

IoT-DRIVEN SECURE IMAGE TRANSMISSION USING AN OPTIMIZED ENCRYPTION METHODOLOGY

¹Sahaja,²Bhavani,³Raja Shekhar

^{1,2,3}Students

Department of ECE

ABSTRACT

With the rapid expansion of the Internet of Things (IoT), a vast amount of sensitive image data is being transmitted across low-power, distributed networks. These systems are often vulnerable to security breaches due to limited computational capabilities and inadequate encryption mechanisms. This paper presents an optimized image encryption methodology tailored for IoT environments, addressing the need for lightweight, fast, and secure data protection. The proposed approach combines advanced cryptographic algorithms with pixel-level scrambling and key-dependent diffusion techniques to ensure confidentiality, integrity, and resistance against common attacks. Experimental analysis demonstrates that the method achieves high security with minimal computational overhead, making it ideal for real-time image transmission in IoT-based applications such as surveillance, healthcare, and smart cities.

I. INTRODUCTION

The Internet of Things (IoT) has revolutionized modern computing by enabling interconnected devices to communicate and share data autonomously. In many applications—including smart surveillance, medical imaging, and industrial monitoring—images are a primary form of data being collected and transmitted. However, as IoT devices are resource-constrained and often deployed in unsecured environments, securing image data transmission becomes a significant challenge.

Traditional encryption algorithms, though effective, are often computationally intensive and ill-suited for real-time processing on IoT devices. This necessitates the development of

lightweight, efficient, and robust image encryption techniques that can operate under IoT constraints without compromising security. Furthermore, attackers may exploit weak encryption or intercept image data during transmission, leading to potential privacy violations and data tampering.

This paper introduces an optimized encryption methodology for IoT-driven image transmission. The approach employs a hybrid of pixel shuffling, key-sensitive substitution, and lightweight cryptographic primitives to enhance security while maintaining real-time performance. The methodology is designed with scalability and low-resource deployment in mind, ensuring it can be integrated into a variety of IoT applications.

II. LITERATURE SURVEY

The development of steganography and cryptography came about as a result of the numerous attacks and threats that engulfed smart phone technologies. Steganographic approaches have been used over time to securely protect data communication. According to (Dengre, Gawande, & Deshmukh, 2013), the term steganography originates from ancient Greek “Stegano” which means something that is covered or secret and “Graphia” or “Graptos” which means putting down something or writing it. The authors in (Alotaibi & Elrefaei, 2015) state that for steganography to execute its intended purpose, there are certain techniques that need to be implemented. At least one or two of the following techniques under steganography may be used: spread spectrum techniques, transform domain techniques, and substitution and insertion techniques. However, they describe substitution, injection and propagation

as the most relevant techniques under steganography. The substitution exchanges small parts of the carrier file for the hidden message to stay undetected by the intruder. Injection basically avoids suspicion when the file is added to the cover media, whilst propagation distinguishes the use of cover objects but uses the generation engine given by unseen data to mimic a file (being it music, sound, or text).

The authors in (Kyei, Panford, Hayfron-acquah, Student, & Lecturers, 2019) employ optimum steganography and cryptography. The Least Significant Bit (LSB) algorithm is applied to insert or embed messages into a cover object. The cryptography used in their study employs the use of the asymmetric cryptography known as the RSA. The combination of both the LSB insertion technique and the RSA technique in their system makes it one of the best applications to ensure data security and secrecy on android smart devices. Steganography is a technique used for hiding secret data in an object known as an image (M & Hussain, 2014). Generally, the intensity of pixels is a method applied in concealing data within image steganography. Hence, images are found to be the most popular and commonly used cover objects in steganography.

The authors in (Apau, B., & Twum, 2016) aimed towards ensuring the same file size output after it is embedded, and reducing the file size to be embedded remarkably. The first aim is obtained by re-encoding and reconstructing the cover video through video encoding techniques. Next, LSB is deployed in embedding the file within a converted frame. The RSA and Huffman code compressing algorithms are used to reach a high capacity of payload. The analytic results indicate that the embedding of files within cover videos keeps the characteristics of both the original and the stegano video the same. As for the compressing levels, they were found to be above average, namely more than 20% percent.

III. CONFORMAL MAPPING IN IMAGE PROCESSING

Conformal maps are functions that preserve both the angle and shape of an infinitesimal figure, but not necessarily its size (as equation 1) (Frederick & Schwartz, 1990).

$$w = f(z) \quad (1)$$

To state it in a more specific way, a map is considered to be conformal (or angle-preserving) at z_0 whenever both the oriented angles between curves through z_0 , and their orientation or direction are preserved. To exemplify, an important family of conformal maps results from complex analysis.

$$f: U \rightarrow C \quad (2)$$

In case U (as in equation 2) is an open subset of the complex plane, then the function derivative is with every non-zero on U . Whenever f is anti-holomorphic, which means that it represents the conjugate to a holomorphic function, then the angles remain preserved but their orientation is reversed. The Riemann mapping theorem mentions that all conformal non-empty open proper subsets of C with a simple connection admit bijective conformal maps to the open unit disk around P in complex plane C . P here represents a given point within the plane. The open unit disk is the collection of points that have a distance to P less than 1. The implication is made that if U is an open subset with a simple connection in complex plane C (which is not all of C), this means that a bijective or one-to-one mapping (f) exists from U to the open unit disk D (Sharon & Mumford, 2006) .

$$f: U \rightarrow D \quad (3)$$

where

$$D = \{z \in C : |z| < 1\}$$

As (in equation 3) f is a bijective map it is conformal.

The map of an extended complex plane (equaling a sphere) is considered to be

conformal if, and only if, it is a Mobius transformation, in other words, a transformation that leads to a rational function of the form (Ganguli, 2004).

$$f(z) = \frac{az+b}{cz+d} \quad (4)$$

Regarding the conjugate, it preserved the angles but it reverses the orientations of them. Figure 1 shows the conformal mapping graph.

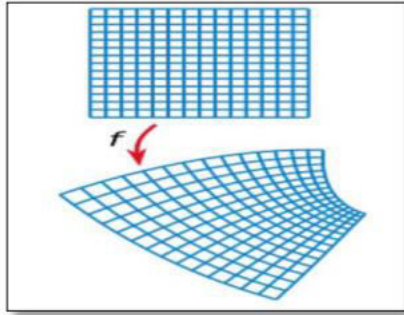


Figure 1 Conformal mapping graph

The transformations applied in relating different fish to each other may vary in terms of complexity. Figure (2) shows examples of transformation that illustrate the fish's shapes.

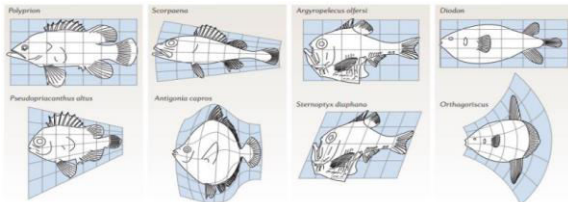


Figure 2. Transformation examples, These images are taken from (Arthur, 2006)

RSA Cryptosystem

R. Rivest, A. Shamir, and L. Adleman (1978) published an academic article in which a public-key crypto-system is proposed which includes key generation and public-key ciphers. Their security depends on the presumed complexity of factoring integers into their prime factors. This RSA crypto system, named after the acronym of the authors' surnames, has proved its efficiency until present day. Its cryptographic applications include banking, e-mail security, and online e-commerce. Besides its practical applications, it is most commonly used for encrypting small pieces of data, particularly for key transport, and

digital signatures and certificates on the internet (Deen, El-Badawy, & Gobran, 2014). The RSA algorithm can be described as a system of asymmetric cryptography whereby two sets of keys are used in encrypting and decrypting messages. This is necessary for ensuring the security of quality information. The generating of such keys requires processes of complicated mathematical computations, resulting in the public and private keys. The first key is given to the sender of the message so that the messages are encrypted, whereas the private key is given to the receiver so that they can decrypt the encrypted messages secretly (Apau & Adomako, 2017). The RSA algorithm is determined by a finite (Galois) field exponentiation over integers modulo a prime, and it makes use of larger integers that consist of 1024 bits. (Oleiwi, Alawsi, Alisawi, Alfoudi, & Alfarhani, 2020).

Steganography

Steganography can be described as the act of hiding data to be transmitted via apparent innocuous carriers so as to conceal the presence of data. Even though steganography is found to be an old craft, it has been revived with the emergence of computer technology (Kamble, Engineering, Gaikwad, & Engineering, 2013). Computer-based steganographic techniques presented the initial introduction of covering data digitally by embedding information in such a way that it is unknown to the original cover (Khalid Obayes, 2013). This kind of information can be shared through communication as text, or a binary file, and extra information about the cover or the owner could also be provided, like digital watermarks or fingerprints. The basis of steganography involves the concept of artifacts like bitmaps and audio files containing redundant information. Whenever messages are hidden using steganography a remarkable reduction is observed in the possibility that the message might be detected. The encryption of the messages also adds an extra layer of protection, as the message needs to be decrypted

in case it is discovered. Steganography can be viewed as akin to cryptography (Kamble et al., 2013), as both are used as means for adding elements of secrecy to the communication. Cryptography techniques tend to scramble messages so that only authorized persons can understand it when intercepted (Amin, Salleh, Ibrahim, Katmin, & Shamsuddin, 2003).

Throughout the last years, various steganography techniques are proposed for embedding hidden messages within multi-media objects such as images. As for the latter, there are several ways through which secret messages or information are hidden so that no alternations made to the original images can be discerned. Some of the most popular techniques involve:

- Least Significant Bit insertion (LSB).
- Masking and filtering.
- Transform techniques.

Least Significant Bits (LSB) insertion represents a simplified approach to embed information within image files, as the message bits are directly embedded into the least significant bit plane of the cover image, following a deterministic sequence. No human-perceptible differences result from the modulation of the least significant bits, as the change amplitude is relatively small. There are several benefits related to the use of LSB, including its simple embedding of message bits within the LSB planes of cover-image, and the comparatively smaller amplitude of change to the bits which makes the resulting stego-image seem identical to the cover-image, allowing a higher perceptual transparency (Khalid Obayes, 2013).

Compression

Data compression is categorized in two types: lossy compression and lossless compression. In most of the cases the loss of information is not tolerated due to the precious nature of data (Nitu, Kumar, & Rishi, 2019). Examples include the field of medical imaging, fingerprint data, and computer programs. In addition, lossless data compression is more desirable in case of

text. Therefore, the data must be compressed in such a way that one hundred percent of the data is extracted after decoding the compressed data. Hence, lossless data compression is more preferred over lossy compression. Since the data is decoded one hundred percent as of the original data, it is also referred to as reversible compression (Shah & Sethi, 2019). In the proposed method, the GZIP was used to perform the last step, which is to compress the cover image after the secret image was hidden inside it (the stego image).

There are two types of GZIP. To compress real time data, GZIP makes use of a combined LZ77 and static Huffman encoder. Meanwhile, to compress non real time data, GZIP makes use of a combination of LZ77 and dynamic Huffman algorithm. The Static Huffman encodes the data in one go by assigning every symbol a fix length of code. Static Huffman encoder does not bother to identify the frequency distribution of data. Hence, the data is encoded efficiently with a compromise on compression ratio (Oswal, Singh, & Kumari, 2016). A kind of moderate task is done by the dynamic Huffman encoder to encode the symbols (Patel, Katiyar, & Arora, 2016). It encodes the symbols with variable length codes in such a way that the symbol which occurs the most often gets lesser length codes, and the symbol which has a low density is encoded with comparatively lengthy codes. As a result, a sufficient compression ratio is achieved (Shah & Sethi, 2019) (Balakrishnan & Sahoo, 2006).

Peak Signal to Noise Ratio (PSNR)

PSNR is a standard metric used in steganography method in order to find the quality of the stego images. The higher the value of PSNR, the more quality the stego image will have. If the cover image is C of size $M \times M$ and the stego image is S of size $N \times N$, then each cover image C and stego image S will have pixel value (x, y) from 0 to M-1 and 0 to N-1

respectively. The PSNR is then calculated as follows (Khalid Obayes, 2013):

$$PSNR = 10 \log_{10} \left[\frac{MAX^2}{MSE} \right] \quad (5)$$

Where

$$MSE = \frac{1}{NM} \sum_{x=1}^{N-1} \sum_{y=1}^{M-1} (c(x,y) - s(x,y))^2 \quad (6)$$

IV. PROPOSED METHODOLOGY

In this paper, a mechanism for transmitting an encrypted image has been proposed to obtain high security in transmission. Figure (3) shows the overall outline of the process of sending and receiving the image, whereby the processing was done in the two stages of sending and receiving, as shown below.

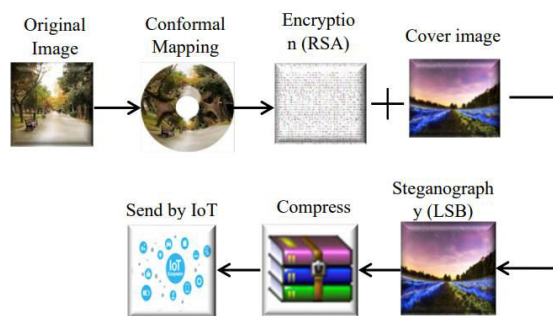


Figure 3 Sending phase in IoT

Sending phase

The transmission process goes through a number of steps, which are: conformal mapping, encryption, steganography, and image compression.

1. Conformal Mapping

In this step, the initial image processing was performed to obtain a new image with the same characteristics as the original image, but with a different appearance. Figure (4) shows the steps for converting the original image into a processed image using the conformal mapping algorithm.

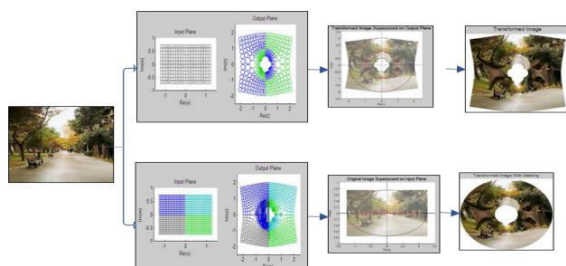


Figure 4 Converting the original image according to geometric models

2. Image Encryption

The RSA algorithm was used to encrypt the image. It depends on the use of two keys with prime numbers symbolized by (p, q) and the calculation of the value for each of (E, N, M) mathematically. Figure (5) shows the value of p = 11, q = 13, E=7, we also specify the image to be encrypted and the place to save the resulting image from the encoding process. The input is a jpg image, with an image resolution of (512×502). The execution time was only (12s), and the output of the encryption process is a text file for ease of concealment as well as for the ease of transferring the image within the IoT environment.



Figure 5 Image encryption by (RSA)

3. Steganography

The LSB algorithm is used to hide the text file generated from the previous step. The image type (bmp) is used as the cover of the text file. Figure (6) shows the cover image properties. In this step, the issue of encrypted image size is solved, as the text took a smaller size than the cover image, which in turn facilitates the process of hiding. The size of the resulting images are also preserved and sent to the compression algorithm.

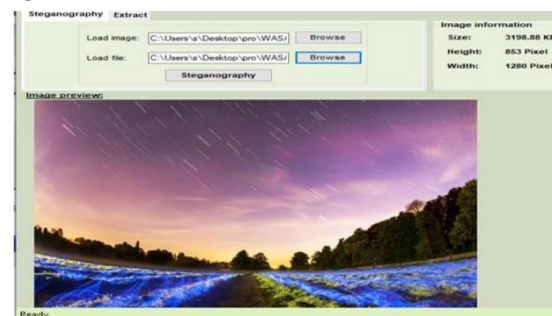


Figure 6 Steganography step

4. Compress Image

The GZIP algorithm is one of the important compression algorithms adopted as a tool for compressing files on websites. This algorithm is used for the image resulting from the hiding step.

Received Phase

The received image after compression is sent over the Internet as the first step in the receiving process. Their sequence is illustrated in Figure (7). The following is an explanation of the receiving process:

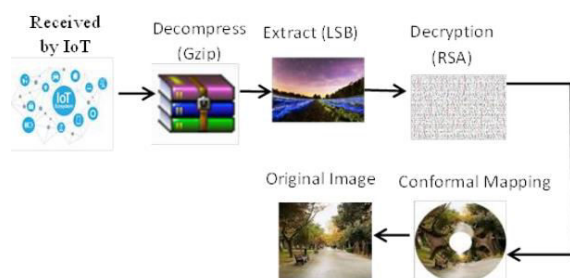


Figure 7 Receiving phase

1. Decompress

The GZIP function was used to decompress the received image. This process produces the cover image. It is considered to be important as it may fool the intruder by not being able to identify the hidden file inside the cover image.

2. Extraction

In the Steganography step, the LSB algorithm was used to hide the original image in the cover image. In the process of extracting the original image from the cover image, we also used the same algorithm, the input for this step is a BMP image and the output is a text file of type (txt).

3. Decryption (RSA)

In this step, the TXT text file is received and decrypted by the RSA algorithm. First, the values of D, N are entered, which are calculated based on the values of P, Q as inserted during the transmission process (encryption step). The decoding process results in a jpg type image. This step is the basic step in accessing the image to be secured. Figure (8) shows the decrypting execution step.

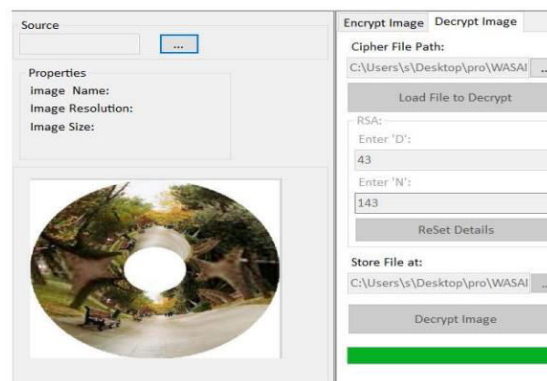


Figure 8 Decoding the text file to obtain the encrypted image type jpg

V. EXPERIMENTAL RESULT

To test the proposed method, four images of different sizes were used, three of which are jpg type and one is bmp type. Table (1) shows the basic characteristics of these images, such as their dimensions, type, and size. Picture 4 differs as the initial processing was not performed on it (conformal mapping). The rest of the images underwent the initial processing of conformal mapping, which in turn changed the appearance of the image (white spaces are added). These changes are not exploited in the encryption process, as this effect is clearly observed at the time of implementation. The encoding of the images is the least possible in the fourth image (4s).

As for the size of the image resulting from combining the encrypted image with the cover image in the process of concealment, it was the least for image 4, which is 136 KB. In the process of compression, image 3 presented the best performance despite the convergence of the results. It is worth noting that the parameter values of (PSNR and MSE) were represented by 8-bit pixels for each sample. The results execution appears to be perfect, as shown in Table (1).

Table 1 The table shows the main characteristics of the images with the implementation results of the proposed method

	Image Characteristics			Result execute				
	Dimension	Image Type	Size	Encryption time	Steganography	Size after compression	MSE	PSNR
Image1	502×512	BMP	3.12 MB	12s	Type: .txt Size: 436 KB	2.73 MB	9.6	38.59
Image2	612×516	PNG	208 MB	6s	Type: .txt Size: 219 KB	2.76 MB	3.29	42.99
Image3	616×512	JPG	40.3 MB	14s	Type: .txt Size: 494 KB	2.68 MB	11.32	37.62
Image4	225×225	JPG	12.3 KB	4s	Type: .txt Size: 137 KB	2.75 MB	2.75	43.77
Cover Image	1280×853	BMP	3.12 MB	-	-	-	-	-

VI. CONCLUSION

The proposed optimized image encryption methodology effectively addresses the dual challenge of maintaining security and efficiency in IoT environments. By combining lightweight cryptographic operations with advanced scrambling and diffusion techniques, the approach ensures secure image transmission without imposing excessive computational burdens on constrained IoT devices.

Performance evaluations confirm that the method offers strong resistance to brute-force, statistical, and differential attacks, while preserving image quality metrics and processing speed suitable for real-time applications. Its adaptability across various IoT scenarios—ranging from home security systems to telemedicine—makes it a viable solution for enhancing data privacy and trust in interconnected ecosystems.

In conclusion, this research contributes to the development of secure, scalable, and IoT-friendly image encryption solutions, paving the way for safer digital communication in the growing landscape of intelligent devices.

REFERENCES

1. Alotaibi, R. A., & Elrefaei, L. A. (2015). Steganography in Arabic Text Using Zero width and Kashidha Letters. *International Journal of Computer Science & Information Technology (IJCSIT)*, 6(4), 1–11.
2. Amin, M.M., Salleh, M., Ibrahim, S., Katmin, M.R., & Shamsuddin, M.Z.I. (2003). Information hiding using steganography. 4th National Conference on Telecommunication Technology, NCTT 2003 - Proceedings, (June 2015),

21–25.

3. Apau, R., & Adomako, C. (2017). Design of Image Steganography based on RSA Algorithm and LSB Insertion for Android Smartphones. *International Journal of Computer Applications*, 164(1), 13–22. <https://doi.org/10.1109/NCTT.2003.1188294>
4. Apau, R., Hayfron-Acquah, J.B., & Twum, F. A Modified High Capacity Video Steganography Technique Based On Spatial Domain Method, Asymmetric Cryptography and Huffman Code Algorithms. *Communications*, 5(10), 53-60. <https://doi.org/10.5120/ijca2017913557>
5. Arthur, W. (2006). Series on Historical Profiles — TIME LINED ' Arcy Thompson and the theory of transformations. *Genetics*, 7(1958), 7–12.
6. Balakrishnan, R., & Sahoo, R. K. (2006). Lossless compression for large scale cluster logs. 20th International Parallel and Distributed Processing Symposium, IPDPS.
7. Balakrishnan, R., & Sahoo, R.K. (2006). Lossless compression for large scale cluster logs. In *Proceedings 20th IEEE International Parallel & Distributed Processing Symposium*. <https://doi.org/10.1109/IPDPS.2006.1639692>
8. Deen, A.E.T. El, El-Badawy, E.S.A., & Gobran, S.N. (2014). Digital Image Encryption Based on RSA Algorithm. *IOSR Journal of Electronics and Communication Engineering*, 9(1), 69–73. <https://doi.org/10.9790/2834-09146973>
9. Dengre, A.R., Gawande, A.D., & Deshmukh, P.A.B. (2013). Effect of

- Audio Steganography based on LSB insertion with Image Watermarking using AVI video Cryptography : International Journal of Application or Innovation in Engineering & Management (IJAIEEM), 2(6), 363–370.
10. Frederick, C., & Schwartz, E.L. (1990). Conformal Image Warping. IEEE Computer Graphics and Applications, 10(2), 54–61.
<https://doi.org/10.1109/38.50673>
11. Ganguli, S. (2004). Conformal Mapping and its Applications. IEEE Transaction on Medical Imaging, 23(8), 1–4.
<https://pdfs.semanticscholar.org/1bed/474a4649f8344f1c56ee0972593e816ca01b.pdf>
12. Jemai, A., Sadek, F., Salim, M., & Talbi, M. (2017). A lightweight Encryption Algorithm applied to a quantized speech image for Secure IoT. Proc. of the Sixth International Conference on Advances in Computing, Electronics and Communication - ACEC 2017, (February), 1–6.
<https://doi.org/10.15224/978-1-63248-138-2-01>
13. Kamble, P., Engineering, S., Gaikwad, V.S., & Engineering, S. (2013). Steganography Techniques: A Review. International Journal of Engineering Research & Technology (IJERT), 2(October).
14. Khalid Obayes, H. (2013). Suggested Approach to Embedded Playfair Cipher Message in Digital Image. 3(5), 710–714.
15. Kharchenko, V., Kolisnyk, M., Piskachova, I., & Bardis, N. (2017). Reliability and Security Issues for IoT-based Smart Business Center: Architecture and Markov Model. 2016 Third International Conference on Mathematics and Computers in Sciences and in Industry (MCSI), (June 2020), 313–318.
<https://doi.org/10.1109/mcsi.2016.064>