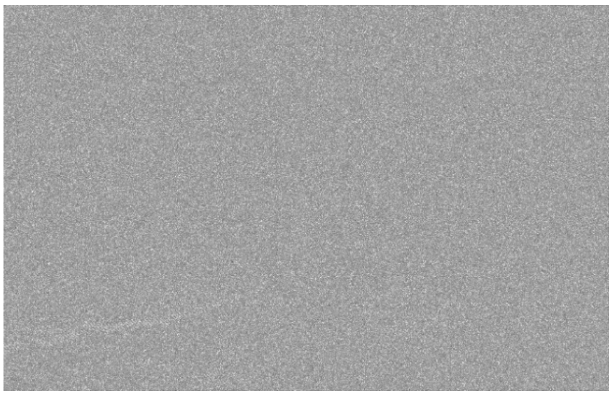

Fig. 8. Steganography analysis output for the perfect maximum distance method implemented along with garbage bit injection. The black pixels in the figure represent the detected bit sequence of the input image. The white pixels represent the unused pixels of the Top Layer image.

The application of multi-threading and GPU for parallel processing of the images significantly reduced the time for encryption and decryption per image. We also provided flexible Private and Public keys. Users can use their keys of varying sizes depending on their needs. We tested the RSA algorithm for encryption of the entire image, but as mentioned in section 3.2.3, it is very unfeasible. Thus, we use RSA only for encryption of the image properties. The GUI provided along with the system helps in convenient and efficient encryption and decryption. The log shown in the GUI part helps debug the errors, especially when the users provide fewer input parameters than expected for encryption or decryption. The wrong format of the input image causes the encryption to fail sometimes. To handle this type of error, we ask users to either input a specific image format that the system accepts or give permission to the system to change the image format to the required one. For the latter, after completing the decryption, we re-format the input image back to the original image format. Thus, in this way, the system was evaluated based on its security, speed, space utilization, efficiency, reliability, and decryption accuracy and tuned to provide the best results. The final proposed system performs exceptionally well in all aspects of these tests.

V. Conclusion

Thus, in this paper, we make use of Optical encryption, Steganography, and RSA encryption methods to build a secure and efficient system for image encryption and decryption. We optimize these algorithms individually to make them perform well after overall integration into a single system. The proposed system can provide individuals and organizations with a one-stop solution for the secure transfer of private/confidential images. With the use of multiple encryption techniques, this system makes it very difficult for unauthorized users to try and access shared images. Also, Users can use their own public and private keys of varying sizes, making this system very flexible. Users can also analyse the steganography process using the features provided. Implementation of Asymmetric-Key cryptography makes this system leak-proof for the public key side. Further research and work on the integration part of the system using various tools would help produce better results [15]. Thus, the proposed encryption-decryption system is an excellent solution for safely sharing and storing medical, military, and private individual images through various networks.

References

[1] Nilesh Uke, "Healthcare 4.0 enabled lightweight security provisions for medical data processing." Turkish Journal of Computer and Mathematics Education (TURCOMAT) doi: 10.17762/turcomat.v12i11.5858

[2] 14Millán, María & Pérez-Cabré, Elisabet. (2011). Optical Data Encryption. Optical and digital image processing: fundamentals and applications. 739-767. 10.1002/9783527635245.ch33.

[3] Moradi Rad, Reza & Attar, Abdolrahman & Ebrahimi Atani, Reza. (2013). A new fast and simple image encryption algorithm using scan patterns and XOR. International Journal of Signal Processing, Image Processing and Pattern Recognition. 6. 275-290. 10.14257/ijsip.2013.6.5.25.

[4] Mohammed, Abdalbasit & Varol, Nurhayat. (2019). A review paper on cryptography. 1-6. 10.1109/ISDFS.2019.8757514.

[5] Agrawal, Ekta & Pal, Parashu. (2017). A secure and fast approach for encryption and decryption of message communication. International Journal of Engineering Science and Computing. 7. 5.

[6] Xin Zhou and Xiaofei Tang, "Research and implementation of RSA algorithm for encryption and decryption," Proceedings of 2011 6th International Forum on Strategic Technology, Harbin, Heilongjiang, 2011, pp. 1118-1121, doi: 10.1109/IFOST.2011.6021216.

[7] Khyioon, Zainab & Hussein AL-Qinani, Iman & Neamah, Farah. (2019). Subject Review : Key generation in different cryptography algorithm. International Journal of Scientific Research in Science, Engineering and Technology. 230-240. 10.32628/IJSRSET196550.

[8] Hamid, N., Yahya, A., Ahmad, R.B. and Al-Qershi, O.M., 2012. Image steganography techniques: an overview. International Journal of Computer Science and Security (IJCSS), 6(3), pp.168-187.

[9] C.C. Chang, J.Y. Hsiao, C.S. Chen, Finding optimal least-significant-bit substitution in image hiding by dynamic programming strategy, Pattern Recognition 36 (2003) 1583–1595.

[10] EL-Emam, N. N. (2007). Hiding a large amount of data with high security using steganography algorithm. Journal of Computer Science, 3(4), 223-232. https://doi.org/10.3844/jcssp.2007.223.232

[11] Chandramouli, Rajarathnam & Memon, Nasir. (2001). Analysis of LSB based image steganography techniques. 3. 1019 - 1022 vol.3. 10.1109/ICIP.2001.958299.

[12] N. Akhtar, P. Johri and S. Khan, "Enhancing the security and quality of LSB based image steganography," 2013 5th International Conference and Computational Intelligence and Communication Networks, Mathura, India, 2013, pp. 385-390, doi: 10.1109/CICN.2013.85.

[13] Liestyowati, Dwi. (2020). Public key cryptography. Journal of Physics: Conference Series. 1477. 052062. 10.1088/1742-6596/1477/5/052062.

[14] Dar, Zain. (2018). The Pigeonhole principle and it's applications. 10.13140/RG.2.2.26177.45923.

[15] S N Uke, A R Mahajan and R C Thool. Article: UML Modeling of Physical and Data Link Layer Security Attacks in WSN. International Journal of Computer Applications 70(11):25-28, May 2013