Analysis of Watermarking Cryptography Algorithm For Secure Transmission of Images

Project work submitted to Avinashilingam Institute for Home Science and Higher Education for Women

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY

SUBMITTED BY

VIJAYALAKSHMI K

21PIT012

Under the Guidance of

Dr. Mrs.A.Sumi M.C.A., M.Phil. Ph.D..

Assistant Professor

Department of Information Technology

AVINASHILINGAM INSTITUTE FOR HOME SCIENCE AND HIGHER EDUCATION FOR WOMEN

SCHOOL OF PHYSICAL SCIENCES AND COMPUTATIONAL SCIENCES

DEPARTMENT OF INFORMATION TECHNOLOGY

COIMBATORE-641043

MAY 2023

DECLARATION

DECLARATION

I hereby declare that the project entitled "**Analysis of Watermarking Cryptography Algorithm For Secure Transmission of Images"** is a record of the original work done by **K.Vijayalakshmi (21PIT012)** under the guidance of **Dr.Mrs.A.Sumi M.C.A., M.Phil., Ph.D.** Assistant Professor, Department of Information Technology, School of Physical Sciences and Computational Sciences**,** Avinashilingam Institute for Home Science and Higher Education for Women in the partial fulfilment for the award of the degree of Master of Science in Information Technology, and this project work has not formed the basis for any Degree/Diploma/Associates.

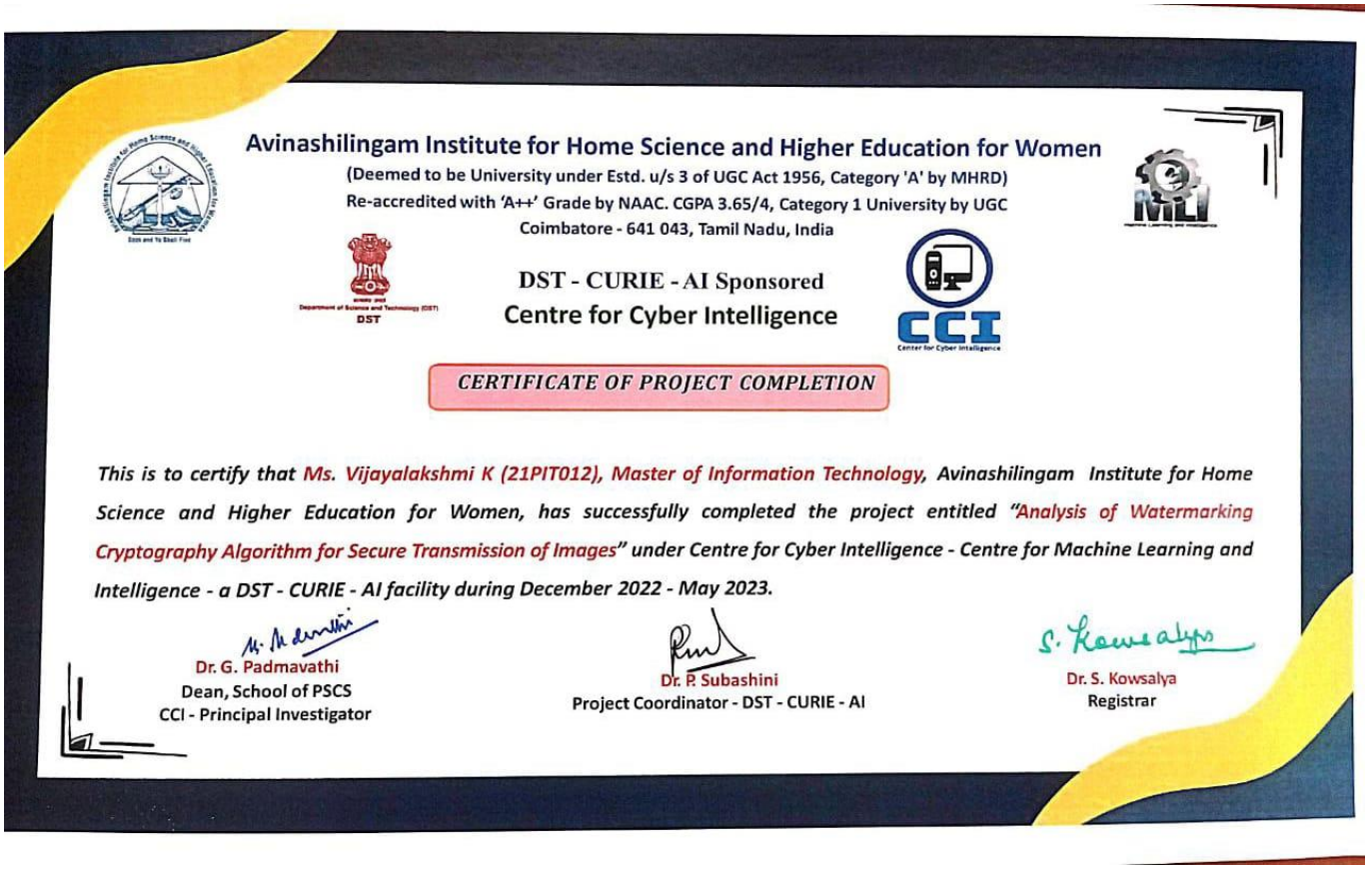
Place: Coimbatore
Date: 19/05/2023

Signature of the Candidate

Countersigned by,

Dr. Mrs. A.Sumi M.C.A., M.Phil. Ph. D
Assistant Professor,
Department of Information Technology,
School of Physical Sciences and Computational Sciences.

CERTIFICATE



Avinashilingam Institute for Home Science and Higher Education for Women

(Deemed to be University under Estd. u/s 3 of UGC Act 1956, Category 'A' by MHRD)

Re-accredited with 'A++' Grade by NAAC. CGPA 3.65/4, Category 1 University by UGC

Coimbatore - 641 043, Tamil Nadu, India

DST - CURIE - AI Sponsored

Centre for Cyber Intelligence

CERTIFICATE OF PROJECT COMPLETION

This is to certify that Ms. Vijayalakshmi K (21PIT012), Master of Information Technology, Avinashilingam Institute for Home Science and Higher Education for Women, has successfully completed the project entitled "Analysis of Watermarking Cryptography Algorithm for Secure Transmission of Images" under Centre for Cyber Intelligence - Centre for Machine Learning and Intelligence - a DST - CURIE - AI facility during December 2022 - May 2023.

Dr. G. Padmavathi
Dean, School of PSCS
CCI - Principal Investigator

Dr. P. Subashini
Project Coordinator - DST - CURIE - AI

Dr. S. Kowsalya
Registrar

CERTIFICATE

This is to certify that this project work entitled "**Analysis of Watermarking Cryptography Algorithm For Secure Transmission of Images"** done by K.Vijayalakshmi(21PIT012) has been submitted to Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore-43 in partial fulfilment of the requirement for the award of the degree of **MASTER OF SCIENCE IN INFORMATION TECHNOLOGY.** This Project has not found the basis for the award of any Degree/Associate/fellowship or similar title to any Candidate of any University. Certified as a Bonafede record of the work submitted for the Viva-voce held on_____.

Signature of the Head of the Department

Signature of the Supervisor

Signature of Examiner

ACKNOWLEDGMENT

ACKNOWLEDGEMENT

I sincerely thank the **Lord Almighty** and **My lovable parents** for showering their generous blessings upon me in all endeavors.

I wish to express my gratitude to **Prof.S.P.Thyagarajan,** Chancellor, AvinashilingamInstitute for Home Science and Higher Education for Women, Coimbatore, for providing thefacilities to conduct this study.

I extend my thanks to **Dr. Bharathi Harishankar, Ph.D., FRSA.,** Vice Chancellor, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, for providing flamboyant help towards the completion of the study.

I record my deep sense of gratitude and indebtedness to **Dr. S. Kowsalya,** M.Sc., M.Phil., Ph.D., Registrar, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, for providing adequate help for the study

I grateful record my sincere thanks to **Dr. G. Padmavathi** M.Sc., M.Phil., Ph.D., Dean and Professor, School of Physical Sciences & Computational of Sciences, Avi nashilingam Institute for Home Science and Higher Education for Women, Coimbatore, for timely help rendered throughout the course of this work.

I heartily Thanks to **Dr. Mrs. D. Shanmugapriya** M.Sc., M.Phil., Ph.D., SET Head of Department of Information Technology for the valuable guidance and encouragement during our project.

I heartily Thank my esteemed project guide **Dr. Mrs. A. Sumi M.C.A., M.Phil., Ph. D., Assistant Professor** Department of Information Technology, for imparting tremendous assistance and well-timed support for triumph of our project.

I like to extend my gratitude to Ms. A. Roshini, Technical Assistant –Center of cyber intelligence, Department of Computer Science, For providing Project guidelines and always

supported me and encouraged me with valuable advice and Profound belief in my work and abilities.

I express my honorable thanks to our project coordinator Department of Information Technology, for imparting tremendous assistanceand well-timed support for triumph of our project.

I sincerely thank all **the staff members** of Department of Information Technology Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, for their help and support.

I would like to express my special thanks to **my parents, my friends** and all **my well-wishers** for their constant encouragement, support and help in carrying out this work successfully.

I would like to acknowledge the help rendered by Center for Cyber Intelligence, DST -CURIE – AI Sponsored Phase II for providing the laboratory facilities to execute my project.

ABSTRACT

ABSTRACT

Today, a wide range of applications employ digital photographs. They might be vital programmes like those used in the military or the medical field or social networking apps. Nevertheless, cyberattacks or privacy issues contribute significantly to major security issues for digital photos. Therefore, a variety of strategies are suggested to guarantee security and privacy.The privacy of data is ensured through encryption techniques. This research examines the elliptic curve cryptography-based encryption technique for digital picture encryption. The factors and security threats discussed in this evaluation determine how well digital picture encryption works. Additionally, reviews of previous research are done.

Aims: To propose a A Hybrid Encryption Scheme By Using modified The Ecdh (Elliptic Curve Diffie–Hellman). This project deals with the cryptography way of Image encryption. This is required to keep sensitive information safe from the undesirable users. In this work an AES (Advanced Encryption Standard) scheme along with water marking filters are used for encryption, decryption process. Peak Signal-to-Noise Ratio (PSNR) and Mean Squared Error are calculated. **Method:** The encryption and decryption of images using this method consists of five phases. Phase 1is A Hybrid Encryption Scheme by Using modified The Ecdh (Elliptic Curve Diffie–Hellman). In Phase 2 is the Key Exchange Scheme to Derive a Shared Secret Key For Symmetric Data Encryption And Decryption. In Phase 3, three different Watermarking Techniques are used for getting a better db. values of image. In Phase 4, we calculate the Peak Signal-to-Noise Ratio (PSNR) and Mean Squared Error. The output of different algorithms is evaluated in phase 5 withperformance measures. It is observedthat some models give better accuracy than others, and the entire project is developed on the Python platform. **Results:** From this proposed system, the best accuracy of Encryption and decryption was proposed and the result was combared with previous methods.

Keywords: **Encryption, Decryption, Cryptography, Watermarking**

TABLE OF CONTENT

CHAPTER 1
INTRODUCTION

1.1 About the Project

Nowadays, digital images and videos have high importance because they have become the main carriers of information. Lately, it has been a must for all organizations to get involved more on the internet for multiple reasons to stay competitive and active in the market. Large organizations need to make sure that any image they put on the internet is kept safe and away from their competitors. The motivation behind this seminar topic is that image encryption and especially AES encryption is a safe solution for this issue, trying as possible to maximize the security of important data, images and videos over the internet. AES scheme was applied on colored digital images to keep its content safe of the internet. AES was used as it is known to be a highly secured method of encryption.

Billions of digital images per second are shared on many platforms, especially on messaging applications and social media. However, there are many security threats such as alteration of digital images, illegal acquisition and distribution, theft or disclosure of personal data [1]. New methods are being tried to minimize these damages caused by malicious third parties. Encryption methods are widely used to ensure the security of digital images. Moreover, there are hiding methods such as steganography and watermarking for digital images [1], [2]. In the steganography method, the digital image can also be hidden in data, message or even a different image. Similarly, the watermark method is widely used to ensure the security of the digital image. It is mainly used in money against forgery and counterfeiting [2]. In the encryption method, the digital image is completely hidden by encrypting the image . These methods can be used for different purposes. Digital image encryption is a method of converting to an encrypted image by applying a transform function to the pixels of the original image. After this encryption, the original image should be completely unrecognizable and not easily restored. The success of a digital image encryption method depends on being useful, fast and secure enough to be used in daily life applications. Image encryption is used in many applications today. These applications include medical [3], internet and communication media [4], telemedicine [5], multimedia systems [6], personal photos, military communication, web browsing [6], mobile network, video conferencing, IoT, cloud computing, connected car and unmanned aerial vehicles. The reason why it is used in such common applications is undoubtedly that the concepts of security and privacy are important

The main purpose in encryption is confidentiality, integrity, authentication, non repudiation [7], [8]. These criteria are taken into account in the selection of the encryption method. It is possible to classify many different encryption methods. Encryption methods can be divided into three symmetric, asymmetric and hashing [6]. In symmetric methods, there is only one private key, and the sender and receiver use the same key to encrypt and decrypt it [9]. AES, DES [10], Hill cipher [4] method can be given as examples. While this method has the advantage of being simple and fast, this key sharing is quite difficult. Key exchange between two or a small number of people is partially easy, but key exchange with thousands of people is very difficult and inefficient.

Asymmetric encryption, or public-key encryption method, has two keys, the public key, and the private key [11]. Key exchanges are performed using mathematical methods. For example, the Diffie-Helman key exchange method is the first to find a solution to the key exchange problem. Public key encryption methods such as RSA, ECC can be given as examples [12] .The hash function is a one-way mathematical method. This function, which is used for many different purposes, is especially popular in digital forensic. By taking the hash values of the digital images, it can be checked that no change has been made on the image afterwards. If any change has been made to the image, the calculated hash values will not be the same. The point to be considered here is the secure hash function selection, otherwise unwanted collisions may occur in hash functions. In this case, the results obtained will not be confidential.

Digital image encryption can be done in many different methods such as encryption system (ECC, RSA, AES, DES etc.), chaos system, Arnold transform [13], 3D chaotic map [13], 4D cat map, Lorenz system [4]. This paper describes ECC-based image encryption. Methods such as Koblitz, ElGamal, Menezes-Vanstone can be used in ECC-based digital image encryption.

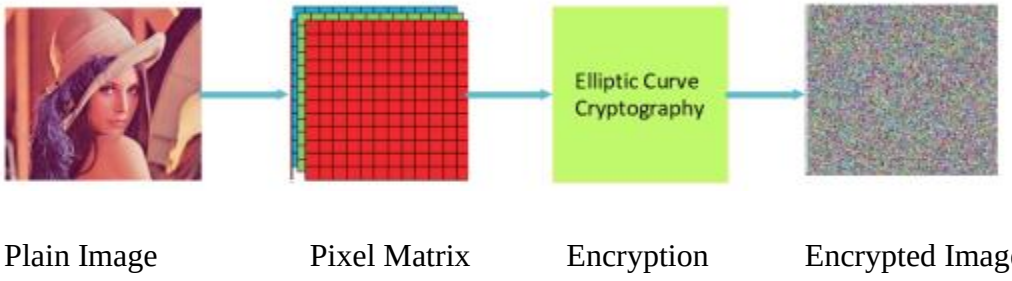
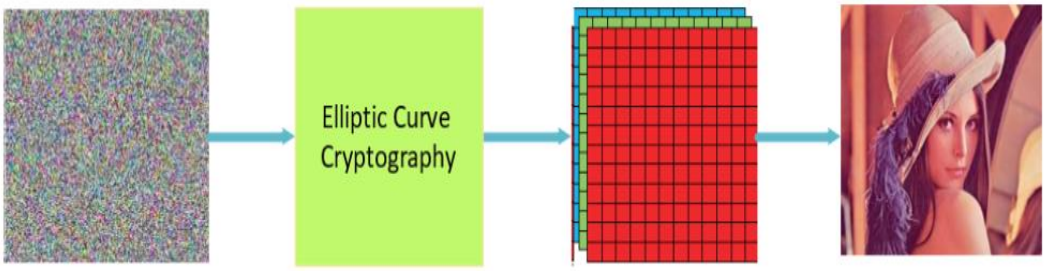


Fig.1. Digital image encryption

Figure 1 shows the steps of encryption of a digital image. First, the plain image is converted to matrices containing its pixels. It is then encrypted with ECC. Finally, the encrypted image is obtained. Digital image encryption is quite different from text encryption. Fewer parameters are used when evaluating the security of text encryption. Parameters such as Entropy, PSNR, UACI are used for performance evaluation in image encryption methods.



Encrypted Image Decryption Pixel Matrix Plain image

Fig.2. Digital image decryption

Figure 2 The process of image decryption is to flip the encryption process. The cipher images and the keys are input to the decryption system, and the chaotic sequences and chaotic matrices are generated by the chaotic system for the inverse diffusion and inverse permutation of the decryption. All cipher images are fused into a cipher cube, and the exclusive OR operation is performed in the order of points, rows, columns, and planes. The inverse diffused cube is reverse permuted from right to left along the columns and from bottom to top along the rows. The decrypted images are obtained by re-partitioning the cube after the inverse permutation and reorganizing them according to the sizes and types of the original images

1.1.1 Digital Images

Nowadays, we cannot think about a world without images. As people go through multiple images m every day through different resources such as TV, magazines and the internet. According to the authors of [1] digital image is an image but using pixels. Pixels is a finite number of digital values that is used as a unit of measurement of the digital image in which any computer can understand. Digital image is basically a type of image which is formed from pixels. Each pixel has some finite size and is represented by some finite intensity to show the image and they are arranged in a rectangular way. The are two types of Digital images which are Raster digital image and Vector digital image.

1.1.2 Digital Image Formats

Digital Image which is formed from pixels. Each pixel has some finite size and is represented by some finite intensity to show the image and they are arranged in a rectangular way. The are two types of Digital images which are Raster digital image and Vector digital image.

Raster Digital Image

When an asymmetric routing option is available, attackers will frequently use various routes to gain access to the targeted device or network. This makes it possible for them to escape being noticed by havinga significant number of suspicious packets skip over specific network segments and any applicable network intrusion system

JPEG (Joint Photographic Experts Group)

GIF (Graphics Interchange Format)

TIFF (Tagged Image File Format)

Vector Digital Image

Vector digital image consists of lines and curves which are called paths. The vector image represents the wireframe type of image. It is the type of image used when the data requires handling such as,

AFD (Affinity Designer document)

EPS (Encapsulated PostScript)

SVG (Scalable Vector Graphics)

2D Image Representation

An image is and artifact that records visual perception of data. The are 4 types of digital image representation which are Pixel-based representation, Block-based representation, Region-based representations and Hierarchical representations.

As the name implies, Trojan Horse infections build backdoors into networks, granting attackers quick access to systems and any stored data. Trojans do not self-replicate or spread by contaminatingother files, in contrast to other viruses and worms. Trojans are frequently created through peer-to- peer file exchanges and can be downloaded through web archives and file repositories.

Pixel-Based Representation

It is the most simple representation for image definition. The pixel is the physical point in the raster image. Raster image is the types of images that are produced when scanning or photographing an object. In this representation there are relations between elements. Only local information for each element is stored in each pixel. The representation usually has a large number of elements and is used to depict the image

Block-Based Representation

Block-based representation the image is made of a set of rectangular array size. Compared to pixel based representation, the number of elements is slightly less. Like the pixel-based representation local information is stored. This representation method can be done for both gray-scale and binary images and also, it uses compression, segmentation and extracting different image features.

R, G and B components are shown in a 7x6 matrix for a selected point anywhere in a color image.

1.1.3 Cryptography

Cryptography comes from the Greek terms cryptos, which means ”hidden,” and grafein, which means ”to write.” Cryptography has been used to hide communications inside traditional modes of communication that could otherwise be intercepted throughout history. This is accomplished by keeping the contents of the message hidden from everyone except those with the key to unlock it, Cryptography is usually divided into 2 segments, classic and modern cryptography. Classic cryptography is an algorithm that uses a secret key to secure data which is usually very easy to break as it has been relatively outdated in comparison with modern techniques.

Caesar cipher substitution is considered one of the oldest classic algorithms. Caesar cipher is simply a type of substitution procedure in which each letter in the original text (plain text) is shifted a particular number of positions down the alphabet's series A shift of the 3 would, for example, mea1.

1.1.4 Elliptic Curve Cryptography (ECC)

ECC is a public key-based encryption method put forward by Ko blitz and Miller in 1985. The reason why it is so popular today is the security it provides and the short key size. Compared to RSA, a very low key size is needed for the same level of security. The mathematical operations in ECC are performed on finite fields.

Basic Terms Utilized in Encryption

(i) Plain image: it is the image that needs security while there is transmission over the public network. It is also known as the original or input image.

(ii) Cipher image or encrypted image: the plain image converted into a non readable form after encryption is called a cipher image.

(iii) Encryption: it is the process of converting a plain image into a cipher image utilizing an encryption approach and a secret key.

(iv) Decryption: at the receiver side, the cipher image is converted into a plain image utilizing a decryption approach and a secret key. *Is process is known as decryption.

1.2. Motivation and Justification

Online attacks are increasing daily in contrast to these changes. Essential techniques are used toidentify the attacks in order to defend information security against them: recognition based on the signature and detection based on anomalies. The more challenging method of the two is anomaly-based detection. This effort aims to build a system that combines layered ensemble learning methods to precisely and quickly recognise various attacks.

1.3 Problem Statement

To define the system for transferring the image secured from one person to another is the main drawbacks which all the algorithms fails at one stage. We formulated those algorithms into single path of securing the system

1.4 Objective

The Objective of the project is to send the personal images end to end encrypted with secured key.

CHAPTER 2

SYSTEM SPECIFICATION

2.1 Hardware Requirement

PROCESSOR Intel I5

ABOVERAM 8 GB

HARD DISK CAPACITY 1TB

2.2 Software Requirement

OPERATING SYSTEM Windows10 Pro or higher configurated with Virtual machine (Ubuntu / Kali Linux)

FRONT END Python

2.3 About the Software

The tools used in this project are listed below.

- Python 3.11.3
- Terminal

2.3.1PEP 654 Exception Groups and except

PEP 654 introduces language features that enable a program to raise and handle multiple unrelated exceptions simultaneously. The built-in types ExceptionGroup and BaseExceptionGroup make it possible to group exceptions and raise them together, and the new except* syntax generalizes except to match subgroups of exception groups.

2.3.2 Python Package

Scikit-Learn

Scikit-learn, a free Python package that is frequently seen as a direct extension of SciPy, is based on NumPy and SciPy. It is especially made for creating supervised and unsupervised machine learning algorithms and data modelling.

Scikit-learn is user-friendly and beginner-friendly because of its straightforward, intuitive, and consistent interface. Scikit-learn performs admirably by enabling users to alter and exchange data as theyneed, despite the fact that its utility is constrained because it only excels at data modelling.

Pandas

Python's Pandas package for data research and analysis enables programmers to create simple, seamlesshigh-level data structures. Pandas, which is based on NumPy, is in charge of getting data sets and data points ready for machine learning. Pandas uses one-dimensional (series) and two-dimensional (Data Frame) data structures. These two types of data structures allow Pandas to be used in a range of industries,from science and statistics to banking and engineering.

Due to its adaptability, the Pandas library can be used with other scientific and numerical libraries. Because they are rapid, compliant, and highly descriptive, their data structures are simple to use. By aggregating, integrating, and re-indexing data with Pandas, one can modify data functionality with a minimum of keystrokes.

Matplotlib

Matplotlib is a data visualization library that is used for making plots and graphs. It is an extension of SciPy and is able to handle NumPy data structures as well as complex data models madeby Pandas. Although it is expertise is limited to 2D plotting, Matplotlib can produce high-quality andpublish-ready diagrams, graphs, plots, histograms, error charts, scatter plots and bar charts.

Matplotlib is intuitive and easy to use, making it a great choice for beginners. It is even easier to use forpeople with pre-existing knowledge in various other graph-plotting tools. It offers GUI toolkit support, including wxPython, Tkinter, and Qt.

NumPy

Open-source and well-known Python library for numbers, NumPy. It is capable of carrying out a wide range of mathematical operations on matrices and arrays. One of the most popular libraries for scientific computing, it is frequently used by scientists to analyse data. It is perfect for machine learning and artificial intelligence (AI) projects since it can process multidimensional arrays, handle linear algebra, and perform Fourier transformation.

NumPy arrays demand a considerable reduction in storage space when compared to standard Python lists.They are also lot easier to operate and considerably faster than the earlier. One can

reshape, transpose, and modify data in matrix form with NumPy.

Seaborn

An open-source Python package for data visualization and graphing is called Seaborn. It uses sophisticatedPandas data structures and is based on the graphing software Matplotlib. Seaborn offers a high-level, feature-rich interface for creating precise, illuminating statistical graphs on its own. Because it can produce logical graphs of learning and execution data, it is employed in machine learning and deep learning applications.

The most beautiful and eye-catching graphs and plots are produced by Seaborn, which makes it ideal for use in publishing and marketing. Seaborn can also save you time and effort because it enables you to build complex graphs with little code and basic instructions.

Evaluation Parameters

Evaluation parameters are utilized to assess the performance of image encryption. There are many security attacks performed by the attackers to break the encryption approach as well as to find the key. Attackers mainly utilize the cryptanalysis to study the encryption approaches [8]. Therefore, it is necessary to hide the statistics of plaintext and the secret key. The strength of image encryption can be evaluated utilizing security and quality analyses. The quality analysis assesses the image quality of decrypted image utilizing peak signal-to-noise ratio, mean square error, etc.

The security analyses include statistical analysis, differential analysis, and key analysis. Statistical properties of the generated cipher image can be tested utilizing entropy, correlation coefficient, and histogram analysis. It is required that the encryption approaches do not provide statistical details of the plain image. Sometimes, we assume that an attacker obtains the details of the encryption approach without knowing the key.

In other words, the key is considered to be embedded in the encryption approach. Then, the attacker supplies an image to the encryption approach and gets a corresponding cipher image. Thereafter, he made small changes in the same image and got another cipher image. Then, he tries to find the similarity between two ciphered images to break the encryption approach.

It means that the encryption approach is required to be sensitive to small changes towards the plain image. It is assessed utilizing differential analysis. In this, unified average changing intensity and number of pixel change rate metrics can be utilized for the same. As we know that the performance of the image encryption approach is mainly dependent on the key, therefore, it should be large enough, so that it cannot be guessed easily.

Secondly, it should be sensitive to small changes. *e encryption approach should generate a totally different cipher image, even if the only one-bit difference is present in two keys. While there is transmission over a noisy channel, the cipher image may get affected. Therefore, the encryption approach should be robust against noise attacks.

Image Encryption Approaches

Different types of image encryption approaches are designed so far. By reviewing the literature, we have divided it into different types such as spatial, transform, optical, and compressive sensing based image encryption approaches. Figure demonstrates the categories of image encryption approaches. In the preceding subsection, these approaches are discussed and analyzed utilizing evaluation metrics. These parameters are KA, NPCR, HA, UACI, IE, CC, and NA. In comparisons, ✓ and ✗ symbols are utilized to represent whether the given approach has considered the respective metric and not, respectively.

Image Encryption in Spatial Domain. The approaches that are directly manipulating the pixels of the image are considered as spatial domain approaches. The various spatial domain-based image encryption is present in the literature. But we have considered the most famous approaches such as chaotic-based, elliptic curve-based, fuzzy-based, DNA, and Metaheuristics-based approaches.

Chaos-Based Image Encryption Approaches. Chaotic maps have great significance in the field of encryption. These maps generate random numbers that are utilized as secret keys in encryption . The reason is its properties such as dynamic and deterministic nature, sensitive to initial conditions, and ergodicity. Different types of chaotic maps are utilized so far. But these are mainly divided as one-dimensional and higher-dimensional chaotic maps.

Chaotic maps help in performing the confusion and diffusion operations in the encryption process. Figure demonstrates the diagrammatic flow of the chaotic maps in the image encryption approach. Chen et al. developed an image encryption approach utilizing a 2D sine map and Chebyshev map. It designed an ant degradation universal approach for chaotic maps, which improves the performance even on low-accuracy devices.

Xuejian and Zahur proposed image encryption based on spatiotemporal chaotic map and DNA encoding. Firstly, a plain image is changed into three DNA matrices dependent on a random encoding rule; afterward, DNA resultant is joined into a modern matrix. Then, it is permutated by the ascent matrix to generate the ciphered image. Wang et al. utilized coupled

map lattices (CML) and the DNA approach to encrypt the images. Ismail et al. examined a new lossless image encryption system that was based on fractional-order and double-humped logistic maps.

Wu et al. [23] designed an encryption approach utilizing a 2D discrete wavelet transform and hyperchaotic system for color images. Chai et al. [24] designed an encryption approach for color images utilizing a 4D memristive hyperchaotic map with genetic recombination. Luo et al. [25] developed an image encryption approach based on quantum coding and hyperchaos system. Kumar Patro and Acharya

proposed an image encryption approach utilizing a piece-wise linear chaotic map (PWLCM). In this, a rotating permutation is applied row-wise and column-wise. At last, it applies a diffusion operation on the row, column, and block to generate the ciphered image.

Feng et al. [7] utilized a discrete logarithm and amemristive chaotic system to encrypt the images. Wang and Gao [8] developed an image encryption strategy based on matrix semitensor. Hyperchaotic Lorenz map is also utilized to generate random numbers. Hua and Zhou designed an approach for encrypting the images that provide excellent effects against differential and statistical attacks.

Image filtering idea is utilized in image encryption to enhance the security of encryption. Gan et al. implemented an image encryption approach based on 3D bit-plane confusion. Lu et al. proposed an image encryption approach based on chaotic map and S-box. The discrete compound chaotic map was designed in this approach. S-box is also constructed utilizing logistic-sine system.

Deng and Zong [3] presented a binary image encryption approach based on chaotic mapping. The authors hypothetically examined the approach and figured out that the approach did not need to have the earlier information on the orbital distribution and one can pick out any chaotic model. Patro et al. developed a color image encryption approach that overcomes the drawbacks of execution blocklevel dispersion processes in arbitrary sized images. Wang et al.implemented an image encryption approach utilizing logistic-dynamic mixed linear-nonlinear coupled map lattices.

CHAPTER 3

REVIEW OF LITERATURE

Table 3.1 Review of Cryptography and Technique used for Analysis of watermarking cryptographic algorithm for secure transmission of image

S.No	Title of thePaper	Authors&Year	Algorithm	Results
1.	A comprehensive review on image encryption technique	M. Kaur and V. Kumar 2019	AES	83.3% Accuracy
2.	Double image encryption algorithm based on compressive sensing and elliptic curve	G. Ye, M. Liu, and M. Wu 2020	Double image encryption algorithm	96.5% Accuracy
3	An efficient image encryption scheme for TMIS based on elliptic curve integrated encryption and linear cryptography	Bensalem, Y. Rhaskali, and K. Douche 2022	TMIS based on elliptic curve	93%. Accuracy
4.	A comprehensive survey on encryption techniques for digital images	M.Singh, and A. K. Singh 2019	DES	80.8% Accuracy
5	Implementation of new message encryption using elliptic curve cryptography over finite fields	Y. Genk¸ and E. Arakan 2019	elliptic curve cryptography	93.98% Accuracy
6.	RGB image encryption based on symmetric keys using Arnold transform, 3D chaotic map and affine hill cipher	S. Qureshi, and M. A. Lone 2017	RSA	88% Accuracy
7.	Image encryption based on elliptic curve cryptosystem	Z. K. Obaid, and N. F. H. Al Saffir 2020	SHA	86% Accuracy
8.	Multiple RGB images encryption algorithm based on elliptic curve, improved Diffie Hellman protocol	Bashir, et al 2020	SF	89% Accuracy

9.	Image encryption method based on improved ECC and modified AES algorithm	A. Hafsa, et al. 2017	ECC, AES	77.26% Accuracy
10	Mapping images over elliptic curve for encryption	Basra, M. Saqib, and A. H. Moon 2020	Elliptic curve for encryption	89.00% Accuracy

Numerous works have been performed to make ECC encryption possible after each pixel has been mapped to a predefined elliptic curve [16–18]. The algorithm is used to perform faster, but complicated in precomputation calculations to find each point of the elliptical curve for a large value of the primary number used to generate the finite field. It also includes communication of the wide mapping table through the unreliable channels for the decryption process. Several research studies have focused mainly on the use of AES as an encryption and decryption algorithm.

While the algorithm claims to perform better than other methods, it did not include concrete results for the same through a variety of security and encryption quality metrics, such as Entropy, NPCR, UACI, etc.The various approaches involve AES with visual cryptography which offers good results by encrypting the image using AES and the original key using visual cryptography by converting it to an image. The algorithm is still susceptible to an attack on the shared image created for the key. Hashim et al. used ElGamal encryption as an asymmetric encryption algorithm and have been tested using MATLAB. The work concluded that it took an increasing amount of time for computation using a large prime number as the encryption parameter.

Bhowmick et al. [24] worked on the security of the text encryption provided by Double Playfair Cipher using 6 key matrices over the regular 5 5 key matrices. However, the algorithm failed due to data loss overcertain characters, such as spaces and special symbols. An updated 5 5 playfair cipher version is presented which enables the user to encrypt and decrypt messages for any square matrix. The fair play cipher, with the unique encoding instructions, is introduced as the first digraph cipher. Hardi et al. [25] combined the use of the ElGamal cryptosystem and the Double Playfair cipher to protect text data using standard keys for the symmetrical method. While the algorithm appears to be operating on digital media, it fails to evalute security measures and metrics. Image file encryption performed using hybrid cryptography

ElGamal algorithm used to perform asymmetric encryption and Double Playfair for symmetric encryption.The result has been proved that these algorithms are capable of encrypting an image file with an appropriate runtime and encrypted file size while maintaining the security level. Hamad et al.worked to increase image encryption protection utilizing standard Playfair cipher using a modified key of size 16 by 16 on the 8-bit pixel range. By carrying out an XOR operation using a random mask, the effects are further enhanced. The additional security of the algorithm through the XOR function fails if the intruder eavesdrops the mask. An algorithm incorporating XOR encryption with a rotational process was designed to effectively encrypt images. Arab et al.proposed the Advanced Encryption Standard (AES) and Visual Cryptographic techniques for images.

Secure image encryption algorithm used for both AES and Visual Cryptographic techniques to protect theimage. The image is encrypted using AES and an encoding schema has been proposed to convert the key into shares based on Visual Secret Sharing's et al. introduced the Cryptography of the Elliptic Curve for images. The proposed work is designed to provide secure authentication to combine image encryption with Elliptic Curve Cryptography.

The matrix operations are carried out on the original image matrix, and the transformed image is further encrypted using a key sequence generated from the elliptic curve. This is a highly secure technique and difficult to get the original data without the key. The system requires high computation which makes it slower.Dawahdeh et al.have adopted the encryption technique incorporating the Elliptic Curve Cryptosystem with Hill Cipher. The researchers selected ECC asymmetric encryption and Hill Cipher for symmetric encryption. The proposed algorithms are capable of encrypting an image file with security measures. More secure as a hybrid approach is used, and faster in computing. A new self-invertible key matrix technique was proposed. It uses single matrix to encrypt the pixels within the image but it takes longer.

Observations

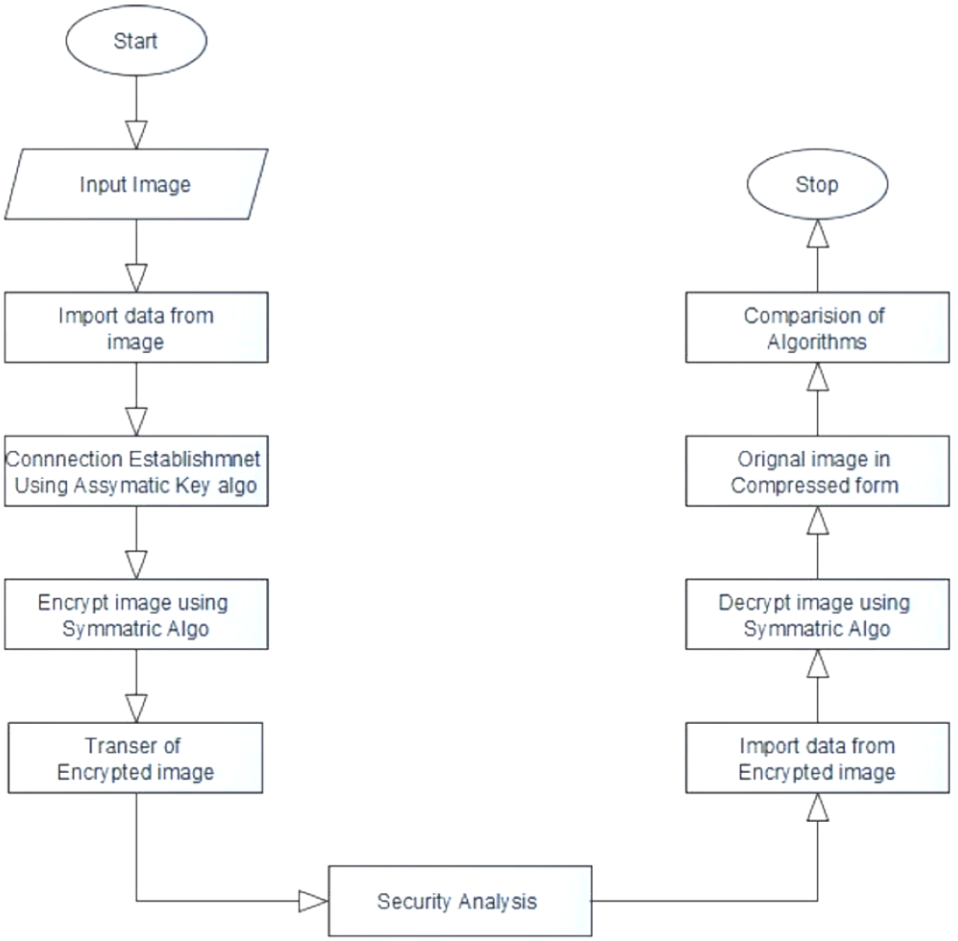
This section deals with related work in the field of image encryption and decryption. Security becomes a significant problem in today's environment, for both storing and transmitting multimedia data. Therefore, data must be protected from unauthorized access and the attacker must thwart the attack during transmission. Images have a major role to play in multimedia data. This represents more information when compared with text details through visualization For attack detection, a variety of algorithms have been proposed, although techniques incorporated with the water marking filters and ECC, AES algorithms are very unique in the proposed system. This project presents a way of analyzing the secured image and transferring to the concern user with end to end encryption, decryption methodologies.

CHAPTER 4

METHODOLOGY

4.1 FLOW DIAGRAM

Effectively in Wireless Sensor Networks (WSN) the implementation of the SF algorithm offers very low complexity architecture [5]. The encryption and decryption process is almost similar in this algorithm. The five encryption rounds increase the effectiveness of the encryption procedure. This is a four step algorithm based on simple mathematical operations (six numbers of operations) that operates on a four bit data. But this causes a huge amount of misunderstanding and dispersal of data is created to combat dissimilar types of attacks.



4.2 FEATURE SELECTION

ECC is used to ensure the security of the data over cloud storage. The maintenance of the data with reduced key size can help to optimize the storage space as well as get the desired results. It uses the same 3072 bits as the RSA. The most beneficial thing about the ECC is the reduced key size and the encryption of data through a public key, which is in an optimized manner [3]. ECC is more beneficial compared to the RSA for using the latest algorithmic techniques applying to the encryption and decryption of data as well as the accuracy of the decrypted data. AES has many performance operations which are limiting on cloud storage like statistical analysis, searching on cloud storage, and others like these. It is the widely used strategic algorithm over cloud computing to improve the security rules over cloud storage. The public key is known by every person while the encryption and decryption process can also be done by the public key [3]. Filter Methods.

Digital watermarking has attracted the interest of numerous researchers both in academia and industry and has become one of the hottest research topics in the multimedia signal processing community. Although the term watermarking has slightly different meanings in the literature, one definition that seems to prevail is the following

[1] Watermarking is the process of imperceptibly altering a piece of data in order to embed information about the data. That definition reveals two important characteristics of watermarking. First, information embedding should not cause perceptible changes to the host image (sometimes called cover image or original image). Second, the message should be related to the host image.

In this sense, the watermarking techniques form a subset of information hiding techniques, which also include cases where the hidden information is not related to the host medium (e.g., in covert communications). However, certain authors use the term watermarking with a meaning equivalent to that of information hiding in the general sense.

A watermarking system should consist of two distinct procedures: a procedure that embeds the information in the host data and a procedure that extract or recover the watermark after being attacked with different image processing attacks.

Requirements of Digital Image Watermarking

Imperceptibility (Visibility)

The watermark should not be visible in the image under typical viewing conditions and should not affect the quality of the host image.

Robustness

The watermark can still be detected after the image has undergone linear or nonlinear image processing operations intentionally or unintentionally like compression, cropping, rotation and noise. So the watermarks should be robust against variety of such attacks.

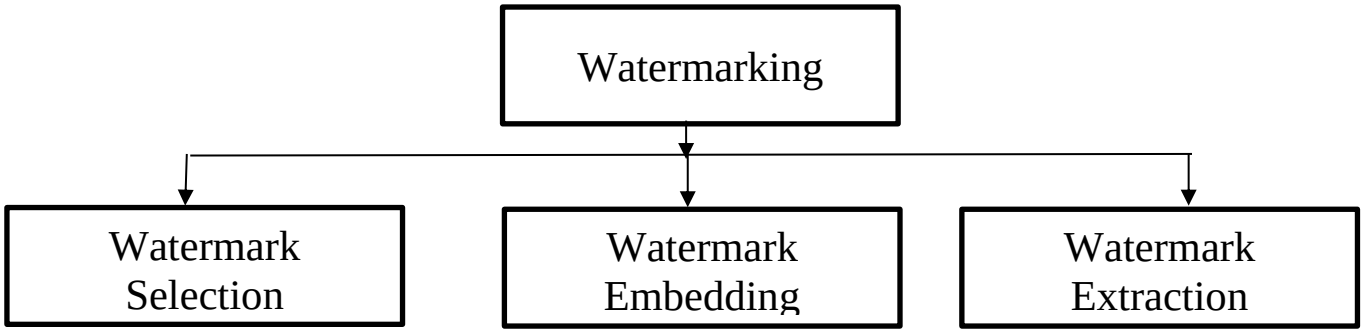
Capacity or Data Payload

The watermarking technique must be capable of allowing multiple watermarks to be inserted in an image, with each watermark still being independently verifiable and can be successfully detected during extraction.

Security

The security of watermarking techniques can be interpreted in the same way as the security of encryption techniques. Hence a watermarking technique is truly secure if knowing the exact algorithm for embedding and extracting the watermark does not help an unauthorized party to detect the presence of the watermark or remove it without knowing the secret key.

The Watermarking process is divided into three steps as They are, Watermark Selection, Watermark Embedding and Watermark Extraction.



4.2.1 Watermarking Processes

Watermark selection

First of all, the appropriate watermark is selected. There are two types of watermarks that can be embedded in an image.

Pseudo-Random Gaussian Sequence

A Gaussian sequence watermark is a sequence of numbers comprising 1 and -1 and which has equal number of 1's and -l's is termed as a watermark. It is termed as a watermark with zero mean and one variation. Such watermarks are used for objective detection using a correlation measure.

Binary or Grey Scale Watermarks

Some watermarking algorithms embed meaningful data in form of a logo image instead of a pseudo-random Gaussian sequence. Such watermarks are termed as binary image watermarks or grey scale watermarks. Such watermarks are used for subjective detection. Example of grey scale and binary image is shown in figure 4.2.1

Fig.4.2.1 Watermarking image

Table 4.2 Sample input and output for proposed hybrid algorithms

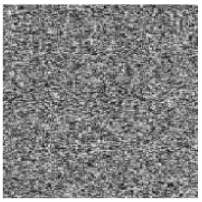
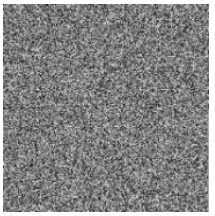
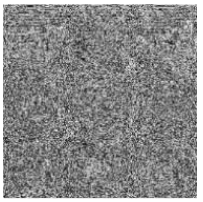
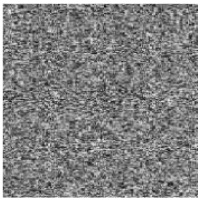
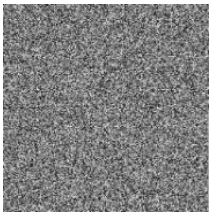
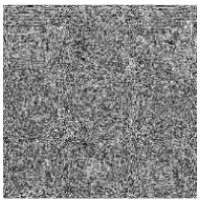
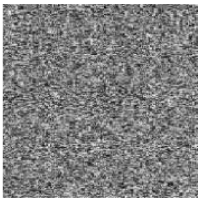
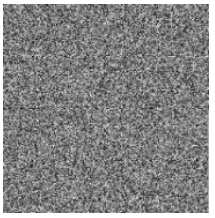
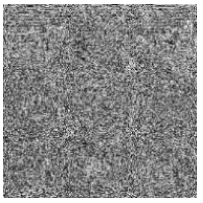
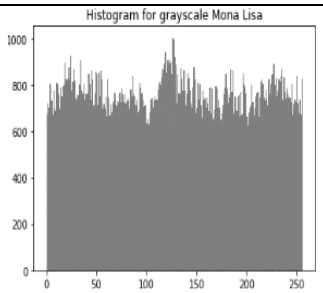
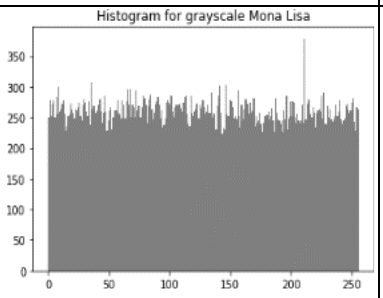
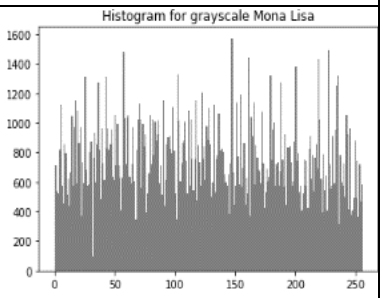
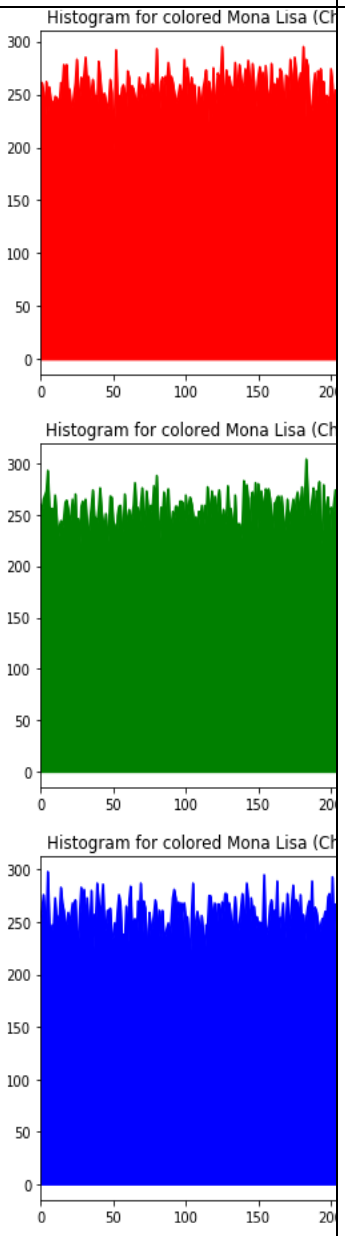
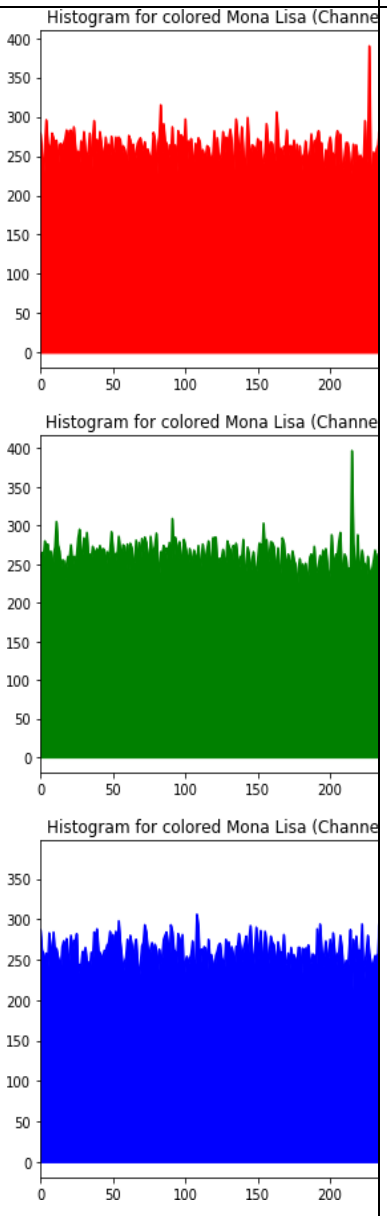
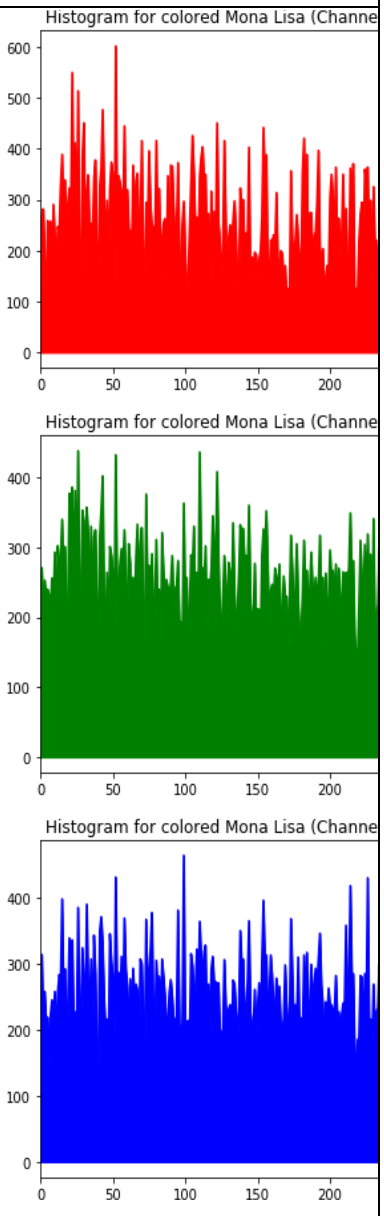
ImageName	Original Image	Encrypted Image	ECC with AES	Encrypted Image	Decrypted Image
Image 1					
Image 2					
Image 3					

Image	ECC with Hill Cipher	ECC with AES	Double Playfair Cipher)
Image 1			
Image 2			

PURPOSE OF CRYPTOGRAPHY

Cryptography provides security to ensure the privacy of data, non-alteration of data and so on. Nowadays

cryptography is widely using due to the great security. There are the various cryptography goals are following as,

Confidentiality

The transmission of data from one computer to another computer has to be accessed by an authorized user and it not access by anyone else.

Authentication

The transmission of data from one computer to another computer has to be accessed by an authorized user and it not access by anyone else.

Integrity

Only the authorized party is allow to modify the transmitted information. And an unauthorized persons should not allow to modify in between the sender and receiver.

Non-Repudiation

Ensures the message that sender or the receiver should be able to deny the transmission.

Access Control

The authorized persons only able to access the information while in transfer.

Watermark Embedding

After selecting the watermark, it can be embedded using different techniques such as: spatial domain or the frequency domain techniques.

Watermark Extraction

The watermark can be extracted from the watermarked image by using the same technique used in the embedding. Extraction can be done with the presence of the host image or the absence of the host image depending on the watermarking system.

Symmetric or private key

In such schemes, both watermark embedding and extraction are performed using the same key K.

Asymmetric or public key

In contrast to the previous class, these watermarks can be extracted with a key that is different than the one that was used in the embedding stage. Actually, a pair of keys is used in this case: a private key to generate the watermark for embedding, and a public one for extraction. For each private key, many public keys may be produced. Despite their advantages over their symmetric counterparts, asymmetric schemes are much more difficult to devise.

Watermarking Techniques

Images can be represented in spatial domain and or frequency domains [12]. In the frequency domain, images are represented in terms of their frequencies, while in the spatial domain images are represented by pixels.

Spatial Domain Techniques

Simple watermarks can be embedded in the spatial domain of images by modifying the pixel values or the least significant bit (LSB) values. This is easy to implement but it is not robust enough to protect watermark information against different kinds of attacks such as the lossy compression. The spatial domain methods are applicable for fragile watermarking scheme.

Frequency Domain Techniques

Frequency domain transfers an image to its frequency representation and the image is segmented into multiple frequency bands. The embedded watermark in the frequency domain of a signal can provide more robustness than spatial domain. It is strong against attacks like compression, cropping where spatial domain is not. Several reversible transforms like Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT) can be used. Each of these transforms has its own characteristics and represents the image in different ways.

The Discrete Cosine Transform (DCT)

The discrete cosine transforms (DCT) is a technique for converting a signal into elementary frequency components. It represents an image as a sum of sinusoids of varying magnitudes and frequencies. With an input image, x, the coefficients for the output "image," y.

The Discrete Wavelet Transforms (DWT)

A two dimensional Discrete Wavelet Transform (DWT) is a mathematical formula which transforms an image from spatial domain to frequency domain. The DWT of an image I(m, n) is calculated by passing it through a series of filters.

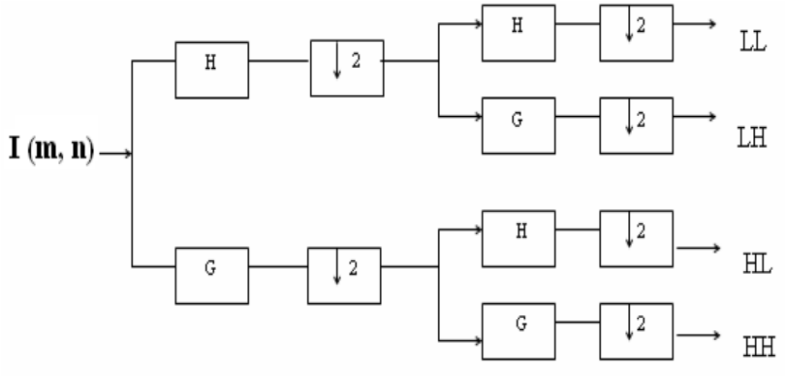


Figure 4.2 DWT Process

The samples are passed through a low pass filter, H, and a high pass filter, G. The signal that results out of each filter has the same number of samples as the original signal, so the result is a signal with double the number of samples. This is solved using “down sampling” in which half the samples are discarded. Down sampling is illustrated with the down sampling operator

4.3 AES ECC ALGORITHM

Data pre-processing is a vital phase in machine learning that improves the quality of the data to promote the extraction of valuable insights from the data. Preparing (cleaning and arranging) raw data in orderto make it acceptable for creating and training Machine Learning Model.

It raises reliability and accuracy. Pre-processing data can increase the correctness and quality of a dataset, making it more stable by removing missing or inconsistent data values brought on by human or computer mistake. It ensures consistency in data. the security of data is a major issue which can be compromised in different ways using either external or internal means.

To protect the transmission of data over the Internet, different encryption techniques are used. The problem with these techniques is that they need large key size, large memory size, and require a lot of computation power to protect the data. As in AES encryption, a key is generated soon after the input file is uploaded, and we know AES uses symmetric key encryption method in which a single key is used for encryption as well as decryption.

Proposed Framework

This highlight the significance of combining ECC and AES and focus on the algorithm used in this approach.

`

ECC and AES—A Combined Hybrid Proposed Approach

ECC and AES create the most advanced and efficient cryptographic technique over the cloud storage. We can say that single AES is little bit slower than the hybrid (ECC-AES) method due to its larger key size, while the hybrid method allows reduced key size as well as a faster security mechanism for securing the data. As small key size is the main property of ECC, when AES uses ECC for encryption, key size is reduced, and performance is increased [3]. ECC uses encryption and decryption key standards to reduce the key size and create the secured key system.

ECC is the most appropriate technique to use along with AES to get the data secured from unauthorized use. Once the key size is set, then ciphertext will generate the encryption and decryption of data. The key generated by ECC is used by AES. The combined effect of both ECC and AES is suitable for the proposed technique at cloud storage to get the secured system.

4.3.1 Key Generation for Cipher

A modified key space of 2 matrices, each of size 16 - 16 is used instead of standard 5 x 5 key matrices
used in Double Playfair cipher.

Create 2 key maps, 1 corresponding to each of the keys. The key maps have pixel intensities as keys and
their location in the respective key matrix as the corresponding value (represented by a tuple of row and column).

Key Generation for ElGamal Cryptosystem

Receiver chooses a prime number p and computes its generator g - gd (mod p)

Publish the public key, prime number p, and generator g.

Sender chooses a random integer i in the range: f2, p x 2g.

Sender computes the temporary key or ephemeral key as: KE = gi(mod p)

The masking key is computed by sender as: KE – Kpubi (mod p)

Encryption process

Read the image to be encrypted and separate it to different channels as R, G, B. Pair up pixels in 2 for
each of the channels separately.

To apply Vertical Double Playfair Cipher find the location of first value from pixel pair from the first key map and the location of second pixel value from the second key map.

Decryption process

The masking key will be computed by receiver as: KM KE (mod p).

The key matrices for Double Playfair Cipher are decrypted as:

$$pij = cij\ (KM\square\ 1)(\text{mod } p).$$

Substitute the decrypted pixel values to create the original image

Image Encryption and Decryption Implementation

The flow diagram for the proposed hybrid encryption and decryption algorithm is shown in Figures. A connection is established using the key generated by the Elliptical Curve Cryptography (ECC) asymmetric key algorithm. The image encryption is by the symmetric hybrid algorithm of Elliptic Curve Cryptography (ECC) with Hill Cipher, ECC with Advanced Encryption Standard (AES), and ElGamal with Double Playfair Cipher. The encrypted image is securely sent to the sender. The image is decrypted by the receiver using this symmetric hybrid algorithm, and the original image retrieved will be compare.

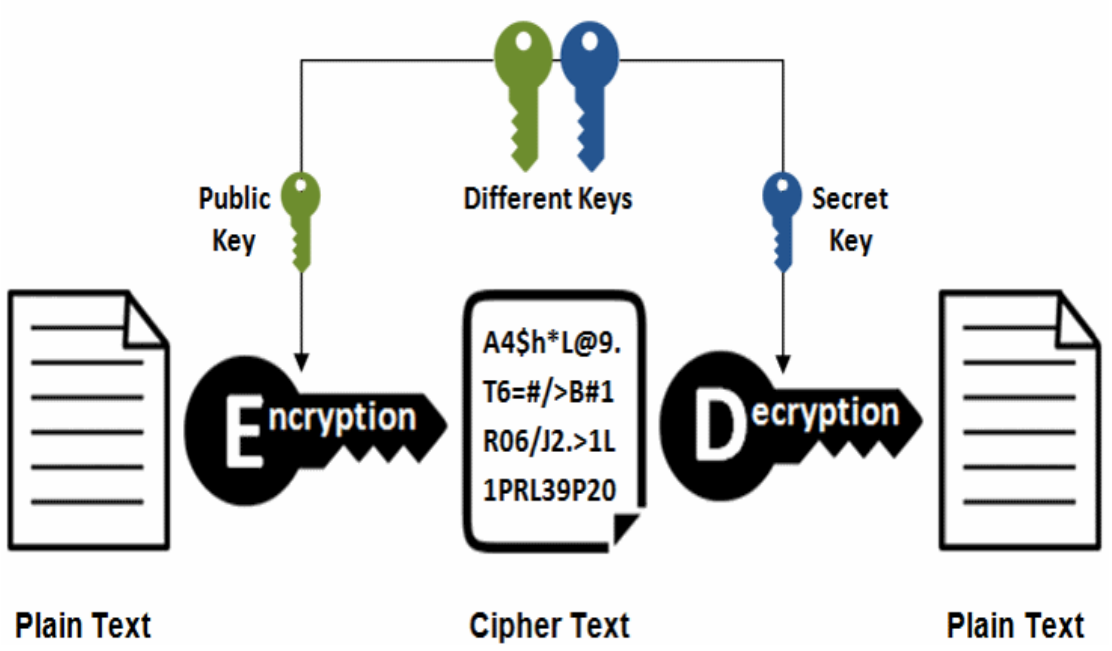


Fig 4.3.1 Asymmetric Encryption

Figure 4.3.1 The above process can be directly applied for the RSA cryptosystem, but not for the ECC. The elliptic curve cryptography (ECC) does not directly provide encryption method. Instead, we can design a hybrid encryption scheme by using the ECDH (Elliptic Curve Diffie–Hellman) key exchange scheme to derive a shared secret key for symmetric data encryption and decryption.

Algorithm 1 Watermark Encryption

Input Grayscale image

1: Consider the grayscale image W.

2: Generate the decimal value for each pixel (Pi) ofW.

3: Find out the corresponding binary value (Bv) of each Pi.

4: Reverse that 8 digit's binary number Bv to get Rv.

5: Consider a 4 digit divisor as the Key (Ke).

6: Now divide the reversed number Rv with the divisor Ke.

7: Next store the remainder and quotient in an 8-bit string. If required, add the number of 0s on the left-hand side of remainder and quotient bits to complete the 8-bit string.

This leads to being the encrypted data (ED).

Output Encrypted image

Algorithm 2 Watermark Decryption

Input: Encrypted image data (ED) and the key (Ke)

1: Multiply the quotient bits of the encrypted data (ED) by the Key (Ke) to produce F.

2: Add the remainder bits of the encrypted data (ED) with the result produced in the above step (F) to get G.

3: If the result produced (G) in the previous step i.e. step 2 is not an 8-bit number, then we require, making it an 8-bit number.

4: Reverse the number G to get the decrypted data (DD).

Output- Decrypted image

ECC Encryption / Decryption

In this section we shall explain how to implement elliptic-curve based public-key encryption / decryption (asymmetric encryption scheme based on ECC). This is non-trivial and usually involves a design of hybrid encryption scheme, involving ECC cryptography, ECDH key exchange and symmetric encryption algorithm.

Assume we have a ECC private-public key pair. We want to encrypt and decrypt data using these keys. Asymmetric encryption works as follows: if we encrypt data by a private key, we will be able to decrypt the ciphertext later by the corresponding public cryptography

4.4 MODEL BUILDING

ECC is used to ensure the security of the data over cloud storage. The maintenance of the data with reduced key size can help to optimize the storage space as well as get the desired results. It uses the same 3072 bits as the RSA. The most beneficial thing about the ECC is the reduced key size and the encryption of data through a public key, which is in an optimized manner. ECC is more beneficial compared to the RSA for using the latest algorithmic techniques applying to the encryption and decryption of data as well as the accuracy of the decrypted data. AES has many performance operations which are limiting on cloud storage like statistical analysis, searching on cloud storage, and others like these. It is the widely used strategic algorithm over cloud computing to improve the security rules over cloud storage. The public key is known by every person while the encryption and decryption process can also be done by the public key. Many advantages and key sizes of ECC and AES over RSA are given in the Table below

Table 4.4 ECC and AES over RSA

ECC	RSA	Key Size Comparison
160 bits	1024 bits	1:6 bits
256 bits	3024 bits	1:12 bits
384 bits	7068 bits	1:20 bits

As you can see in the Table, a medium key size is required for ECC as compared to RSA. Because of this, it provides better security than RSA. ECC-AES also provides better security with a smaller key size as compared to other cryptographic algorithms. It optimizes memory space and reduces computational complexity because of the smaller key size. Thus, by using a medium key size, a high level of data security can be obtained.

Setup

We used Python to implement our proposed algorithm and to validate the efficiency and uniqueness of our proposed system. We used SimpleCV library for Python to read the images. The image format used by our code was Portable Network Graphics. The uniqueness and efficiency of the system by the combination of ECC and AES was enhanced in the different way that many of the data over cloud storage were secured and achieved the secured connections for the encryption and description of the data.

Data Size for Proposed Scheme

We took three different datas for the comparison of our proposed method with other existing schemes because images usually take more time than text data, so we wanted to check the computational cost as well as the time required to complete the encryption and decryption of the images. The reason for taking different images size is to compare performance in multiple scenarios. Figure represents the encryption time comparison of different algorithms.

The hybrid algorithm is the proposed algorithm. Results are prominent in that the hybrid ECC-AES approach took less time to encrypt data than the existing approaches due to its smaller key size. Moreover, the hybrid ECC AES algorithm has characteristics of both algorithms which provides higher security by increasing the complexity and making the system strong against attacks.

It can also be clearly seen that encryption time for the proposed h algorithm is much less as compared to other algorithms. As the time for encryption is reduced, our computational cost is also reduced, which is very effective. Hence, our proposed approach works more efficiently than others.

4.5 PERFORMANCE EVALUATION

4.5.1 EVALUATING MODEL PERFORMANCE

- Step 1- Read Data file.
- Step 2- Add watermark on the original data file.
- Step 3- By using crypto technique we blur the original image.
- Step 4- Designed Repository is cloned
- Step 5- Local host address is created for Encryption and Decryption process.
- Steps 6- Now encrypt the blur image by using algorithm and get encrypted data, which may be noisy.
- Step 7- The generated key applied to the image cipher.
- Step 8- Now using decrypt technique gets the original data.
- Step 9- Debugger pin is auto generated while processing

In this work we have used Py3 on the Standard image data to implement our proposed technique. The algorithm and watermarking process have been used to implement our hybrid technique. By the use of these two processes we have studied the decrypted image which was found to be a close match of the input image.

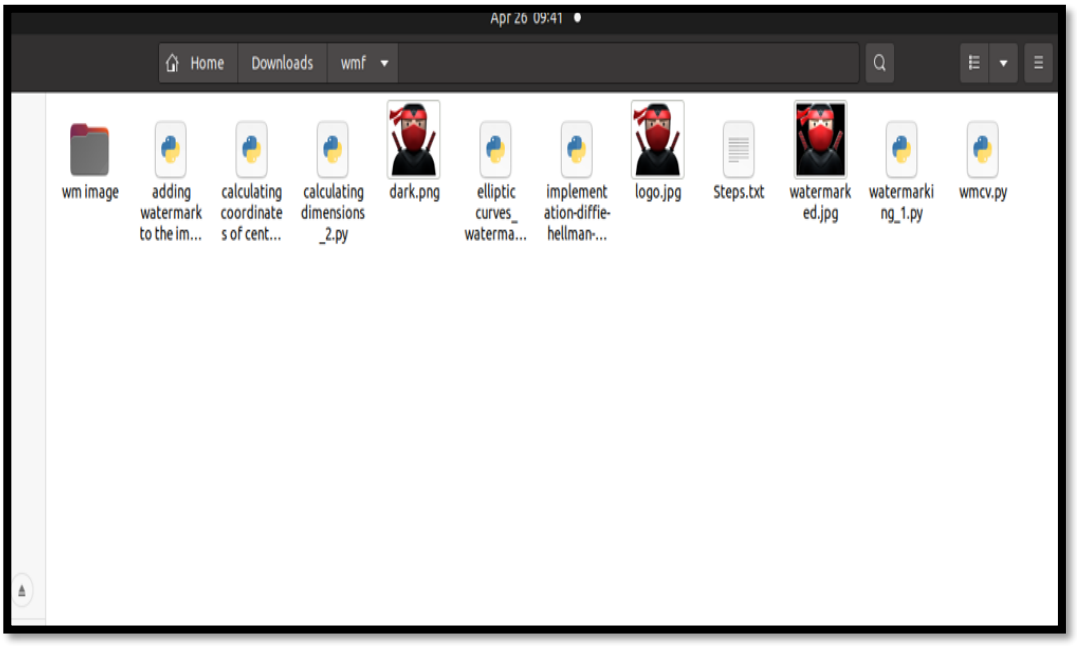


Figure 4.5.1 Before Watermarking Images

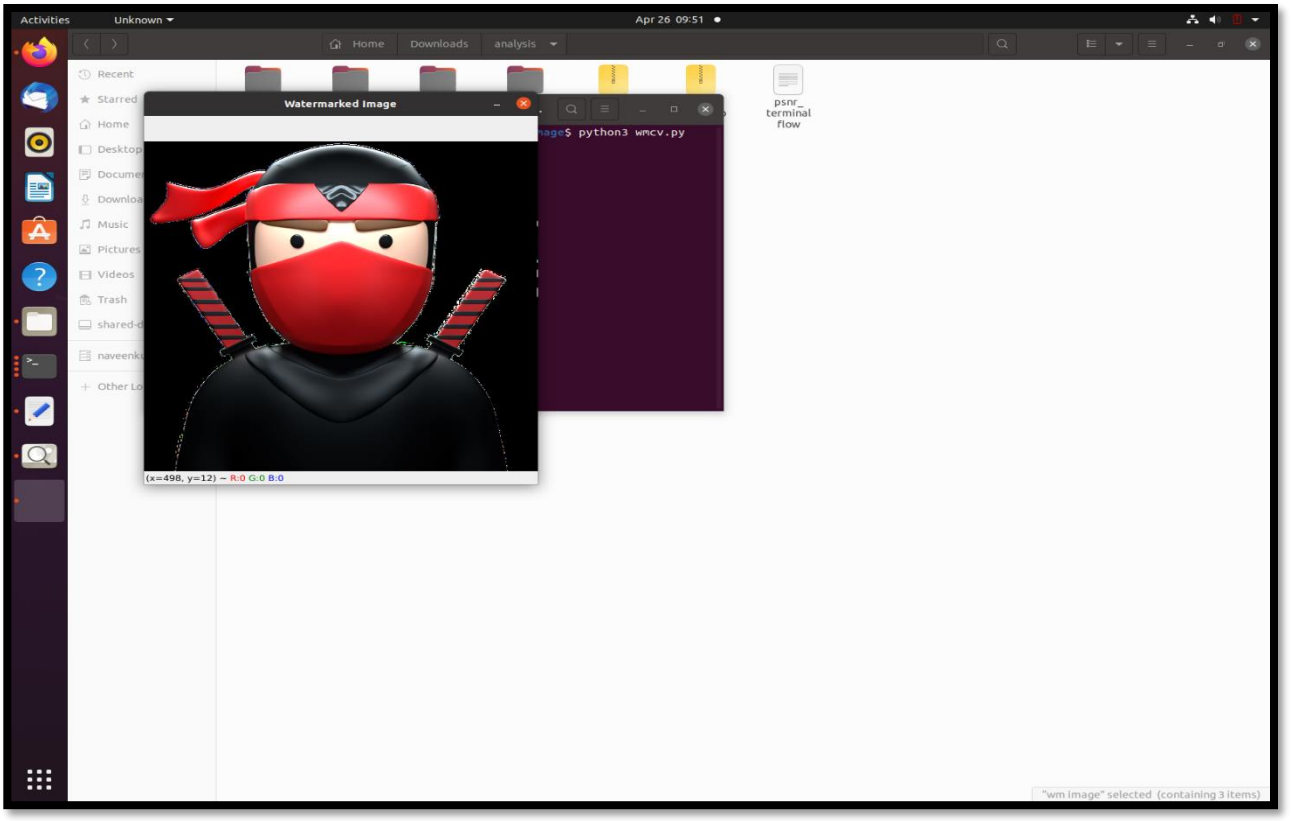


Figure 4.5.2 Watermarked Images

CHAPTER 5

RESULTS AND DISCUSSION

In this work we determine the most effective technique for identifying secured Encryption and Decryption techniques.

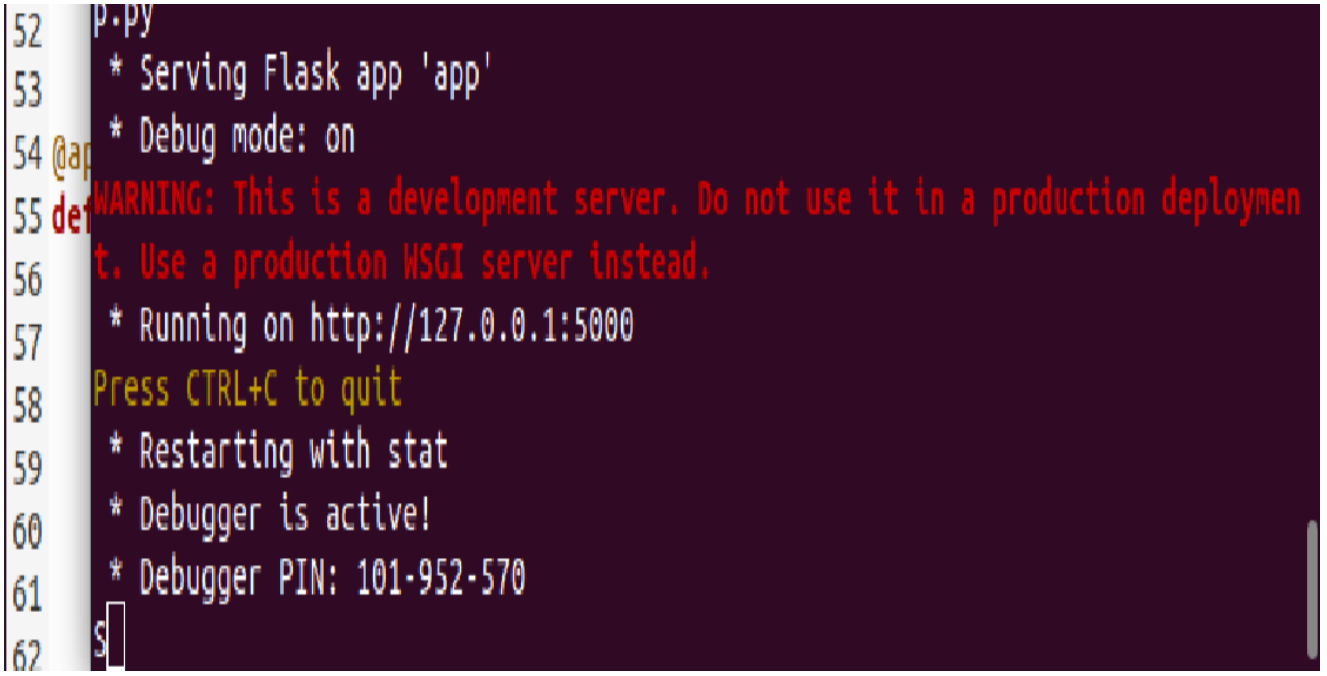


Fig 5.1 Debugger Mode

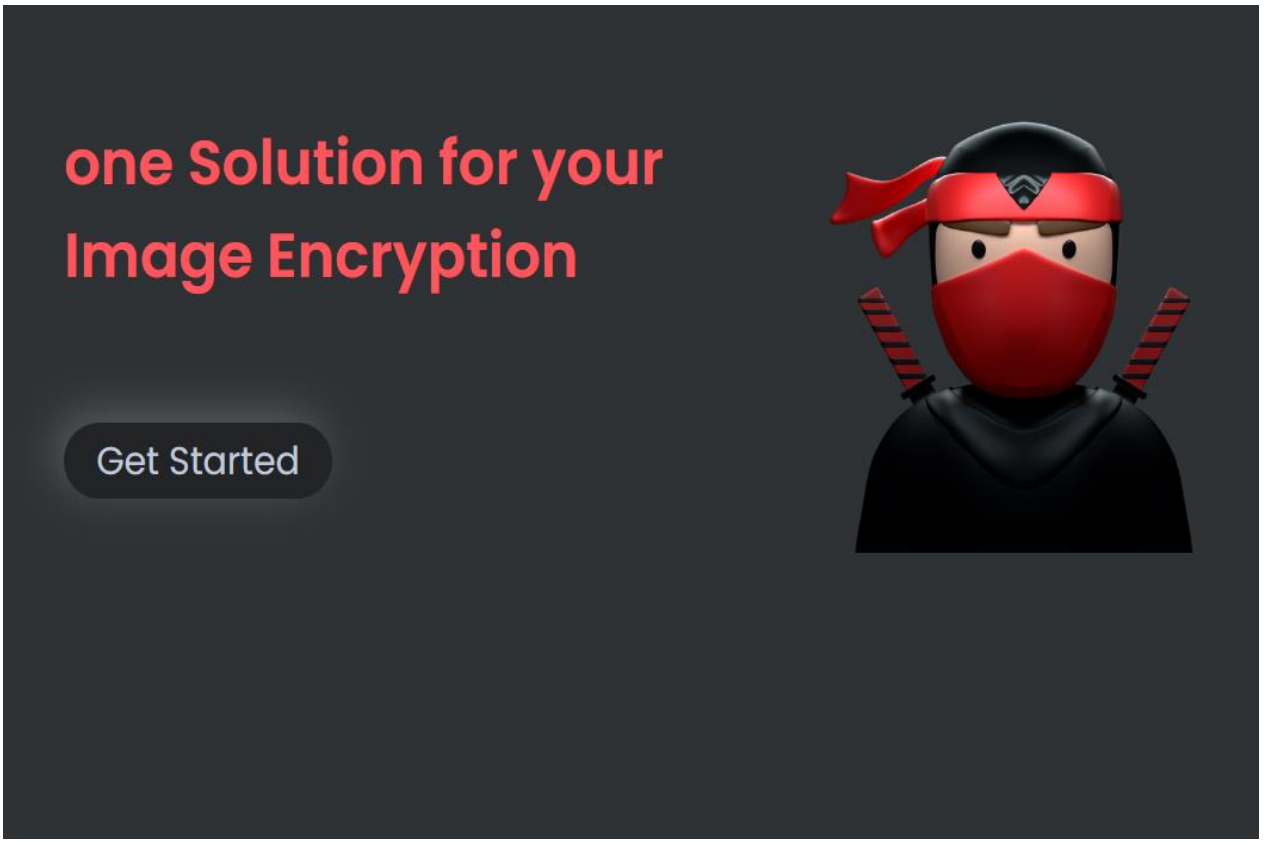


Fig.5.2 First Page

Fig5.3 Encryption and Decryption page

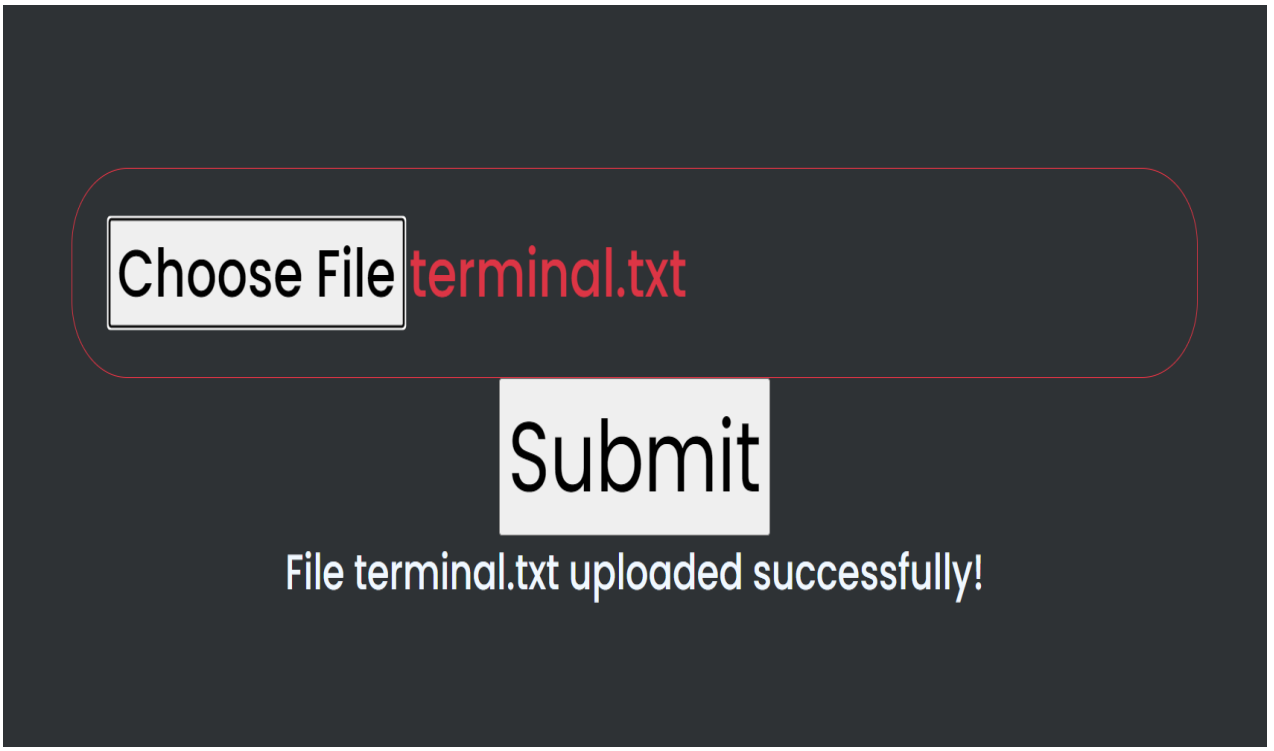


Fig5.4 Encryption

```
Public key: (n=0x9a11485bccb9569410a848fb1afdf2a81b17c1fa9f9eb546fd1deb873b49b693a
4edf20eb8362c085cd5b28ba109dbad2bd257a013f57f745402e245b0cc2d553c7b2b8dbba57ebda7
f84cfb32b7d9c254f03dbd0188e4b8e40c47b64c1bd2572834b936ffc3da9953657ef8bee80c49c2c1
2933c8a34804a00eb4c81248e01f, e=0x10001)
-----BEGIN PUBLIC KEY-----
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCaEUhbzLlWlBCoSPsa/fKoGxfB
+p+etUb9HeuHO0m2k6Tt8g64NiwIXNWyi6EJ260r0legE/V/dFQC4kWwzC1VPHsr
jbulfr2n+Ez7MrfZwlTwPb0BiOS45AxHtkwb0lcoNLk2/8PamVNlfvi+6AxJwsEp
M8ijSASgDrTIEkjgHwIDAQAB
-----END PUBLIC KEY-----
Private key: (n=0x9a11485bccb9569410a848fb1afdf2a81b17c1fa9f9eb546fd1deb873b49b693a4
edf20eb8362c085cd5b28ba109dbad2bd257a013f57f745402e245b0cc2d553c7b2b8dbba57ebda7f84
cfb32b7d9c254f03dbd0188e4b8e40c47b64c1bd2572834b936ffc3da9953657ef8bee80c49c2c12933
c8a34804a00eb4c81248e01f, d=0x318ab12be3cf0d4a1b7921cead454fcc42ba0704626394833394d6
fb9529547827e9c8d23b294a8e01f8a1019da34e350f2307740e06a270bef1fe646e6ad213e31b528fd
d5f5d03e633c07c44755ed622a629d79e822c095ebdf9cc80e517b5566dd3d3e5b16ec737987337a0e4
97fdba4b5ad97af41c1c3cdd87542a4637d81)
-----BEGIN RSA PRIVATE KEY-----
MIICXAIBAAKBgQCaEUhbzLlWlBCoSPsa/fKoGxfB+p+etUb9HeuHO0m2k6Tt8g64
NiwIXNWyi6EJ260r0legE/V/dFQC4kWwzC1VPHsrjbulfr2n+Ez7MrfZwlTwPb0B
iOS45AxHtkwb0lcoNLk2/8PamVNlfvi+6AxJwsEpM8ijSASgDrTIEkjgHwIDAQAB
AoGAMYqxK+PPDUobeSHOrUVPzEK6BwRiY5SDOU1vuVKVR4J+nI0jspSo4B+KEBna
NONQ8jB3QOBqJwvvH+ZG5q0hPjG1KP3V9dA+YzwHxEdV7WIqYp156CLAlevfnMgO
UXtVZt09PlsW7HN5hzN6Dkl/26S1rZevQcHDzdh1QqRjfYECQQDGDUIQXlOiAcGo
d5YqAGpWe0wzJ0UypeqZcqS9MVe9OkjjopCkkYntifdN/1oG7S/1KUMtLoGHqntb
c428zOO/AkEAxyV0cmuJbFdfM0x2XhZ+ge/7putIx76RHDOjBpM6VQXpLEFj54kB
qGLAB7SXr7P4AFrEjfckJOp2YMI5BreboQJAb3EUZHt/WeDdJLutzpKPQ3x7oykM
wfQkbxXYZvD16u96BkT6WO/gCb6hXs05zj32x1/hgfHyRvGCGjKKZdtwpwJBAJ74
y0g7h+wwoxJ0S1k4Y6yeQikxUVwCSBxXLCCnjr0ohsaJPJMrz2L30YtVInFkHOlL
i/Q4AWZmtDDxWkx+bYECQG8e6bGoszuX5xjvhEBslIws9+nMzMuYBR8HvhLo58B5
N8dk3nIsLs3UncKLiiWubMAciU5jUxZoqWpRXXwECKE=
-----END RSA PRIVATE KEY-----
Encrypted: b'99b331c4e1c8f3fa227aacd57c85f38b7b7461574701b427758ee4f94b1e07d791ab70b
55d672ff55dbe133ac0bea16fc23ea84636365f605a9b645e0861ee11d68a7550be8eb35e85a4bde6d73b
0b956d000866425511c7920cdc8a3786a4f1cb1986a875373975e158d74e11ad751594de593a35de765fe329c0d3dfbbfedc'
Decrypted: b'A message for encryption'
```

Fig 5.5 Encrypted.txt

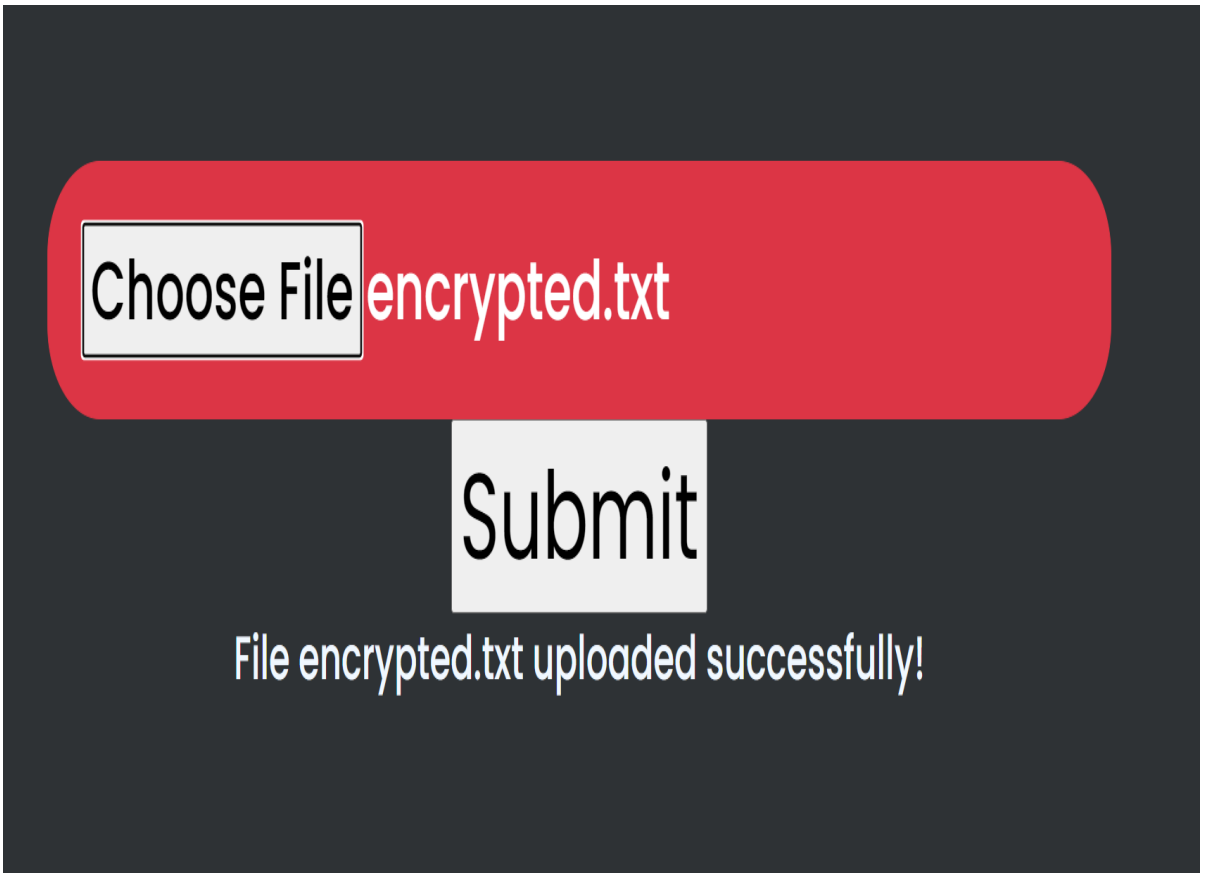


Fig 5.6 Decryption

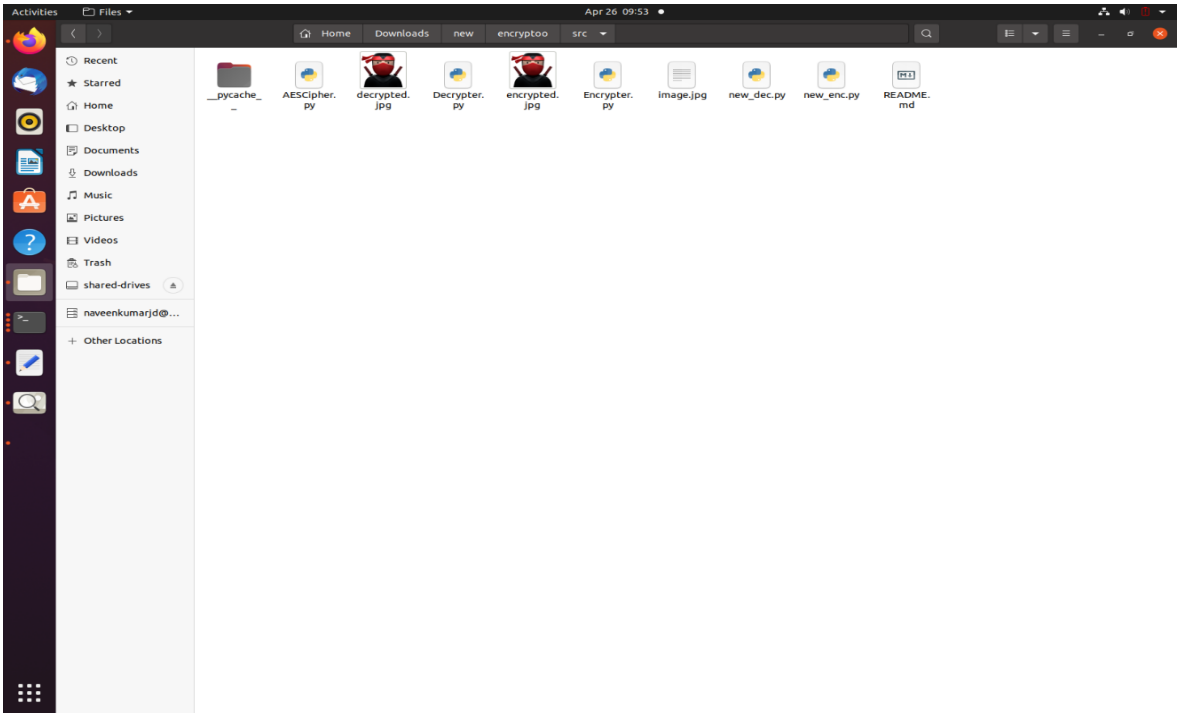


Fig 5.7 Decrypted Image

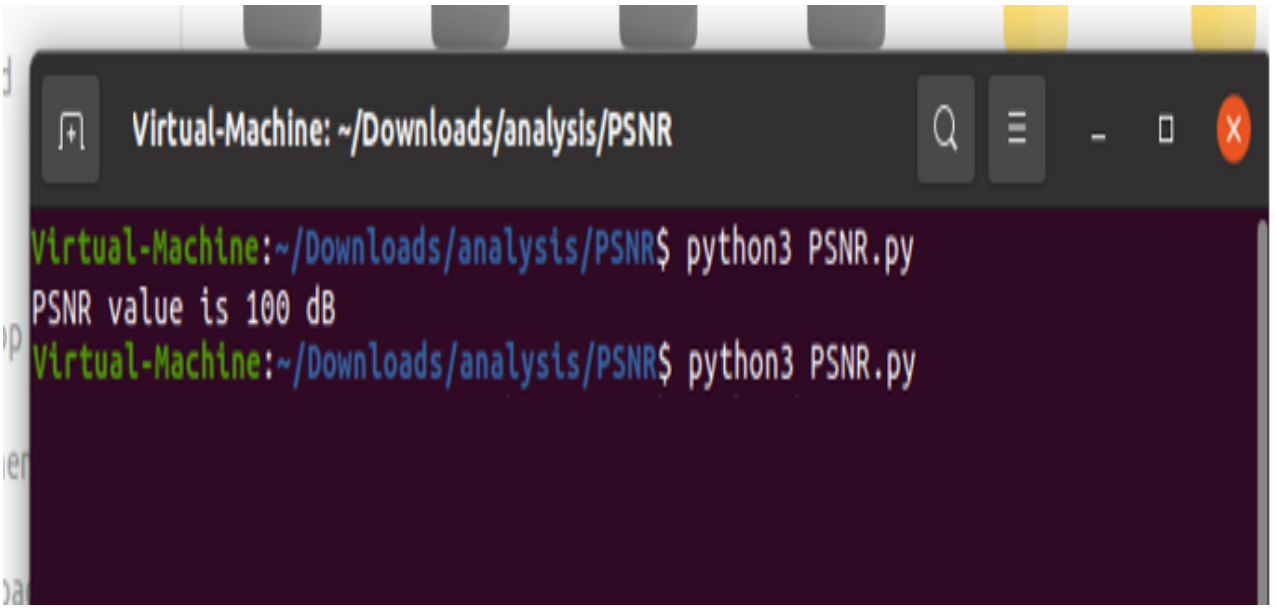


Fig 5.8. PSNR Value

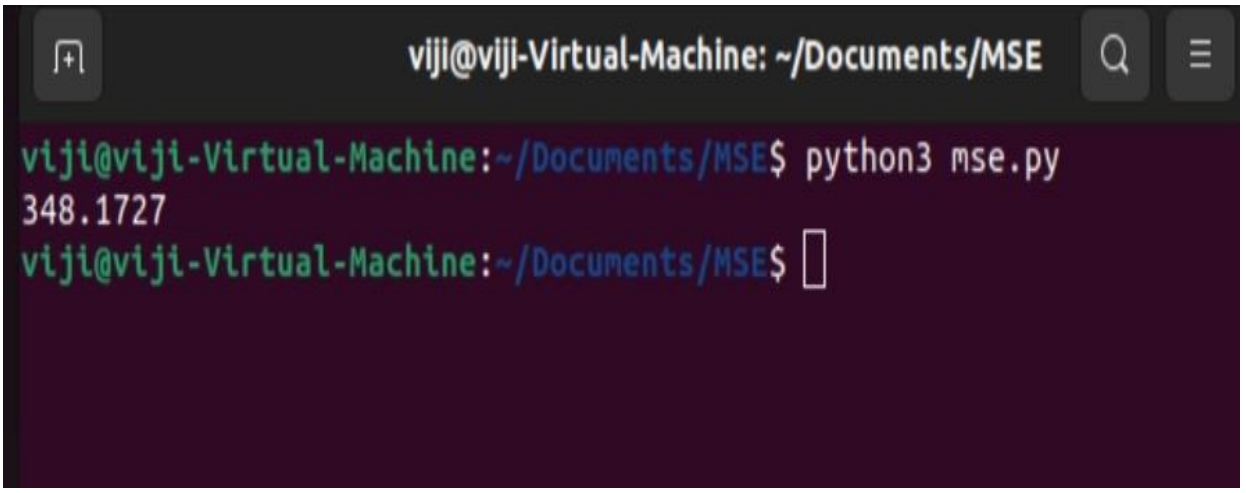


Fig 5.9 MSE Value

TABLE 5.10 Comparison of PSNR, MSE

Parameter	PSNR	MSE
Exisiting Method	34.44	23.56
Proposed Hybrid Technique	120.24	18.25

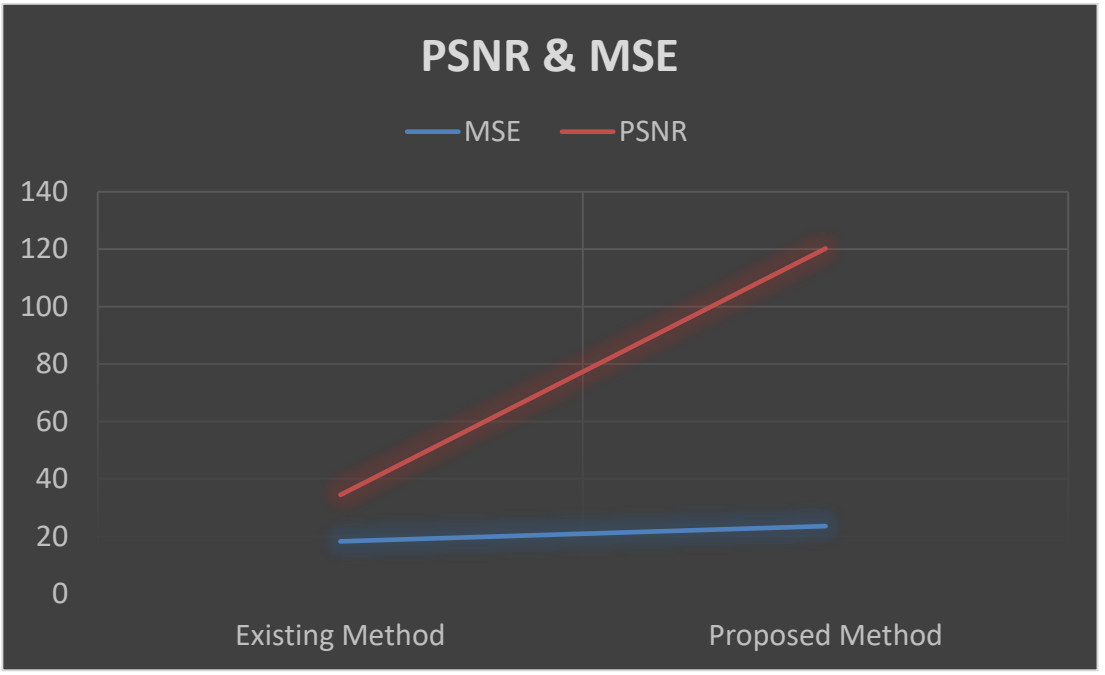


Fig 5.10 Graph showing Comparison of PSNR, MSE

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

A hybrid technique for encryption and decryption of the image by using cryptography and watermarking process was implemented here. This was done in order to achieve higher level of Security. Level wise we can choose the level of security to secure the image safely without any corruption. Using this technique, we can secure image, which cannot be shared with anyone else. It gives the owner and authorized receiver of the image access to the data using different level of security. In this paper we have calculated the MSE which we saw was better that non hybrid technique using only our algorithm. We achieved a better PSNR value. The entropy of image and the elapsed time for the complete process was also calculated. Further work needs to be done to optimize and compare our work with other hybrid algorithms to achieve better results in the future

CHAPTER 7

REFERENCE

1. Chowdhary, Charanjit Lal, et al. "Analytical study of hybrid techniques for image encryption and decryption." *Sensors* 20.18 (2020): 5162
2. Varun, B. V., et al. "Implementation of Encryption and Decryption Algorithms for Security of Mobile Devices." *2019 IEEE 19th International Conference on Communication Technology (ICCT)*. IEEE, 2019
3. Sanchez, Javier, et al. "Encryption techniques: A theoretical overview and future proposals." *2016 Third International Conference on democracy & eGovernment (ICEDEG)*. IEEE, 2016.
4. Prakash, G. L., Manish Prateek, and Inder Singh. "Data encryption and decryption algorithms using key rotations for data security in cloud system." *2014 International conference on signal propagation and computer technology (ICSPCT 2014)*. IEEE, 2014.
5. Agarwal, Priya, et al. "Authenticating cryptography over network in data." *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*. IEEE, 2019.
6. Hoekstra, Korn, and Pablo Rivas. "A review of machine learning and cryptography applications." *2020 International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE, 2020.
7. Jha, Deeksha Priya, Rashi Kohli, and Archana Gupta. "Proposed encryption algorithm for data security using matrix properties." 2016 International conference on innovation and challenges in cyber security (ICICCS-INBUSH). IEEE, 2016.
8. Sharma, Arvind K., and S. K. Mittal. "Cryptography & network security hash function applications, attacks and advances: A review." 2019 Third International Conference on Inventive Systems and Control (ICISC). IEEE, 2019.
9. Carried, Jorge Martinez, et al. "Cryptography for security in IoT." 2018 Fifth International Conference on Internet of Things: Systems, Management and Security. IEEE, 2018.
10. Rout ray, Sudhir K., et al. "Quantum cryptography for IoT: perspective." 2017 International Conference on IoT and Application (ICIOT). IEEE, 2017.
11. Yuan, Shahan, and Jintao Wu. "Deep learning for insider threat detection: Review, challenges and opportunities." Computers & Security 104 (2021): 102221.
12. Kaur, Mandeep, Surender Singh, and Manjit Kaur. "Computational image encryption techniques: a comprehensive review." *Mathematical Problems in Engineering* 2021 (2021): 1-17.

13. Kanagarathinam, DI George, and J. Sai Geetha. "Image encryption and decryption in public key cryptography based on MR." *2015 International Conference on Computing and Communications Technologies (ICCCT)*. IEEE, 2018

14. Kaur, Manjit, and Vijay Kumar. "A comprehensive review on image encryption techniques." *Archives of Computational Methods in Engineering* 27 (2020): 15-43.

15. Pan, Hailan, Yongmei Lei, and Chen Jian. "Research on digital image encryption algorithm based on double logistic chaotic map." *EURASIP Journal on Image and Video Processing* 2018.1 (2018)

16. Oad, Ambika, Himanshu Yadav, and Anurag Jain. "A review: image encryption techniques and its terminologies." *International Journal of Engineering and Advanced Technology (IJEAT) ISSN* (2014): 2249-8958.

17. Singh, Laiphrakpam Dolendro, and Khumanthem Manglem Singh. "Image encryption using elliptic curve cryptography." *Procedia Computer Science* 54 (2015): 472-481.

18. Astya, ParmaNand, Bhairvee Singh, and Divyanshu Chauhan. "Image encryption and decryption using elliptic curve cryptography." *Int. J. Adv. Res. Sci. Eng* 29.3-10 (2014): 198-205.

19. Shukla, Ashutosh, Jay Shah, and Nikhil Prabhu. "Image encryption using elliptic curve cryptography." *International Journal of Students' Research in Technology & Management* 1.2 (2015): 115-117.

20. Habek, Muhammed, et al. "Digital Image Encryption Using Elliptic Curve Cryptography: A Review." *2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*. IEEE, 2022.

21. Arab, Alireza, Mohammad Javad Rostami, and Behnam Ghavami. "An image encryption method based on chaos system and AES algorithm." *The Journal of Supercomputing* 75 (2019): 6663-6682.

22. Zhang, Yong, and Yingjun Tang. "A plaintext-related image encryption algorithm based on chaos." *Multimedia Tools and Applications* 77.6 (2018): 6647-6669.

23. Xiao, Di, Xiaofeng Liao, and Pengcheng Wei. "Analysis and improvement of a chaos-based image encryption algorithm." *Chaos, Solitons & Fractals* 40.5 (2009): 2191-2199.

24. "Image encryption and decryption using AES algorithm." *International Journal of Electronics and Communication Engineering & Technology* (2015): 23-29.e encryption algorithm." *Chaos, Solitons & Fractals* 40.5 (2009): 2191-2199.

25. Singh, Amandeep, Praveen Agarwal, and Mehar Chand. "Image encryption and analysis using dynamic AES." *2019 5th International Conference on Optimization and Applications (ICOA)*. IEEE, 2019.

CHAPTER 8

APPENDIX

Coding

```
import os
import sys
import urllib.request
from src.new_enc import enc
from src.new_dec import dec
from werkzeug.utils import
secure_filename
from flask import Flask, flash,
request, redirect, url_for,
render_template

UPLOAD_FOLDER = 'src/'

app = Flask(__name__)
app.secret_key = "secret key"
app.config['UPLOAD_FOLDER'] =
UPLOAD_FOLDER
ALLOWED_EXTENSIONS =
set(['png', 'jpg', 'jpeg', 'gif', 'txt'])

def allowed_file(filename):
        return '.' in filename and
filename.rsplit('.', 1)[1].lower() in
ALLOWED_EXTENSIONS

@app.route("/")
def index_page():
   return
render_template('index.html')

@app.route("/output/")
def output_page():
   return
render_template('output.html')

@app.route("/uploadenc/")
def uploadenc_page():
   return
render_template('uploadenc.html')

@app.route("/uploaddec/")
def uploaddec_page():
```

```
    return
render_template('uploaddec.html')

@app.route('/uploadenc/',
methods=['POST'])
def upload_image():
        if 'file' not in request.files:
                flash('No file part')
                return
redirect(request.url)
        file = request.files['file']
        if file.filename == "flash('No
image selected for uploading')
                return
redirect(request.url)
        if file and
allowed_file(file.filename):
                filename =
secure_filename(file.filename)
        file.save(os.path.join(app.con
fig['UPLOAD_FOLDER'],
"image.jpg"))
                print('upload_image
filename: ' + filename)
                enc()
                return
render_template('index.html')
        else:
                flash('Allowed image
types are -> png, jpg, jpeg, gif')
                return
redirect(request.url)
@app.route('/uploaddec/',methods=['
POST'])
def upload_file():
        if 'file' not in request.files:
                flash('No file part')
                return
redirect(request.url)
        file = request.files['file']
        if file.filename == '':
                flash('No image
selected for uploading')
                return
redirect(request.url)
        if file and
allowed_file(file.filename):
                filename =
secure_filename(file.filename)
```

```
            file.save(os.path.join(app.con
fig['UPLOAD_FOLDER'],
"encrypted.txt"))
                    print('upload_image
filename: ' + filename)
                    dec()
                    return
render_template('index.html')
            else:
                    flash('Allowed image
types are -> png, jpg, jpeg, gif')
                    return
redirect(request.url)

if __name__ == "__main__":
    app.run(debug=True)
```

AESCipher

```
import base64
import hashlib
from Crypto import Random
from Crypto.Cipher import AES

class AESCipher(object):

    def __init__(self, key):
        self.bs = 32
        self.key =
hashlib.sha256(key.encode()).digest(
)

    def encrypt(self, raw):
        raw = self._pad(raw)
        iv =
Random.new().read(AES.block_size
)
        cipher = AES.new(self.key,
AES.MODE_CBC, iv)
        return base64.b64encode(iv +
cipher.encrypt(raw))

    def decrypt(self, enc):
        enc = base64.b64decode(enc)
        iv = enc[:AES.block_size]
        cipher = AES.new(self.key,
AES.MODE_CBC, iv)
        return
self._unpad(cipher.decrypt(enc[AES.
```

```
block_size:])).decode('utf-8')

    def _pad(self, s):
        return s + (self.bs - len(s) %
self.bs) * chr(self.bs - len(s) %
self.bs).encode('utf-8')

    @staticmethod
    def _unpad(s):
        return s[:-ord(s[len(s)-1:])
```

Decrypted

```
import base64
import hashlib
from PIL import Image
from random import randint
from src.AESCipher import
AESCipher

class Decrypter:
    def __init__(self, cipher):
        self.cipher = cipher

    def decrypt_image(self,k):
        key = k
        cipher = self.cipher
        aes = AESCipher(key)
        base64_decoded =
aes.decrypt(cipher)
        fh =
open("decryptedImage.png", "wb")

fh.write(base64.b64decode(base64_d
ecoded))
        fh.close()
        return
(base64.b64decode(base64_decoded)
)
```

Encryption

```
import base64
import hashlib
from PIL import Image
from random import randint
from src.AESCipher import
AESCipher

class Encrypter:
    def __init__(self, text,key):
        self.text = text
        self.key =  key

  def encrypt_image(self):
        aes = AESCipher(self.key)
        cipher = aes.encrypt(self.text)
        return cipher
```

DEC

```
import os
import io
import sys
import base64
from PIL import Image as Img
from Crypto.Cipher import AES
from PIL import ImageTk as ImgTk
from src.Encrypter import Encrypter
from src.Decrypter import Decrypter

def dec():
    myKey = "1234"
    file  = "./src/encrypted.txt"
    text=open(file).read()
    cipher= text.encode('utf-8')
    x = Decrypter(cipher)
    image=x.decrypt_image(myKey)
    print(type(image))
    imag =
Img.open(io.BytesIO(image))

imag.save('./src/decryptedimage.png'
)
```

ENC

```
import os
import sys
import base64
from PIL import Image as Img
from Crypto.Cipher import AES
from PIL import ImageTk as ImgTk
from src.Encrypter import Encrypter
from src.Decrypter import Decrypter

def enc():
    with open("./src/image.jpg", "rb")
as imageFile:
        value =
base64.b64encode(imageFile.read())

    myKey = "1234"
    x = Encrypter(value, myKey)
    cipher = x.encrypt_image()
    fh = open("./src/cipher.txt", "wb")
    fh.write(cipher)
    fh.close()
```

ECC + AES hybrid encryption

```
from tinyec import registry
from Crypto.Cipher import AES
import hashlib, secrets, binascii

def encrypt_AES_GCM(msg,
secretKey):
    aesCipher = AES.new(secretKey,
AES.MODE_GCM)
    ciphertext, authTag =
aesCipher.encrypt_and_digest(msg)
    return (ciphertext,
aesCipher.nonce, authTag)

def decrypt_AES_GCM(ciphertext,
nonce, authTag, secretKey):
    aesCipher = AES.new(secretKey,
AES.MODE_GCM, nonce)
    plaintext =
aesCipher.decrypt_and_verify(cipher
text, authTag)
    return plaintext
```

```
def
ecc_point_to_256_bit_key(point):
    sha =
hashlib.sha256(int.to_bytes(point.x,
32, 'big'))
    sha.update(int.to_bytes(point.y,
32, 'big'))
    return sha.digest()

curve =
registry.get_curve('brainpoolP256r1')

def encrypt_ECC(msg, pubKey):
    ciphertextPrivKey =
secrets.randbelow(curve.field.n)
    sharedECCKey =
ciphertextPrivKey * pubKey
    secretKey =
ecc_point_to_256_bit_key(sharedEC
CKey)
    ciphertext, nonce, authTag =
encrypt_AES_GCM(msg, secretKey)
    ciphertextPubKey =
ciphertextPrivKey * curve.g
    return (ciphertext, nonce, authTag,
ciphertextPubKey)

def decrypt_ECC(encryptedMsg,
privKey):
    (ciphertext, nonce, authTag,
ciphertextPubKey) = encryptedMsg
    sharedECCKey = privKey *
ciphertextPubKey
    secretKey =
ecc_point_to_256_bit_key(sharedEC
CKey)
    plaintext =
decrypt_AES_GCM(ciphertext,
nonce, authTag, secretKey)
    return plaintext

msg = b'Text to be encrypted by
ECC public key and ' \
      b'decrypted by its corresponding
ECC private key'
print("original msg:", msg)
privKey =
secrets.randbelow(curve.field.n)
pubKey = privKey * curve.g
```

```
encryptedMsg = encrypt_ECC(msg,
pubKey)
encryptedMsgObj = {
    'ciphertext':
binascii.hexlify(encryptedMsg[0]),
    'nonce':
binascii.hexlify(encryptedMsg[1]),
    'authTag':
binascii.hexlify(encryptedMsg[2]),
    'ciphertextPubKey':
hex(encryptedMsg[3].x) +
hex(encryptedMsg[3].y % 2)[2:]
}
print("encrypted msg:",
encryptedMsgObj)

decryptedMsg =
decrypt_ECC(encryptedMsg,
privKey)
print("decrypted msg:",
decryptedMsg;)
```

RSA Key

```
from Crypto.PublicKey import RSA
from Crypto.Cipher import
PKCS1_OAEP
import binascii

keyPair = RSA.generate(3072)

pubKey = keyPair.publickey()
print(f"Public key:
(n={hex(pubKey.n)},
e={hex(pubKey.e)})")
pubKeyPEM = pubKey.exportKey()
print(pubKeyPEM.decode('ascii'))

print(f"Private key:
(n={hex(pubKey.n)},
d={hex(keyPair.d)})")
privKeyPEM = keyPair.exportKey()
print(privKeyPEM.decode('ascii'))
```