Project Report

on

COMPARATIVE ANALYSIS OF AES AND RSA ENCRYPTED LSB IMAGE STEGANOGRAPHY

Submitted in partial fulfilment of the requirements for the degree of

BACHELOR OF SCIENCE

in

MATHEMATICS

by

HAJRA NAUSHAD (AB23AMAT016)

Under the Supervision of

Dr. Elizabeth Reshma M T



DEPARTMENT OF MATHEMATICS AND STATISTICS

ST. TERESA'S COLLEGE (AUTONOMOUS),

ERNAKULAM, KOCHI- 682011

MARCH 2026

ST. TERESA'S COLLEGE (AUTONOMOUS), ERNAKULAM

CERTIFICATE

This is to certify that the dissertation entitled **"Comparative Analysis of AES and RSA Encrypted LSB Image Steganography"** is a bonafide record of the work carried out by **Ms. HAJRA NAUSHAD** under my guidance, in partial fulfilment of the requirements for the award of the degree of Bachelor of Science in Mathematics at St. Teresa's College (Autonomous), Ernakulam, affiliated to Mahatma Gandhi University, Kottayam. No part of this work has been submitted for the award of any other degree elsewhere.

Date: 30 September 2025
Place: Ernakulam

ST. TERESA'S COLLEGE (AUTONOMOUS) DEPARTMENT OF MATHEMATICS AND STATISTICS ERNAKULAM

Dr. Elizabeth Reshma M T
Assistant Professor and Head
Department of Mathematics,
St. Teresa's College (Autonomous),
Ernakulam

Dr. Elizabeth Reshma M T
Assistant Professor and Head
Department of Mathematics,
St. Teresa's College (Autonomous),
Ernakulam

External Examiners:

1:..... 2016/26

2:.....

Dr. SREEJA K.U. (PEN:716628)
Assistant Professor
Department of Mathematics
Maharaja's College
Ernakulam - 682 011

DECLARATION

I hereby declare that the work presented in this project is based on my original work carried out under the guidance of Dr. Elizabeth Reshma M T, Assistant Professor, Department of Mathematics, St. Teresa's College (Autonomous), Ernakulam, and has not been included in any other project submitted previously for the award of any degree.

Date: 30 September 2025

Place: Ernakulam

HAJRA NAUSHAD (AB23AMAT016)

ACKNOWLEDGEMENT

I extend my deepest gratitude to Dr. Elizabeth Reshma M T, Assistant Professor in the Department of Mathematics at St. Teresa's College, for her invaluable contribution as the coordinator of my project. Her guidance and support have been instrumental in facilitating the completion of all stages of my project. I am also grateful to the Department of Mathematics and Statistics for providing me with the necessary resources and environment to successfully complete this project. I would like to sincerely thank my group members for their collaboration, insightful contributions, and constant encouragement throughout this project. I am also grateful to my family and well-wishers for their constant support and motivation. Lastly, I wish to acknowledge the divine guidance that has illuminated my path and provided strength and inspiration amidst challenges. Each day, I have felt Your guidance.

Date: 30 September 2025

Place: Ernakulam

HAJRA NAUSHAD (AB23AMAT016)

CONTENTS

CERTIFICATE……...……..i
DECLARATION…….ii
ACKNOWLEDGMENT…….iii
CONTENTS……..……..iv

1. INTRODUCTION AND PRELIMINARIES…….1-10
1.1 BACKGROUND AND MOTIVATION……1
1.2 PRELIMINARIES…….2
1.3 OBJECTIVE, SCOPE AND CHAPTER OVERVIEW…….9

2. STEGANOGRAPHY…….11-23
2.1 ORIGIN, EVOLUTION AND SIGNIFICANCE……11
2.2 TYPES OF STEGANOGRAPHY…….....14
2.3 TECHNIQUES OF IMAGE STEGANOGRAPHY……17

3. QUALITY METRICS……24-32
3.1 INTRODUCTION AND EVALUATION AXES……24
3.2 PEAK SIGNAL-TO-NOISE RATIO AND
MEAN SQUARED ERROR……25
3.3 STRUCTURAL SIMILARITY INDEX MEASURE……..……27
3.4 COMPARISON OF IMAGE QUALITY METRICS…….……..30

4. COMPARATIVE ANALYSIS OF CRYPTOGRAPHY EMBEDDED IMAGE STEGANOGRAPHY USING QUALITY METRICS…….33-54
4.1 METHODOLOGY AND SETUP……33
4.2 QUALITY ASSESSMENT OF STEGO IMAGE……35
4.3 SUMMARY AND COMPARATIVE INSIGHTS……51

REFERENCES……53-54

CHAPTER 1: INTRODUCTION AND PRELIMINARIES

1.1 BACKGROUND AND MOTIVATION

As digital technologies continue to advance, the protection of sensitive information has emerged as a major challenge. Open networks expose transmitted data to potential threats such as eavesdropping, alteration, and illegal intrusion. Traditional security mechanisms such as encryption protect the content of the message but can still reveal the presence of confidential communication. The escalation of cybercrime and information-related risks calls for the adoption of advanced and highly reliable security technologies.

In today's interconnected world, digital communication is the backbone of personal, financial, and governmental operations. Confidential communication upholds the principles of data privacy, integrity, and authenticity during transmission. Without proper safeguards, private messages, banking transactions, and classified data can easily be intercepted or manipulated. Hence, creating methods that ensure both safety and confidentiality is crucial for building trust in digital platforms and avoiding the misuse of information.

Steganography is a powerful technique for data protection because it hides the very presence of secret information inside harmless-looking media such as images, audio, or video. Unlike cryptography, which only makes a message unreadable, steganography ensures that outsiders don't even suspect a message exists. This concealment greatly reduces the chances of interception and attack, making it a valuable tool for secure communication. When steganography is combined with cryptography known as crypto-steganography it provides both invisibility and unreadability, offering a stronger shield against cyber threats.
The selection of this project is motivated by the rising demand for secure and discreet communication in the digital era. While encryption protects confidentiality, it can also reveal the presence of sensitive information, attracting potential attacks. Alternatively, steganography hides information within regular communication channels, making it significantly more difficult to uncover. Specifically, this project focuses on a comparative analysis of cryptography-embedded LSB (Least Significant Bit) image steganography using quality metrics to evaluate and optimize the effectiveness of different approaches. The objective of this project is to investigate the combination of steganography and cryptographic techniques to enhance information security and tackle the rising challenges in data protection.

1.2 PRELIMINARIES

1.2.1 Information Security:

Information Security better known as InfoSec is basically the protection of information especially important information such as financial information, personal information, sensitive information or confidential information and also the information systems from unauthorised access, illegal use, exposure, alteration, disruption or even destruction. The information can be digital or physical. It spans from the details of our birth to the pictures we post in our social media accounts [9].

Information security is a critical aspect of today's digital world because it protects valuable data from unauthorized access, misuse, or alteration. It ensures confidentiality by keeping sensitive information private and preventing exposure to unauthorized individuals. It maintains integrity by making sure that data remains accurate and reliable, without being tampered with or corrupted. It also guarantees availability by ensuring that information and digital services are accessible to authorized users whenever they are needed.Beyond these core functions, information security plays a major role in helping organizations comply with legal and regulatory requirements, especially in sensitive sectors such as banking, healthcare, and defense. It also builds and preserves trust, both for individuals who rely on their personal information being safe and for businesses that must safeguard client and operational data. Without strong information security, the digital systems that modern life depends on would be vulnerable to privacy breaches, fraud, financial loss, and disruption of essential services [9].

The foundation of information security is built on a set of guiding principles that ensure data is protected from threats while remaining useful to authorized users. The most widely recognized framework is the **CIA** triad, which includes confidentiality, integrity, and availability. **Confidentiality** ensures that sensitive information is accessible only to authorized individuals. It prevents unauthorized access and protects privacy. Techniques such as encryption, authentication, and access control systems are used to enforce confidentiality. **Integrity** guarantees that data remains accurate, consistent, and trustworthy. Information must not be altered, deleted, or corrupted in unauthorized ways. Mechanisms like hashing, checksums, and version control help maintain data integrity. **Availability** ensures that information and services are accessible when needed by authorized users. It protects systems against disruptions caused by attacks, failures, or natural disasters. Methods such as backups, redundancy, and disaster recovery planning are applied to maintain availability [9].

In addition to the CIA triad, modern information security also emphasizes extended principles. Authentication verifies the identity of users or systems, while authorization ensures they access only permitted resources. Non-repudiation prevents parties from denying their actions, often using digital signatures. Accountability tracks user activities through logging and monitoring. Together, these principles strengthen information security, keeping

data protected, reliable, and accessible in the face of growing cyber threats. Information Security Systems can be broadly divided into two categories and they are **Cryptography** and **Information Hiding**. **Steganography** and **Watermarking** comes under the category of Information Hiding [12].

1.2.2 Cryptography:

Cryptography which came from the two ancient Greek words, kryptós which means "hidden" or "secret" and graphein which means "to write" is the art and science of securing information by transforming it into an unreadable format, ensuring confidentiality, integrity, authentication, and non-repudiation. Its origins trace back thousands of years to ancient civilizations. For example, the Spartans used a device called a scytale, a cylindrical tool around which they wrapped a strip of parchment to encode messages. Similarly, Julius Caesar employed a simple substitution cipher, now called the Caesar cipher, in which each letter in the message was shifted by a fixed number of positions. During the Renaissance, cryptographers developed polyalphabetic ciphers, such as the Vigenère cipher, which used multiple substitution alphabets to make decryption more complex for attackers. These methods significantly increased security but were still vulnerable to modern analytical techniques. With the invention of computers in the 20th century, modern cryptography emerged. Algorithms became more mathematically sophisticated, introducing symmetric and asymmetric key systems, digital signatures, and hash functions to secure digital communications. These developments allowed secure communication over public channels like the internet, which was not possible with classical ciphers. [1]

Cryptography is essential in today's digital world because it protects sensitive information from unauthorized access and cyber threats. Its main goals are **Confidentiality**, **Integrity**, **Authentication** and **Non-repudiation**. Confidentiality ensures that information is accessible only to authorized parties, integrity ensures that data is not altered during transmission, authentication verifies the identity of the sender and receiver and non-repudiation prevents entities from denying their actions, such as sending a message or performing a transaction [2]. Applications of cryptography span financial transactions, government communications, healthcare records, online banking, and e-commerce, making it a backbone of modern information security systems. [1]

The main types of cryptography are **Symmetric Cryptography**, **Asymmetric Cryptography** and **Hash Functions**. In symmetric cryptography, the same key is used for both encryption and decryption. This method is efficient for large data sets because encryption and decryption processes are fast. The main challenge is key distribution: both sender and receiver must securely exchange the key without interception. Few examples of symmetric cryptography are **Advanced Encryption Standard** (AES) which is widely used for secure data storage and online transactions and **Data Encryption Standard** (DES) which is an older standard, largely replaced by AES due to vulnerabilities. Asymmetric cryptography, also called public-key cryptography, uses two keys: a public key for encryption

and a private key for decryption. The public key can be shared openly, while the private key is kept secret. This method solves the key distribution problem inherent in symmetric cryptography. Few examples of asymmetric cryptography are **Rivest-Shamir-Adleman (**RSA) which is used for secure email communication, SSL/TLS encryption, and digital signatures and **Elliptic Curve Cryptography** (ECC) which offers strong security with smaller keys, improving efficiency for mobile and IoT devices. A hash function converts input data of any size into a fixed-size string of characters, known as a hash. Hash functions are one-way, meaning the original data cannot be derived from the hash. They ensure data integrity and are essential in password storage, digital signatures, and blockchain technology. SHA-256 is a widely used cryptographic hash function that produces a 256-bit hash [1].

Cryptography is applied across a wide range of fields to ensure the security and integrity of digital information. In secure communication, it encrypts emails, messages, and voice calls so that only the intended recipient can access the content [1]. Digital signatures are used to verify the authenticity and integrity of digital documents, preventing forgery or tampering. Cryptography also protects sensitive data stored in databases, such as medical records and financial information, from unauthorized access. Additionally, it underpins authentication protocols, verifying user identities for secure logins on websites, mobile apps, and banking systems. In emerging technologies like blockchain, cryptography secures cryptocurrency transactions, smart contracts, and the integrity of distributed ledgers. While cryptography is crucial, it faces several challenges. Key management, including generating, distributing, and storing keys securely, can be difficult. Computational overhead is another concern, as asymmetric cryptography requires significant processing power, which can slow down systems. Quantum computing also poses a threat, since quantum computers could potentially break current cryptographic algorithms, prompting research into quantum-resistant methods. Additionally, poor implementation of cryptographic algorithms can result in security breaches, even if the algorithm itself is mathematically strong [1]. For this project, we are using AES as the symmetric key algorithm and RSA as the asymmetric key algorithm.

1.2.2.1 Advanced Encryption Standard (AES) Algorithm:

The **Advanced Encryption Standard (AES)** is one of the most widely used cryptographic algorithms in modern information security. It was established by the National Institute of Standards and Technology (NIST) in 2001 as the official replacement for the older Data Encryption Standard (DES), which had become vulnerable to brute-force attacks due to its relatively small key size. AES is based on the Rijndael cipher, designed by Belgian cryptographers Vincent Rijmen and Joan Daemen, who won the AES competition held by NIST in 1997. AES is a symmetric-key block cipher, which means the same secret key is used for both encryption and decryption. It operates on fixed-size blocks of 128 bits (16 bytes) and supports three different key lengths: 128, 192, or 256 bits. The number of rounds used in the encryption process depends on the key length: 10 rounds for AES-128, 12 rounds for AES-192, and 14 rounds for AES-256 [3]. Each round consists of a series of

transformations that substitute, permute, mix, and combine the input data with round keys derived from the original key.

The AES encryption process takes a readable message (plaintext) and turns it into scrambled, unreadable data (ciphertext). To do this, the message is first broken down into small chunks—specifically, a block of 128 bits—which is then arranged into a 4×4 grid called the state. This grid goes through a series of steps, called rounds, where the data is gradually mixed, substituted, and transformed until it becomes completely unrecognizable [3]. In the encryption process, every tiny piece of data, or byte, in the grid is replaced with another byte using a pre-defined lookup table called the S-box [11]. This substitution is similar to swapping letters in a secret codebook, where each input has a corresponding substitute. This step adds complexity and makes it very difficult to trace any direct link between the original message and the encrypted output. Next, the rows of the state matrix are shifted cyclically by different offsets. The first row remains unchanged, the second row is shifted one position to the left, the third row by two positions, and the fourth row by three. This operation increases diffusion by mixing data across columns [11]. After that, each column of the state is multiplied by a fixed polynomial in a finite field. This step mixes the data within each column and further enhances diffusion. Finally, the state is combined with a round key derived from the original key through a key scheduling algorithm. These steps are repeated for multiple rounds depending on the key size, except for the final round, which omits the column mixing. The result after the last round is the ciphertext.

The **encryption workflow** for AES cryptography is given as follows, describing the step-by-step process of securing the data.

Step 1: Message Selection

The plaintext selected for encryption in this study was:

“The picture hides a secret message”.

This message served as the test input to demonstrate the encryption and subsequent embedding process.

Step 2: Key and Initialization Vector Generation

AES requires both a secret key and an Initialization Vector (IV) to ensure secure encryption in Cipher Block Chaining (CBC) mode. The following randomly generated values were used in this project:

Key (Base64): TMXO1P2rSPj2bbR9J3Hz5A==

IV (Base64): Bs6z/sWcZBJLiJn4dvGSoA==

The secret key ensures that only authorized users can perform decryption, while the IV introduces randomness so that even identical plaintext messages produce different ciphertext outputs.

Step 3: Encryption of the Message

The message was encrypted using AES in CBC mode, which combines each plaintext block with the previous ciphertext block before encryption. This mode strengthens security by preventing pattern repetition. The output of this process was the ciphertext:

“seyUoJJ5v0w4jwkuREX08AtlUEmEGhHt19PIuFZZTJ0sLJ5tCbYyo8q/eNct1mWO”

The ciphertext appeared as a random sequence of characters, ensuring that the original message was concealed effectively[3].

Step 4: Conversion to Binary

The ciphertext, which was 48 bytes in size, was converted into binary form, producing a sequence of 384 bits. This conversion was essential because the Least Significant Bit (LSB) steganography technique used in the next phase operates at the bit level.

1.2.2.2 Rivest-Shamir-Adleman (RSA) Algorithm:

The **Rivest–Shamir–Adleman (RSA)** algorithm is a type of asymmetric cryptography, also known as public-key cryptography (PKC). It was developed in 1977 by Ron Rivest, Adi Shamir, and Leonard Adleman at the Massachusetts Institute of Technology, and its name is derived from the initials of its inventors [1]. RSA was the first widely published and practical public-key cryptosystem. It solved the problem of securely sharing a secret key before sending and receiving messages by introducing the use of a public key and a private key. The public key can be shared openly, while the private key remains secret, which makes communication more secure between the sender and receiver. The security of RSA is based on the mathematical difficulty of factoring large integers and the use of modular arithmetic [3].

The RSA algorithm works in three main stages: Key Generation and Distribution, Encryption, and Decryption. In the first stage, a pair of keys—public and private—are generated, with the public key being shared and the private key kept confidential. In the encryption stage, the plaintext message is converted into ciphertext using the recipient's public key. Finally, in the decryption stage, the recipient uses their private key to transform the ciphertext back into the original plaintext, ensuring that only the intended recipient can read the message [3].

Stage 1: Generation and Distribution of Keys

This is the most important stage of the RSA algorithm.

Step1: Select two large prime numbers and denote them as p and q.

Step 2: Calculate n, where $n = p \times q$ (product of the two selected prime numbers)

Step 3: Calculate the Euler Totient Function $\Phi(n)$, which is given by $\Phi(n) = \Phi(p * q) = \Phi(p) * \Phi(q) = (p - 1) * (q - 1)$

Step 4: Select the encryption exponent, denoted as e. It is chosen such that the following conditions are satisfied:$1 < e < \Phi(n)$ and $\gcd(e, \Phi(n)) = 1$; that is, e must be relatively prime to $\Phi(n)$.

Step 5: Calculate the decryption exponent, denoted as d. It is computed such that $(d * e) \equiv 1 \bmod \Phi(n)$, or equivalently $d \equiv e^{-1} \bmod \Phi(n)$. That is, d is the modular multiplicative inverse of e modulo $\Phi(n)$. The modular multiplicative inverse can be calculated using several methods, including the Extended Euclidean Algorithm, which is the most common, Fermat's Little Theorem (when the modulus is prime), and brute force search [1]

Step 5: The resulting public key is given by Public Key = (n , e), and the resulting private key is given by Private Key = (n , d).
Step 6: Distribute the public and private keys appropriately [3].

Stage 2: Encryption
Let M denote the message to be encrypted. To encrypt M, we first convert each letter of the message into its numerical representation using ASCII (American Standard Code for Information Interchange) or another suitable method. Next, we encrypt M into the ciphertext C using the generated public key and the formula C = M^e mod n, where C is the ciphertext, M is the message, e is the encryption exponent, and n is the modulus [3].

Stage 3: Decryption
For decrypting the ciphertext back into its original message , we use the formula
M = C^d mod n, where M is the message, C is the ciphertext, d is the decryption exponent and n is the modulus [3].

Now, let's look at an example to see how we encrypted the message 'The picture has a secret message' using RSA cryptography.

Stage 1: Key Generation
The large prime numbers selected for this example are p = 101 and q = 113. Therefore, n = 101 × 113 = 11413 and Φ(n) = (101 - 1) × (113 - 1) = 100 × 112 = 11200. We chose e = 17 since 1 < 17 < 11200 and gcd (17, 11200) = 1. Next, we find d such that d ≡ e^−1 mod Φ(n).
Applying the extended euclidean algorithm:
11200 = 17 × 658 + 14
17 = 14 × 1 + 3
14 = 3 × 4 + 2
3 = 2 × 1 + 1
2 = 1 × 2 + 0
Back-substituting to express 1 as a combination of 17 and 11200:
1 = 3 - 2 × 1
2 = 14 - 3 × 4 => 1 = 5 × 3 - 14
3 = 17 - 14 × 1 => 1 = 5 × 17 - 6 × 14
14 = 11200 - 17 × 658 => 1 = 3953 × 17 - 6 × 11200
This, 17 × 3953 ≡ 1 (mod 11200), giving the modular inverse d = 3953
The Public and Private Keys are:
Public Key = (11413 , 17)
Private Key = (11413 , 3953)

Stage 2: Encryption
The encryption formula used is C = M^17 mod 11413. The message to be encrypted is "The picture has a secret message".

Each character of the message is first converted into its numerical representation using the ASCII code (decimal values):

	A	B	C
1	Character	ASCII Decimal	8-bit Binary
2	T	84	01010100
3	h	104	01101000
4	e	101	01100101
5	(space)	32	00100000
6	p	112	01110000
7	i	105	01101001
8	c	99	01100011
9	t	116	01110100
10	u	117	01110101
11	r	114	01110010
12	e	101	01100101
13	(space)	32	00100000
14	h	104	01101000
15	a	97	01100001
16	s	115	01110011
17	(space)	32	00100000
18	a	97	01100001
19	(space)	32	00100000
20	s	115	01110011
21	e	101	01100101
22	c	99	01100011
23	r	114	01110010
24	e	101	01100101
25	t	116	01110100
26	(space)	32	00100000
27	m	109	01101101
28	e	101	01100101
29	s	115	01110011
30	s	115	01110011
31	a	97	01100001
32	g	103	01100111
33	e	101	01100101

Next, each numeric value is encrypted using the formula.

	A	B	C
1	Character	Equation	RSA Encrypted (C)
2	T	C = 84^17 mod 11413	794
3	h	C = 104^17 mod 11413	2573
4	e	C = 101^17 mod 11413	10706
5	(space)	C = 32^17 mod 11413	567
6	p	C = 112^17 mod 11413	6553
7	i	C = 105^17 mod 11413	9665
8	c	C = 99^17 mod 11413	5682
9	t	C = 116^17 mod 11413	3198
10	u	C = 117^17 mod 11413	11364
11	r	C = 114^17 mod 11413	4521
12	e	C = 101^17 mod 11413	10706
13	(space)	C = 32^17 mod 11413	567
14	h	C = 104^17 mod 11413	2573
15	a	C = 97^17 mod 11413	5283
16	s	C = 115^17 mod 11413	2026
17	(space)	C = 32^17 mod 11413	567
18	a	C = 97^17 mod 11413	5283
19	(space)	C = 32^17 mod 11413	567
20	s	C = 115^17 mod 11413	2026
21	e	C = 101^17 mod 11413	10706
22	c	C = 99^17 mod 11413	5682
23	r	C = 114^17 mod 11413	4521
24	e	C = 101^17 mod 11413	10706
25	t	C = 116^17 mod 11413	3198
26	(space)	C = 32^17 mod 11413	567
27	m	C = 109^17 mod 11413	2422
28	e	C = 101^17 mod 11413	10706
29	s	C = 115^17 mod 11413	2026
30	s	C = 115^17 mod 11413	2026
31	a	C = 97^17 mod 11413	5283
32	g	C = 103^17 mod 11413	10579
33	e	C = 101^17 mod 11413	10706

The complete numeric ciphertext is:
[794, 2573, 10706, 567, 6553, 9665, 5682, 3198, 11364, 4521, 10706, 567, 2573, 5283, 2026, 567, 5283, 567, 2026, 10706, 5682, 4521, 10706, 3198, 567, 2422, 10706, 2026, 2026, 5283, 10579, 10706].
For easier handling and embedding into an image using steganography, the numeric ciphertext is converted into a Base64-encoded form.
Base64-encoded ciphertext:
AxoKDSnSAjcZmSXBFjIMfixkEakp0gI3Cg0UowfqAjcUowI3B+op0hYyEakp0gx+AjcJdinSB+oH6hSjKVMp0g==
Base64 encoding allows the numeric ciphertext to be represented in a text-safe format, making it easier to embed in images or transmit. Alternatively, the numeric ciphertext could also be directly hidden in the image by converting each number into its binary form.

1.3 OBJECTIVE, SCOPE AND CHAPTER OVERVIEW

The main objective of this project is to securely conceal encrypted information within digital images using the Least Significant Bit (LSB) steganography technique, and to evaluate the quality of the resulting stego images using established performance metrics. To achieve this, advanced cryptographic methods such as Advanced Encryption Standard (AES) and Rivest–Shamir–Adleman (RSA) are integrated before embedding, thereby ensuring a dual layer of security—encryption for data confidentiality and steganography for concealment. Additionally, the project aims to analyze and compare the effectiveness of this approach across different RGB color channels (R, G, B, and combined RGB) by applying quality assessment metrics like Peak Signal-to-Noise Ratio (PSNR), Mean Squared Error (MSE), and Structural Similarity Index (SSIM).

The scope of this project is focused on several key areas. At the cryptographic layer, AES and RSA algorithms are implemented to secure the secret message prior to embedding. At the steganographic layer, LSB substitution is employed to embed the encrypted data into the RGB channels of cover images. The project also evaluates image quality and imperceptibility using metrics such as PSNR, MSE, and SSIM. A comparative study is conducted to observe how embedding in different color channels affects imperceptibility, taking into account both bright and dull images. Furthermore, the project explores potential applications of integrating cryptography and steganography in fields such as confidential communication, defense, journalism, and data privacy. Overall, this scope ensures that the project remains focused on both security and imperceptibility, while also addressing the trade-off between the two.

The following provides an overview of the chapters included in this project, summarizing their key content and focus areas. Chapter 1 provides an introduction to the project by covering four main components. The background explains the general area of study and its significance. The motivation highlights the reasons for selecting this topic, including its practical applications and the research gap it addresses. The preliminaries introduce the basic

concepts necessary to understand the project, and the objectives outline the main goals and intended outcomes. Chapter 2 focuses on steganography, tracing its origins from ancient techniques to modern digital methods. It reviews the different types, such as text, audio, video, and image-based steganography, with a special emphasis on image-based methods and the RGB model, which hides data within the red, green, and blue components of an image. Chapter 3 discusses the evaluation parameters for stego images. Mean Squared Error (MSE) quantifies the average difference between the original and stego image, indicating the level of distortion. Peak Signal-to-Noise Ratio (PSNR) measures the ratio of signal strength to noise, where higher values correspond to better image quality. Structural Similarity Index Measure (SSIM) evaluates perceptual similarity by comparing brightness, contrast, and structure. Together, these metrics provide both mathematical and visual insights into image quality. Chapter 4 presents the experimental study, detailing the tools, methods, and embedding process used to hide encrypted data within images. The quality assessment is performed using MSE, PSNR, and SSIM to evaluate distortion and clarity. A comparative analysis of different color channels demonstrates how embedding affects the red, green, and blue components differently. The chapter concludes with key findings, practical applications, and final remarks drawn from the analysis.

CHAPTER 2: STEGANOGRAPHY

2.1 ORIGIN, EVOLUTION AND SIGNIFICANCE

Steganography is the art and science of hiding information within another medium in such a way that the very existence of the hidden message remains concealed[12]. The term derives from the Greek words “steganos” (meaning “covered” or “hidden”) and “graphein” (meaning “to write”). Unlike cryptography, which makes a message unreadable to outsiders but still obvious in its encrypted form, steganography focuses on disguising the presence of communication itself[6].

In practice, steganography works by embedding secret data into an ordinary, non-secret medium known as the cover medium. Once the data is hidden, the result is called the stego object. The cover medium can take various forms, such as an image, audio file, video, text, or even network packets. For example, in image steganography, information may be concealed within the pixel values of a picture, making the alterations imperceptible to the human eye.

The primary objective of steganography is covert communication—ensuring that sensitive data can be transmitted without attracting suspicion. This makes it an effective tool not only in secure communication but also in areas such as digital watermarking, copyright protection, and authentication.

Steganography is often used in combination with cryptography to strengthen security. While cryptography protects the content of a message by converting it into ciphertext, steganography adds another layer by concealing the very fact that a message exists. Together, they form a robust approach to safeguarding information in the digital age.

The general model of steganography consists of three main components: a cover object, secret data, and embedding/extraction processes.At the sender side, the secret data is embedded into a cover object (which could be an image, audio, video, or text) using an embedding technique. This produces a stego object, which appears almost identical to the original cover, making the presence of the hidden information imperceptible.At the receiver side, the stego object is processed using an extraction technique to recover the hidden secret data.

Mathematically, this process can be represented as: Stego Object = Embedding (Cover Object, Secret Data) and Secret Data = Extraction (Stego Object)

In summary, steganography enables secure communication by concealing information within a cover medium[12]. The sender hides the secret data inside a cover object, and the receiver extracts it from the stego object, all while maintaining the appearance of the original cover.

2.1.1 Origin and History:

The term steganography originates from the Greek words “steganos” (meaning “covered” or “hidden”) and “graphein” (meaning “to write”). Thus, steganography literally translates to “covered writing.” It refers to the practice of hiding the existence of a message by concealing it within another medium. Unlike cryptography, which focuses on obscuring the meaning of a message, steganography conceals the very presence of communication.
The origins of steganography can be traced back to ancient civilizations, where people employed creative methods to secretly pass information[12]:

One of the earliest documented uses was described by the historian Herodotus (484–425 BC). Messages were tattooed onto the shaved scalp of a slave, and after the hair grew back, the slave was sent to deliver the message undetected. Another method involved writing messages on wooden tablets and covering them with wax to give the appearance of a blank surface.
The Romans often used invisible inks made from natural substances such as milk or fruit juice. The hidden text would only appear when exposed to heat, providing a simple but effective method of concealment.
During this period, secret communication was often carried out using coded letters, invisible ink, or hidden markings within religious texts and artwork. Steganography was used by spies and diplomats to transmit sensitive information without arousing suspicion.
Steganographic techniques evolved significantly during wartime. For instance, messages were hidden within microdots—tiny images or text reduced to the size of a period in a sentence, which could then be embedded in letters or printed materials. These methods allowed critical intelligence to be transferred while avoiding detection.
In modern times, the concept of steganography has moved from physical techniques to digital applications. With the advent of computers and multimedia technologies, information can now be hidden in digital images, audio, video, and even network traffic. The most common digital method is Least Significant Bit (LSB) insertion, where data is embedded in the least significant bits of pixels in an image file. This transition marks the evolution of steganography from simple physical tricks to highly sophisticated digital tools for secure communication.

2.1.2 Evolution:

Steganography has a long history that extends from ancient physical methods of secret communication to modern digital techniques supported by advanced computing. Over time, the techniques and mediums used have changed significantly, but the fundamental purpose—concealing the existence of information—has remained constant. The evolution of steganography can be understood in three broad phases: classical, pre-digital, and digital.

Classical steganography dates back to ancient civilizations, where creative methods were used to conceal secret messages. In 5th century BC Greece, Herodotus documented how messages were tattooed on the shaved scalps of messengers and hidden once their hair grew

back. Others wrote on wooden tablets and covered the writing with wax, giving the appearance of a blank surface. During the Roman Empire, invisible inks made from natural substances such as milk and fruit juice were used to hide information, with the hidden writing revealed only by applying heat. In the Medieval era, covert messages were often concealed within religious manuscripts, artworks, or written with special inks. Acrostics, where the first letters of lines spelled out hidden words, were also a common method of secret communication. These simple yet effective techniques laid the foundation for the evolution of covert communication across history[12].

Pre-digital steganography in the 19th and early 20th centuries introduced more sophisticated physical techniques, particularly during times of war[6]. One of the most notable methods was the use of microdots during World War II, where large amounts of information were reduced to the size of a typographical dot and embedded into letters or documents, enabling sensitive intelligence to pass undetected. Another advancement was photographic steganography, in which messages were hidden within altered photographs or developed using chemical processes that rendered them invisible to the naked eye. Espionage also relied heavily on steganographic practices, with spies using coded markings on letters, invisible inks activated by chemical developers, and messages concealed inside everyday objects.
This period marked a clear transition from basic physical concealment to more technical and scientific approaches, making steganography increasingly difficult to detect.

Digital steganography emerged with the rise of computers and the internet, enabling vast amounts of data to be hidden efficiently across multiple media formats. Image steganography became the most common method, using techniques like Least Significant Bit (LSB) insertion, where secret data is stored in the lower bits of image pixels without creating noticeable changes. Similarly, audio and video steganography conceal information within sound waves or video frames by modifying frequencies, phases, or pixel values. In addition, text steganography embeds secret data within digital text files by manipulating spacing, font variations, or hidden characters, while network steganography hides information within network traffic patterns, packet headers, or transmission timings, making it difficult to detect during communication.
Today, digital steganography is frequently combined with cryptography to enhance security. In this dual-layer approach, cryptography encrypts the data into ciphertext, while steganography conceals its very existence, providing stronger protection against interception or unauthorized access.

2.1.3 Significance:

Steganography is significant for secure communication, data protection, avoiding detection, digital watermarking, authentication, Military and Intelligent uses and for combining with cryptography[12].

Steganography enables secure communication by hiding the existence of a message, unlike cryptography, which only scrambles it, making the communication less suspicious. It allows sensitive information, such as personal data, financial records, or government documents, to be hidden inside harmless-looking files like images, audio, or video, preventing unauthorized access. In environments with strict surveillance or censorship, steganography helps share information secretly without attracting attention. It also supports digital watermarking, embedding hidden marks in digital content such as music, videos, or images, which helps verify ownership and protect copyrights.

Hidden information can be used for authentication, confirming whether a file has been tampered with and ensuring its integrity. Governments and defense agencies use steganography for military and intelligence purposes[3], sending covert instructions or data without alerting adversaries. When combined with cryptography, steganography not only secures the content but also hides its existence, making communication more robust.

Steganography is widely used in areas where hiding information is crucial. It enables secret communication between two parties without attracting attention, ensuring that the very existence of the message remains hidden. It is also used for copyright protection[11] and watermarking, embedding ownership information in images, videos, or audio to prevent piracy. In cybersecurity, steganography helps protect sensitive data during transmission over public networks, while in digital forensics and law enforcement, it assists investigators in recovering or identifying hidden evidence. Other applications include medical data protection, where patient records can be safely stored and shared within medical images, and military and defense, where it ensures confidential communication in hostile environments. Steganography is also employed for content authentication, embedding hidden information to detect any tampering, though it can be misused by hackers to hide malware or illicit data in images or videos.

Steganography is employed primarily because it conceals the existence of a message, rather than just its content, providing an extra layer of privacy and security, especially when combined with encryption[12]. It is useful for ownership verification through watermarking and can be robust enough to survive common image processing operations such as compression or resizing. Overall, steganography allows secret data transfer in a way that does not arouse suspicion, making it an effective tool for secure communication.

2.2 TYPES OF STEGANOGRAPHY

Steganography is broadly categorized into five types: text, audio, video, image, and network, each using a unique medium to conceal information.

Text steganography is a technique that involves hiding secret information within textual content by carefully modifying the structure of existing text, substituting or rearranging words, generating sequences of seemingly random characters, or utilizing context-free

grammars (CFG) to create new text that appears meaningful and coherent, thereby ensuring that the concealed information remains undetectable to casual observers. Text steganography can be categorized into three pieces, including format-based methods (FBM), random and statistical generation, and linguistics.[5] **Audio steganography** is a technique that involves concealing secret information within digital audio signals, which can include recordings, telephone calls, or audio streams from video conferences, by using the audio signal itself as a cover medium. This method takes advantage of the characteristics of the Human Auditory System (HAS), particularly its masking effect, which allows certain sounds to be imperceptible when overlaid by louder or more dominant audio components. By embedding information in ways that exploit this auditory masking, audio steganography ensures that the hidden data remains undetectable to casual listeners while preserving the natural quality and intelligibility of the original audio signal.[11] **Video steganography** is a technique used to hide any form of data or information within a digital video file, making it an effective means of covert communication. Because a video is composed of a sequence of individual frames, each containing numerous pixels and color channels, it provides a large and versatile medium for embedding hidden content without significantly altering the visual or auditory quality of the video. By strategically modifying certain frames, pixel values, or other attributes of the video, sensitive information can be concealed in a way that is difficult for unintended viewers or automated detection systems to notice, thereby ensuring the secrecy and integrity of the embedded data. **Network steganography** is a technique used to conceal secret information within data that is transmitted over computer networks. This form of steganography takes advantage of the structure and behavior of network communication protocols, often by exploiting unused, optional, or less monitored fields in widely used protocols such as TCP, IP, or HTTP. By carefully embedding data within these fields, network steganography allows information to be transmitted covertly across networks without drawing attention, making it difficult for unauthorized parties or monitoring systems to detect the hidden content while the normal flow of communication appears unaffected.

2.2.1 Image Steganography:

Image steganography is a widely used technique in the field of information security that allows secret information to be hidden inside a digital image in such a way that the presence of the hidden data remains unnoticed. Unlike cryptography, which only encrypts or scrambles the content of a message but makes it obvious that some confidential information exists, steganography hides the very existence of the message, thereby offering an additional layer of secrecy. A digital image can be defined as a two-dimensional representation of visual information that is stored in numerical form and processed with the help of computers. Every digital image is made up of very small units called pixels (short for picture elements), which are the smallest components of an image. Each pixel contains information about brightness and color, and when millions of these pixels are arranged in rows and columns, they together form the complete image we see on a screen. The size of an image is determined by the total

number of pixels it contains, usually expressed in terms of its resolution, such as 512×512 pixels or 1920×1080 pixels, which directly influences the level of detail and clarity in the image. Depending on how pixels store information, images can be of different types, such as grayscale images, where each pixel carries only shades of gray between black and white, or color images, where each pixel represents a combination of colors. The most commonly used model for representing color images is the RGB model, in which every pixel is described by three components: Red (R), Green (G), and Blue (B). By varying the intensity values of these three channels, a wide range of colors can be produced, making RGB images ideal for digital storage and display. [8]

In the context of steganography, the original image that is used to hide secret data is called the cover image, and after the embedding of secret information, the resulting image is known as the stego image. The process of embedding works by making small and carefully controlled modifications to the pixel values of the cover image. Since digital images contain a very large number of pixels, even tiny changes to the least significant part of pixel values remain visually imperceptible, which ensures that the stego image looks almost identical to the original cover image. One of the most popular approaches is the Least Significant Bit (LSB) method, where the last bit in the binary representation of a pixel value is replaced with a bit of the secret data. For instance, if a pixel value in the red channel is 10110110 (which equals 182 in decimal), replacing the last bit with 1 changes it to 10110111 (183 in decimal), which makes no visible difference to the human eye but successfully hides one bit of secret information. During extraction, the reverse process is applied, where the embedded bits are read from the image and reassembled to reconstruct the original secret message, which could be text, another image, or even audio data. Because of the large storage capacity offered by high-resolution images, combined with the imperceptibility of minor pixel alterations, image steganography has become one of the most effective techniques for secure and covert communication in the digital era, with applications ranging from confidential message transmission to digital watermarking and copyright protection.

2.2.2 RGB Model - The Ideal Choice For LSB Image Steganography:

When applying the LSB approach, the choice of color model plays a key role in how effectively information can be concealed. Among the different models, RGB stands out because of its widespread use in digital imaging and its ability to represent colors with remarkable detail. Each pixel is described through varying intensities of red, green, and blue light, and by combining these values a vast spectrum of colors can be generated. With intensity levels generally ranging from 0 to 255 for each channel, images can achieve more than 16 million distinct color variations.

This wide range makes even the smallest bit-level alterations difficult to detect, which is highly advantageous for data hiding. Since every pixel carries three separate values, modifications can be distributed across channels, balancing concealment with image quality.

As a result, the RGB model provides both the flexibility and the capacity needed for embedding secret information while preserving the natural appearance of the cover image.

Representation of Pixels in Binary Digital images are stored and processed in binary format, where every pixel is represented by a combination of bits. In an RGB image, each channel (red, green, and blue) is represented by 8 bits, giving 256 possible intensity levels for each color. Together, the three channels make a 24-bit pixel, allowing a total of 16,777,216 different color combinations. For example, a pixel with RGB values of (120, 200, 75) would be represented in binary as 01111000 (red), 11001000 (green), and 01001011 (blue). This binary structure is crucial in steganography because data embedding is carried out at the bit level, typically by modifying the least significant bits of pixel values. Since only the smallest units of binary data are altered, the visual changes in the image remain negligible, ensuring that the hidden message stays imperceptible to the human eye.

One of the reasons RGB images are frequently chosen for steganographic purposes lies in the way pixel values are represented. Each pixel consists of three components—red, green, and blue—each stored as an 8-bit value. Even the smallest change in these binary values, such as increasing or decreasing an intensity by one unit, has little to no effect on visual perception. For instance, a shift from 100 to 101 in intensity is almost impossible for the human eye to detect. Because every pixel offers three separate values to work with, there is considerable room for subtle modifications without noticeably degrading image quality. In practice, certain channels, like blue, are often better suited for carrying hidden changes, since variations there tend to be less perceptible. These characteristics make RGB images a natural choice for steganographic techniques, supporting both capacity and imperceptibility in data hiding.

RGB images offer several distinct advantages that make them ideal for steganography. First, their widespread use across digital platforms makes them less likely to raise suspicion when used as carriers for hidden information. Second, the 24-bit depth of RGB images provides a high embedding capacity, since each pixel can potentially conceal multiple bits of data across its three channels without noticeable visual degradation. Third, RGB images allow flexibility in choosing embedding strategies that can be hidden in a single channel, such as blue for better imperceptibility, or spread across multiple channels to balance payload and security. Another advantage is that the modifications made during LSB embedding in RGB images are resistant to casual observation, since changes in pixel intensity are minimal and remain within acceptable limits of human visual tolerance. Together, these features make RGB images a reliable and effective medium for secure information hiding.

2.3 TECHNIQUES OF IMAGE STEGANOGRAPHY

The main objective of image steganography is not just to hide data, but to do so in a way that neither visual quality (how the image looks after hiding data) nor statistical characteristics (hidden patterns in an image's pixels values that computers can detect) reveal the presence of

hidden information making it difficult to detect [6] [11]. For this exact purpose, researchers have developed several techniques. Steganography techniques can be broadly classified based on the embedding location, which refers to where the data can be hidden, and the embedding method, which refers to how the data can be hidden [7] [12]

Based on the embedding location, there are two main categories: Spatial Domain Techniques and Transform (Frequency) Domain Techniques. Spatial Domain Techniques involve directly modifying the pixels of an image or the samples of a media file to hide data. Transform (Frequency) Domain Techniques embed data in the transformed coefficients of the media, such as in the frequency components of an image or audio file, which often makes the hidden data more robust against attacks or compression [7] [6].

Based on the embedding method, there are several approaches: Distortion Techniques, Statistical Techniques, Spread Spectrum Techniques, and Masking or Filtering Techniques. Distortion Techniques hide data by slightly altering the original file in a way that can be reversed or detected. Statistical Techniques embed information by changing statistical properties of the media, often without visibly altering it. Spread Spectrum Techniques distribute the hidden data across a wide frequency spectrum to reduce the chance of detection. Masking or Filtering Techniques hide data by embedding it in significant parts of the media, similar to how watermarks are applied, making it less likely to be noticed or removed [7] [11].

2.3.1 Spatial Domain Techniques:

In image processing, the spatial domain refers to the normal form of an image— as a grid of pixels with intensity values. It refers to the domain in which image information is represented as raw pixel intensities at specific spatial positions. Image processing in the spatial domain involves direct operations on these pixel values, such as modifying their brightness, contrast, or replacing selected bits and the easiest way to hide data inside an image is to work directly with its pixels. Therefore, this becomes the whole principle of spatial domain techniques. Also, this makes spatial domain techniques among the simplest and most widely used methods of image steganography [8] [14]. Small changes to pixel values often go unnoticed by the human visual system because adjusting a few numbers in the pixel array does not significantly alter the overall appearance of the image, allowing hidden information to remain invisible

Spatial domain steganography encompasses a variety of techniques for embedding secret data directly into the pixel values of a cover image. Pixel Value Differencing (PVD) utilises the difference between adjacent pixels to determine how many bits can be embedded, allowing smooth regions to carry fewer bits and edge regions to carry more, improving imperceptibility. Edges-Based Data Embedding (EBE) specifically targets the edges of an image, where human vision is less sensitive, to hide data according to the required payload. In Random Pixel Embedding (RPE), embedding positions are determined using a

pseudo-random number generator rather than sequential selection, making unauthorized extraction more difficult. Histogram Shifting is a reversible method that embeds data by shifting pixel-value histograms, allowing full recovery of the original image. Other methods include mapping pixels to hidden data, which modifies pixel values according to controlled rules; pixel intensity-based techniques, which decompose pixel values into multiple bit-planes such as binary, Fibonacci, Prime, or Lucas; labeling or connectivity methods, which embed data in dark regions identified through pixel connections; and texture-based methods, which target complex or textured regions identified via texture analysis, such as gray-level co-occurrence matrices, to enhance imperceptibility and security. Methods such as labeling or connectivity, texture-based, and histogram shifting can also be considered statistical techniques, as they rely on the statistical properties of pixel values [7] [6] [14].

These spatial domain techniques offer several advantages, including high embedding capacity, low computational complexity due to simple algorithms, and minimal perceptual distortion since changes are often imperceptible to the human eye. However, they also have limitations. Their robustness is relatively low, making them vulnerable to compression, filtering, cropping, and noise. Additionally, security can be a concern, as statistical steganalysis can detect LSB-based methods if the embedding is not sufficiently randomized or carefully implemented [8]. Overall, while spatial domain methods are efficient and easy to implement, careful consideration is required to balance capacity, imperceptibility, and security.

2.3.1.1 Least significant bit (LSB):

The Least Significant Bit (LSB) technique is the most widely used and well-known spatial domain approach for image steganography. In a digital image, every pixel is stored as a set of binary bits [6] [14]. In the case of an RGB image, each pixel has three color channels — red, green, and blue — and each channel typically uses 8 bits to represent intensity values. In LSB embedding, the rightmost bit (least significant bit) of each pixel channel is replaced with bits of the secret message. Because this bit contributes the least to the pixel's overall value, modifying it causes negligible change to the image, making the cover image and the resulting stego image appear visually identical. This characteristic allows LSB steganography to achieve high embedding capacity while maintaining imperceptibility, making it the most popular spatial-domain method [7] [2] [3].

The general model of LSB image steganography involves two major stages: embedding and extraction. In the embedding stage, the secret message is first converted into a binary bitstream. A cover image is then selected, and its pixels are processed sequentially. The least significant bits of the pixel values are replaced with the message bits one by one until the entire message is embedded. The output of this process is the stego image, which is then transmitted to the receiver. In the extraction stage, the receiver performs the reverse process: reading the LSBs of the same pixels in the same order and reconstructing the original message bitstream, which is then decoded back into text [6].

For the conversion of text into binary bits, ASCII (American Standard Code for Information Interchange) is often used. ASCII represents characters as numerical codes — for instance, H is 72 and I is 73 in decimal form. ASCII is technically a 7-bit code, but since most computer systems process data in 8-bit bytes, a leading 0 is added, resulting in an 8-bit binary value. For example, H = 01001000 and I = 01001001. ASCII is preferred in basic steganography applications because it is simple, standardized, and widely supported. Other alternatives, such as Unicode, may also be used but are usually necessary only for extended character sets [5].

To illustrate, let us consider a practical example. Suppose a pixel in an RGB image has the value (255, 0, 127). In binary form, this corresponds to:
R = 11111111, G = 00000000, B = 01111111
If the secret message to hide is “HI,” we first convert it to binary as mentioned above. We then replace the least significant bit of each color channel with the message bits. For example, the first bit of “H” is 0, so the R channel changes from 11111111 to 11111110. This process continues across the remaining channels and pixels until all bits of the message are embedded. Because we modify only the least significant bits, the visual difference between the cover image and the stego image remains imperceptible.

To make this clearer, the process can be demonstrated step by step with images.



Figure 1: Cover image used for LSB steganography

	A	B	C	D	E
1	X	Y	R	G	B
2	0	0	94	180	207
3	1	0	94	180	207
4	2	0	94	180	207
5	3	0	94	180	207
6	4	0	94	180	207
7	5	0	94	180	207
8	6	0	94	180	207
9	7	0	94	180	207
10	8	0	94	180	207
11	9	0	94	180	207
12	10	0	94	180	207
13	11	0	94	180	207
14	12	0	94	180	207
15	13	0	94	180	207
16	14	0	94	180	207
17	15	0	94	180	205
18	16	0	94	180	205
19	17	0	94	180	203
20	18	0	94	180	203

Figure 2: Pixel values of the first few pixels of the cover image

	B	C	D	E	F	G	H
1	Y	R	G	B	R_bin	G_bin	B_bin
2	0	94	180	207	01011110	10110100	11001111
3	0	94	180	207	01011110	10110100	11001111
4	0	94	180	207	01011110	10110100	11001111
5	0	94	180	207	01011110	10110100	11001111
6	0	94	180	207	01011110	10110100	11001111
7	0	94	180	207	01011110	10110100	11001111
8	0	94	180	207	01011110	10110100	11001111
9	0	94	180	207	01011110	10110100	11001111
10	0	94	180	207	01011110	10110100	11001111
11	0	94	180	207	01011110	10110100	11001111
12	0	94	180	207	01011110	10110100	11001111
13	0	94	180	207	01011110	10110100	11001111
14	0	94	180	207	01011110	10110100	11001111
15	0	94	180	207	01011110	10110100	11001111
16	0	94	180	207	01011110	10110100	11001111
17	0	94	180	205	01011110	10110100	11001101
18	0	94	180	205	01011110	10110100	11001101
19	0	94	180	203	01011110	10110100	11001011
20	0	94	180	203	01011110	10110100	11001011

Figure 3: Binary representation of the pixel values before embedding.

	A	B	C	D
1	Character	ASCII Decimal	8-bit Binary	
2	H	72	01001000	
3	I	73	01001001	
4				
5				
6				
7				

Figure 4: ASCII and binary (bit) representation of the message “HI” to be embedded.

B	C	D	E	F	G
Color	Original	Original_Bin	New	New_Bin	HiddenBit
R	94	01011110	94	01011110	0
G	180	10110100	181	10110101	1
B	207	11001111	206	11001110	0
R	94	01011110	94	01011110	0
G	180	10110100	181	10110101	1
B	207	11001111	206	11001110	0
R	94	01011110	94	01011110	0
G	180	10110100	180	10110100	0
B	207	11001111	206	11001110	0
R	94	01011110	95	01011111	1
G	180	10110100	180	10110100	0
B	207	11001111	206	11001110	0
R	94	01011110	95	01011111	1
G	180	10110100	180	10110100	0

Figure 5: Modified pixel values after embedding “HI” in the RGB channels.



Figure 6: Stego image obtained after embedding the message.

2.3.2 Other Image Steganography Techniques:

Other than spatial domain methods, image steganography also employs several advanced techniques that work beyond direct pixel modification [6] [8] [12]. **Transform (Frequency) Domain Techniques** embed secret data by modifying the frequency coefficients of an image rather than the pixel values themselves. The image is first converted into a frequency

representation using mathematical transforms such as the Discrete Cosine Transform (DCT), commonly used in JPEG compression, where data is embedded in middle-frequency coefficients to minimize visible artifacts; the Discrete Wavelet Transform (DWT), which decomposes the image into sub-bands (LL, LH, HL, HH) and embeds data often in LH/HL sub-bands for a balance between imperceptibility and robustness; and the Discrete Fourier Transform (DFT), which allows embedding in the phase or magnitude of frequency components. Transform domain techniques provide higher robustness against compression, filtering, and noise, while maintaining visual quality, though they are computationally more complex than spatial-domain methods and often require an inverse transform for data recovery [7] [6].

Distortion Techniques hide information by intentionally modifying selected features of the cover image in a controlled manner [6]. The sender applies a known pattern of changes, such as slightly altering pixel positions or adding a known noise pattern representing the secret message. The receiver must have access to the original image or a reference image to detect these differences and recover the hidden data. While prediction-based distortion methods can be highly robust, they are not suitable for blind steganography, where no original image is available [7] [6].

Statistical Techniques embed data by modifying measurable properties of an image, such as pixel value distributions or histograms [8]. Changes are computationally detectable but visually imperceptible, allowing the accurate extraction of hidden information while maintaining the statistical characteristics of the image. Examples include labeling or connectivity methods, texture-based methods, and histogram shifting. These techniques often complement other methods by improving security and imperceptibility [8] [12].

Spread Spectrum Techniques enhance robustness by dispersing secret data across multiple frequencies of an image, making it appear as noise [12]. Examples include Direct Sequence Spread Spectrum (DSSS), where the message is multiplied by a pseudo-random sequence, and frequency domain embedding using pseudo-random sequences in DCT or DWT coefficients. These methods are resistant to compression, cropping, and noise, but the embedding and extraction process is more complex and may reduce capacity compared to simpler techniques [12].

Finally, **Masking and Filtering Techniques** exploit the human visual system by embedding data in perceptually significant areas, such as edges or textured regions, where small changes are less noticeable [6]. Data can be hidden by modifying edge regions or adjusting luminance and chrominance components without degrading visual quality. Although these methods provide high imperceptibility and resistance to image processing operations, their embedding capacity is limited, and the techniques are slightly more complex than basic spatial-domain methods [7] [6].

CHAPTER 3: QUALITY METRICS

3.1 INTRODUCTION AND EVALUATION AXES

In image steganography, the primary goal is to conceal information in a way that does not compromise the visual quality or usability of the image [6] [8]. To objectively assess this, researchers rely on quality metrics, which quantify the distortion introduced during data hiding and measure how effectively the hidden information is embedded without being detected [13]. These metrics are essential for comparing different steganography techniques and optimizing them for practical applications. Quality metrics focus on imperceptibility, robustness, capacity, and security, forming a comprehensive framework to evaluate the performance of steganographic methods.

Embedding secret data alters pixel values, which can affect both the statistical properties and the perceptual quality of the image. Without proper evaluation, stego images may be visually degraded, raising suspicion, fail to maintain fidelity required for applications like medical imaging or satellite imagery, or become vulnerable to detection through steganalysis techniques [5] [12]. Hence, evaluating image quality post-embedding is critical for ensuring imperceptibility, where the hidden data should not be noticeable to human observers or statistical tests, comparing algorithm efficiency by benchmarking different methods objectively using consistent metrics, and balancing capacity and fidelity to maximize hidden data size without significantly reducing image quality [8].

The performance of image steganography techniques is generally assessed along several axes. Imperceptibility measures the visual similarity between the original cover image and the stego image, with common metrics including Peak Signal-to-Noise Ratio (PSNR), which quantifies the difference in pixel intensity values where a higher PSNR indicates less distortion, Structural Similarity Index (SSIM), which evaluates perceived changes in structural information, luminance, and contrast, and Mean Squared Error (MSE), which measures the average squared difference between cover and stego pixels [4] [13]. Capacity refers to the maximum amount of data that can be embedded without degrading the image beyond acceptable thresholds, though larger capacity often requires trade-offs with imperceptibility [6]. Robustness is the ability of the stego image to resist manipulations or attacks such as compression, noise addition, filtering, and cropping, and is measured through metrics such as Bit Error Rate (BER) and Normalized Correlation [10]. Security assesses how difficult it is for attackers to detect or extract the hidden information using steganalysis techniques, often evaluated through entropy analysis, histogram analysis, or specialized steganalysis algorithms [2] [11]. In the context of cryptosystems, cryptography-related evaluation metrics are also considered, including key sensitivity, where small changes in the key should produce significantly different encrypted images, statistical analysis resistance, which ensures that ciphertext does not reveal patterns of the plaintext, avalanche effect,

which measures how a single bit change in plaintext or key affects the output, and entropy, which quantifies randomness in encrypted data, where higher entropy indicates better security [1] [3].

3.2 PEAK SIGNAL-TO-NOISE RATIO (PSNR) AND MEAN SQUARED ERROR (MSE)

In image processing, steganography, and video compression, assessing the quality of a processed image compared to its original version is critical. Mean Squared Error (MSE) and Peak Signal-to-Noise Ratio (PSNR) are two of the most widely used full-reference quality metrics for this purpose, meaning they require access to the original image for comparison. These metrics are essential for quantifying the differences introduced in images due to processes such as embedding, compression, transmission, or noise. While MSE provides a pixel-level numerical representation of error, PSNR translates this error into a logarithmic scale, offering a more interpretable assessment of image fidelity. The relationship between the two metrics is straightforward: PSNR is inversely proportional to MSE. A smaller MSE results in a higher PSNR, indicating better image quality, whereas a larger MSE corresponds to a lower PSNR, reflecting greater distortion. Despite their simplicity and widespread use, these metrics do not always perfectly correlate with human visual perception, highlighting the importance of combining them with perceptual measures like SSIM for more comprehensive quality assessment [4] [13].

MSE and PSNR are inherently linked: PSNR is a logarithmic function of MSE, meaning PSNR translates pixel-level errors into a scale that is easier to interpret in terms of signal strength versus noise. The main difference is that MSE provides a direct numeric measure of error, while PSNR expresses the same error in decibels, allowing for intuitive comparisons across images or algorithms. Both metrics are complementary: MSE is excellent for raw error computation, and PSNR is preferable for interpretability and benchmarking [4] [13].

3.2.1 Mean Squared Error (MSE):

Mean Squared Error (MSE) is a widely used full-reference quality metric that measures the average squared difference between the pixel values of an original image and a processed image. Mathematically, it is expressed as:

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2$$

where I (i, j) represents the pixel value of the original image, K (i, j) is the pixel value of the processed image at the same position, m and n denote the dimensions of the image. By computing the squared difference, MSE ensures that both positive and negative deviations are

treated equally, while also penalizing larger deviations more heavily. This squaring makes MSE particularly sensitive to significant changes in pixel values, emphasizing areas where the processing or embedding has introduced larger errors. By averaging these squared differences over all pixels in the image, MSE produces a single numeric value representing the overall error across the entire image [4] [13].

MSE can be interpreted as a pixel-wise similarity measure. An MSE value of zero indicates that the original and processed images are identical, whereas increasing MSE values reflect a decrease in similarity and greater distortion. This characteristic makes MSE a straightforward and objective metric for evaluating the fidelity of image processing algorithms, including steganography, image compression, and image transmission. In steganography, for instance, MSE quantifies the imperceptibility of embedded data by measuring how much the pixel values have changed after hiding a secret message. Similarly, in compression or transmission, MSE evaluates the loss or distortion introduced by encoding or channel noise. Despite its simplicity and widespread use, MSE has several limitations. It does not account for perceptual quality or structural information, meaning that two images with similar MSE values can appear visually different if distortions occur in perceptually sensitive regions. MSE is also sensitive to small shifts, rotations, or localized noise patterns, which may be imperceptible to the human eye but still increase the metric. Therefore, while MSE is an excellent objective measure for quantifying pixel-level changes, it is often complemented by perceptual metrics such as SSIM or VMAF to provide a more accurate assessment of image quality. MSE is crucial for assessing algorithm performance in image processing and steganography. It provides objective, reproducible, and standardized measures of distortion, helping researchers and developers compare methods and ensure that embedded or compressed images maintain high fidelity to the original. Its advantages include simplicity, wide adoption, and direct interpretation, making it an indispensable metric in both research and industry contexts [4] [13].

3.2.2 Peak Signal-To-Noise Ratio (PSNR):

Peak Signal-to-Noise Ratio (PSNR) is a widely used full-reference quality metric that complements MSE by expressing image fidelity in a logarithmic, decibel (dB) scale (Forms 1, 5). PSNR is derived directly from MSE and provides a measure of the ratio between the maximum possible pixel value of an image and the power of the error (noise) introduced during processing. It is mathematically expressed as:

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right)$$

where MAX_I is the maximum possible pixel value (e.g., 255 for 8-bit images) and MSE is the mean squared error between the original and processed images. Because PSNR is logarithmic, it compresses the scale of large differences, making it easier to interpret and compare the quality of images processed by different algorithms.

PSNR values are inversely related to MSE: a lower MSE results in a higher PSNR, indicating better image fidelity, whereas a higher MSE leads to a lower PSNR, reflecting greater distortion. This relationship allows PSNR to serve as a human-interpretable measure of image quality, providing insight into how closely a processed image approximates the original. Typically, PSNR values ranging from 30–50 dB indicate good image quality, values above 50 dB suggest excellent quality with imperceptible changes, and values below 30 dB are generally considered poor [4] [13].

PSNR is widely applied in steganography, image compression, video streaming, and transmission quality assessment. In steganography, it helps quantify the imperceptibility of embedded data, indicating how much the embedding process alters the original image. In compression and transmission, PSNR is used to evaluate reconstruction fidelity, measuring how closely the decoded or received image matches the original. Its importance lies in its simplicity, standardization, and ability to provide reproducible, objective assessments of image quality. Despite its advantages, PSNR has limitations. Like MSE, it does not fully capture human visual perception. Two images with identical PSNR values can appear visually different if structural distortions or localized artifacts exist. PSNR is also insensitive to perceptually significant variations such as edge degradation, color shifts, or texture changes. To address these shortcomings, PSNR is often used alongside perceptual metrics such as SSIM or VMAF, which incorporate structural and visual system information for a more accurate evaluation of perceived quality. While standard PSNR provides a global measure of image quality, several variants exist to better align with human perception. For example, PSNR-HVS incorporates aspects of the human visual system, and PSNR-B evaluates block-based distortions common in compressed images. These variants enhance PSNR's applicability in scenarios where perceptual fidelity is crucial [4] [13].

3.3 STRUCTURAL SIMILARITY INDEX MEASURE

When evaluating the quality of images in steganography, it is essential to measure how closely the stego image resembles the original cover image. In this project, two metrics were used for quality assessment: Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM). While PSNR is a widely used metric, it has a significant limitation: it only measures pixel-level errors between two images [4]. A higher PSNR value generally suggests that two images are similar, but PSNR does not always reflect how humans actually perceive visual quality. For example, two images can have the same PSNR score but look very different to the human eye. This happens because PSNR is based on mathematical error calculations (MSE) and does not account for visual perception [13].
To address this limitation, SSIM was introduced. Unlike PSNR, which simply measures error magnitude, SSIM focuses on structural information, luminance, and contrast, which are critical aspects of human vision. It provides a more reliable and perceptual measure of image quality, ensuring that the structural integrity of the image is preserved after data embedding [13]. By using SSIM alongside PSNR, this project ensures a comprehensive evaluation:

PSNR confirms that the pixel-level distortion is minimal. SSIM verifies that the structural and perceptual quality of the image is maintained [2].
Thus, SSIM serves as a complementary metric to PSNR, strengthening the validity of the quality analysis in this study.

SSIM is widely used in the field of image processing because it provides a perceptual measure of image quality that aligns closely with the way humans evaluate visual information. Unlike traditional error-based metrics, such as MSE and PSNR, which only quantify differences in pixel intensity values, SSIM focuses on the preservation of structural information. This makes it more effective in capturing how natural, sharp, and undistorted an image appears to the human eye. Its ability to highlight perceptual degradations makes SSIM a preferred choice in applications like compression, transmission, denoising, and steganography, where maintaining visual fidelity is more important than minimizing raw numerical error [12].

The Mean Squared Error (MSE) and the Peak Signal-to-Noise Ratio (PSNR) are traditional error-based metrics used to evaluate image quality. MSE calculates the average squared difference between corresponding pixels of two images, while PSNR is derived from MSE and expresses the ratio between the maximum possible signal power and the power of the error. Both metrics are simple to compute and widely used, but they treat all pixel differences equally and fail to consider how humans perceive visual quality. As a result, two images with the same MSE or PSNR may look very different to the human eye.
The Structural Similarity Index Measure (SSIM), on the other hand, was developed to overcome these shortcomings [4]. Instead of focusing only on pixel-level errors, SSIM evaluates image similarity based on luminance, contrast, and structural information, which are more closely aligned with the human visual system. While MSE and PSNR can indicate whether numerical differences exist between two images, SSIM provides a perceptual measure of similarity, capturing whether the images appear natural and undistorted to the viewer. This makes SSIM a more reliable metric for applications such as steganography, where maintaining imperceptibility is essential.

In SSIM, three **main components** are compared to evaluate image quality: luminance, contrast, and structure. Luminance refers to the overall brightness level of an image, and SSIM first compares the average intensity values of the two images to determine how close they are in terms of brightness. If the luminance of both images is similar, it indicates that the general lightness or darkness has been preserved after processing, which is important because even small changes in brightness can significantly affect human perception. Contrast measures the variation in intensity across different regions of an image, and in SSIM, contrast comparison evaluates how well the differences between light and dark regions are maintained between the original and processed image. Preserving contrast is crucial because it enhances details, textures, and depth, whereas loss or exaggeration of contrast can make an image appear flat or unnatural. Structure comparison focuses on the arrangement of pixels, including edges, patterns, and textures, which carry the most meaningful information in an image. SSIM evaluates the correlation between structural patterns of the two images to

determine how much of the fine detail and geometry has been preserved [13]. This aspect is especially important because the human visual system is highly sensitive to structural distortions, such as blurred edges or missing textures, which strongly affect perceived image quality.

3.3.1 Types and Properties of SSIM:

The Structural Similarity Index Measure (SSIM) has been extended into different variations to address the limitations of the basic model and improve its performance across diverse image processing applications. The original version, Single-Scale SSIM (SSIM), proposed by Wang et al. in 2004, evaluates similarity between two images at a single scale, usually at the original resolution.

It considers three components—luminance, contrast, and structure—combining them into a single similarity index. While it is computationally simple and widely used for quick evaluation, it may fail to capture distortions that appear at different resolutions or scales. To overcome this, Multi-Scale SSIM (MS-SSIM) was introduced, which computes similarity at multiple scales by progressively downsampling the image. This allows it to capture both fine details such as edges and textures, as well as global structures like shapes and patterns, making it more robust and comprehensive than single-scale SSIM. MS-SSIM is particularly effective in applications such as image compression, video streaming, and steganography [2].

Another variation, Complex Wavelet SSIM (CW-SSIM), uses the complex wavelet transform instead of pixel intensities and is designed to handle geometric distortions such as small shifts, rotations, or translations that traditional SSIM struggles with. By analyzing phase information in wavelet coefficients, CW-SSIM provides stable similarity measures even when images are not perfectly aligned, which is especially useful in image registration and medical imaging. Information-Weighted SSIM (IW-SSIM) further improves upon SSIM by assigning different weights to regions of the image based on their importance.

High-information regions, such as edges and textures, are given greater weight since the human eye is more sensitive to distortions in these areas. This makes IW-SSIM closer to subjective visual judgments and particularly valuable in perceptual coding, adaptive image compression, and quality optimization systems.

SSIM offers several advantages, including closely matching human visual perception, detecting structural and perceptual distortions more effectively than error-based metrics, and being relatively simple and computationally efficient to implement. However, it also has limitations, as it is less effective when images undergo geometric distortions such as rotation or scaling, may not perform well for images viewed under different conditions, and focuses only on structural similarity, potentially overlooking other perceptual aspects.

SSIM finds applications in various fields of image and video processing. It is used for evaluating image compression by measuring quality loss after compression, assessing

denoising performance by comparing original and denoised images, and ensuring that hidden data in steganography and cryptography does not distort visual quality [12]. Additionally, SSIM is employed in video quality assessment to measure frame-by-frame quality and in medical imaging to evaluate the quality of MRI, CT, and other diagnostic images.

The computation of SSIM involves several steps. First, both images are converted to grayscale, although color-based SSIM variants also exist. Next, the images are divided into sliding windows across the image. For each window, local statistics such as mean, variance, and covariance are calculated. The SSIM formula is then applied to compute SSIM values for each local window. Finally, the SSIM values of all windows are averaged to obtain the global SSIM index. The resulting SSIM score ranges between 0 and 1, with a value of 1 indicating perfect similarity and values closer to 0 indicating greater dissimilarity.

The choice of SSIM in this study is motivated by its effectiveness in evaluating image quality in a manner consistent with human visual perception. In the context of LSB steganography, the primary objective is to embed data without introducing distortions that are easily noticeable [14]. Traditional error-based metrics often fail to capture perceptual quality, whereas SSIM provides a more comprehensive evaluation by considering structural and perceptual aspects of the image. This makes it a suitable quality metric for conducting a comparative analysis of steganographic techniques.

3.4 COMPARISON OF IMAGE QUALITY METRICS

The **Mean Squared Error (MSE)** is one of the simplest and most commonly used full-reference metrics for image quality assessment. It measures the quality of an image by calculating the average of the squared differences between the original and the distorted image pixels. A lower MSE value, closer to zero, indicates higher image quality. Although MSE is mathematically simple and widely used, it does not always correspond to how humans perceive image quality. From MSE, another metric called the Root Mean Squared Error (RMSE) can be derived, which presents the error in the same units as the original data, making it easier to interpret. The **Peak Signal-to-Noise Ratio (PSNR)** is another popular metric, which is closely related to MSE. Instead of reporting absolute errors, PSNR expresses the ratio between the maximum possible signal power and the power of the noise (distortion) introduced in the image. It is expressed in decibels (dB), where higher PSNR values represent better quality and lower distortion. For typical 8-bit images, PSNR values usually range between 30–50 dB, while for 16-bit images, values range between 60–80 dB. PSNR is widely used for evaluating image and video compression, but like MSE, it does not always match human perception of visual quality. The **Structural Similarity Index (SSIM)**, on the other hand, is a perception-based model that was developed to overcome the limitations of MSE and PSNR. SSIM evaluates the similarity between two images by considering structural information, luminance, and contrast, which are more closely aligned with how the human visual system perceives image quality. The value of SSIM ranges between 0 and 1, where 1

indicates perfect similarity with the reference image. Advanced versions of SSIM, such as the Multi-Scale SSIM (MS-SSIM) and the Three-Component SSIM (3-SSIM), provide even more accurate results by analyzing images at different scales or focusing separately on edges, textures, and smooth regions.
In summary, MSE is simple and efficient but does not reflect visual perception well. PSNR provides a more interpretable measure in decibels and is widely used in compression, yet still lacks perceptual accuracy. SSIM is the most effective among the three for representing human-perceived quality, as it takes structural and perceptual factors into account.

Every evaluation metric has its **weaknesses** and **strengths**, which must be considered when evaluating the effectiveness of steganography techniques. The Mean Squared Error (MSE) is a widely used metric because of its simplicity and ease of calculation. Its strength lies in being straightforward, mathematically efficient, and providing a clear measure of absolute error between the original and distorted images. It also allows derivation of related measures such as the Root Mean Squared Error (RMSE). However, its main weakness is that it does not align well with human visual perception, since it only considers pixel-by-pixel differences and treats all errors equally, even if they are not noticeable to the human eye. The Peak Signal-to-Noise Ratio (PSNR) builds upon MSE and is valued for being easy to interpret because it expresses error in decibels. Its strength is that it is widely used in image and video compression, with clear reference ranges (e.g., 30–50 dB for 8-bit images) that help in evaluating reconstruction quality. It is especially useful when comparing the performance of compression codecs. The weakness of PSNR, however, is similar to MSE—it does not always reflect human perception of image quality, meaning that two images with the same PSNR can still look quite different to the human eye. The Structural Similarity Index (SSIM) overcomes many of the drawbacks of MSE and PSNR. Its strength lies in being perception-based, as it considers luminance, contrast, and structural information, which makes it better suited for evaluating images in a way that matches human vision. SSIM also has advanced versions such as MS-SSIM and 3-SSIM, which provide even more accurate assessments across scales and image regions. The main weakness of SSIM is that it is computationally more complex than MSE and PSNR, and may take more time and resources to calculate.

It is important to know **when and how to use each metric** to effectively assess steganographic methods. The Mean Squared Error (MSE) should be used when a straightforward mathematical measure of error is needed. Since it calculates the average squared difference between the original and the reconstructed image, it is most effective in cases where precise numerical error values are required. Because lower values indicate better results, MSE is often used as a baseline or reference metric in image processing tasks. The Peak Signal-to-Noise Ratio (PSNR) is best used in evaluating the quality of image and video compression, where the main concern is how much distortion has been introduced during compression.PSNR is widely applied to measure reconstruction quality in lossy codecs, with typical acceptable ranges defined in decibels. It is useful when comparing different compression methods or analyzing transmission quality, especially when results need to be expressed in a standard numeric scale. The Structural Similarity Index (SSIM) is most

suitable when the goal is to assess image quality in a way that matches human perception. SSIM works by comparing structural information, luminance, and contrast, which means it is particularly effective when small distortions might not change numerical error much but still affect how humans see the image. It is therefore ideal for applications like steganography, watermarking, or image enhancement, where perceptual similarity is more important than raw error values.

Other metrics used for image quality assessment include DSSIM and FSIM. DSSIM, or Structural Dissimilarity, is a distance metric derived from the Structural Similarity Index (SSIM) that measures the dissimilarity between images. The Feature Similarity Index Matrix (FSIM) maps image features and evaluates the similarities between two images, providing an alternative approach for assessing image quality based on feature information.

CHAPTER 4: COMPARATIVE ANALYSIS OF CRYPTOGRAPHY EMBEDDED LSB IMAGE STEGANOGRAPHY USING QUALITY METRICS

While steganography hides the presence of data, cryptography protects its contents. Advanced Encryption Standard (AES) is a symmetric block cipher standardized by NIST in 2001. AES works on 128-bit blocks with keys of 128, 192, or 256 bits. It encrypts data in rounds of substitution, shifting, mixing, and adding a round key—10, 12, or 14 rounds based on key size.
In our project, the secret text is first encrypted using AES before being embedded into the image. This adds a second layer of security: even if the hidden message is detected, it cannot be read without the correct key. The encrypted data is then converted into binary form so it can be embedded into the image's pixel values using the LSB method.

Rivest–Shamir–Adleman (RSA) is a public-key encryption technique that uses a pair of keys: a public key to encrypt data and a private key to decrypt it. Unlike AES, which is symmetric, RSA can securely transmit information without sharing a secret key in advance. RSA encryption relies on the difficulty of factoring large prime numbers. A key pair is generated by selecting two large primes, calculating their product (modulus), and deriving the encryption and decryption exponents.
In our project, RSA is used to encrypt the secret message before any further processing. Because RSA encryption usually produces a longer ciphertext than AES, it allows us to compare how message length affects encryption performance and efficiency.

4.1 METHODOLOGY AND SETUP

The methodology in this study focuses on combining cryptography methods with LSB image steganography and evaluating the resulting image quality of the original and new image using PSNR and SSIM metrics. First we encrypt a secret message using both AES method and RSA method separately. Then the process of Steganography is done in 4 different ways: separately in the Red, Green and Blue channels and finally in the combined RGB channel.
This consists of three main phases:
Phase 1: **ENCRYPTION** of the message to be hidden using AES Cryptography.
Phase 2: **EMBEDDING** the AES encrypted message to the cover images using LSB steganography in multiple colour channels.
Phase 3: **ENCRYPTION** of the same message using RSA Cryptography.
Phase 4: **EMBEDDING** the RSA encrypted message to the cover images using the same LSB Steganography in all the colour channels.

Phase 5: **EVALUATION** of the image quality using PSNR and SSIM metrics.

For this study, we applied Least Significant Bit (LSB) image steganography to the individual Red (R), Green (G), and Blue (B) channels, as well as to the combined RGB channels, in order to compare their performance in terms of invisibility and visual distortion. The goal was to analyze how embedding encrypted data in different color channels affects the perceptual quality of the image and the detectability of the hidden message and also how the length of the message affects its invisibility after embedding. . The message chosen for embedding was the text: "The picture hides a secret message". Before embedding, as the first step, the message was encrypted using the Advanced Encryption Standard (AES) in Cipher Block Chaining (CBC) mode. This ensured that even if the hidden data were extracted from the image, it would remain unintelligible without the correct decryption parameters. The encryption was carried out using a randomly generated key and initialization vector. The resulting ciphertext was:
“seyUoJJ5v0w4jwkuREX08AtlUEmEGhHt19PIuFZZTJ0sLJ5tCbYyo8q/eNct1mWO”
This ciphertext, rather than the plain text message, was used as the input for the embedding process. By encrypting the message prior to steganography, the experiment provided an additional layer of security, ensuring that the concealed data could not be read without both the steganographic extraction process and the correct AES decryption credentials.

To conduct this comparison, four images with distinctly different brightness levels and their Black and white (grayscale) version were selected. The first image is a bright, high-luminance photograph of a landscape containing strong color contrasts and saturated tones, making it ideal for assessing how vibrant pixel values influence the embedding process. The second image is a dim, low-luminance photograph having darker shades, muted colors, and reduced contrast, which provides a contrasting case for evaluating embedding quality in low-light visuals. Both images were chosen to be of similar resolution to maintain fairness in comparison. In addition to the bright and dull images, both images were also converted to their black and white (grayscale) versions. This was done to evaluate whether hiding the secret message performs differently in grayscale images compared to coloured images of varying brightness.



Bright Image:
Image Resolution: 736 x 460 pixels
Total Pixels: 338560
Image Format: JPEG
Color Mode: RGB
Number of Channels: 3 (Red, Green, Blue)
LSB Capacity: 1015680 bits (126960 bytes)



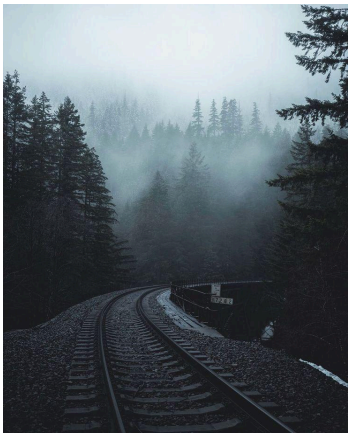
Dull Image:
Image Resolution: 736 x 920 pixels
Total Pixels: 677120
Image Format: JPEG
Color Mode: RGB
Number of Channels: 3 (Red, Green, Blue)
LSB Capacity: 2031360 bits (253920 bytes)
Approximate Text Capacity: 253920 characters

4.2 QUALITY ASSESSMENT OF STEGO IMAGES

Bright Image:



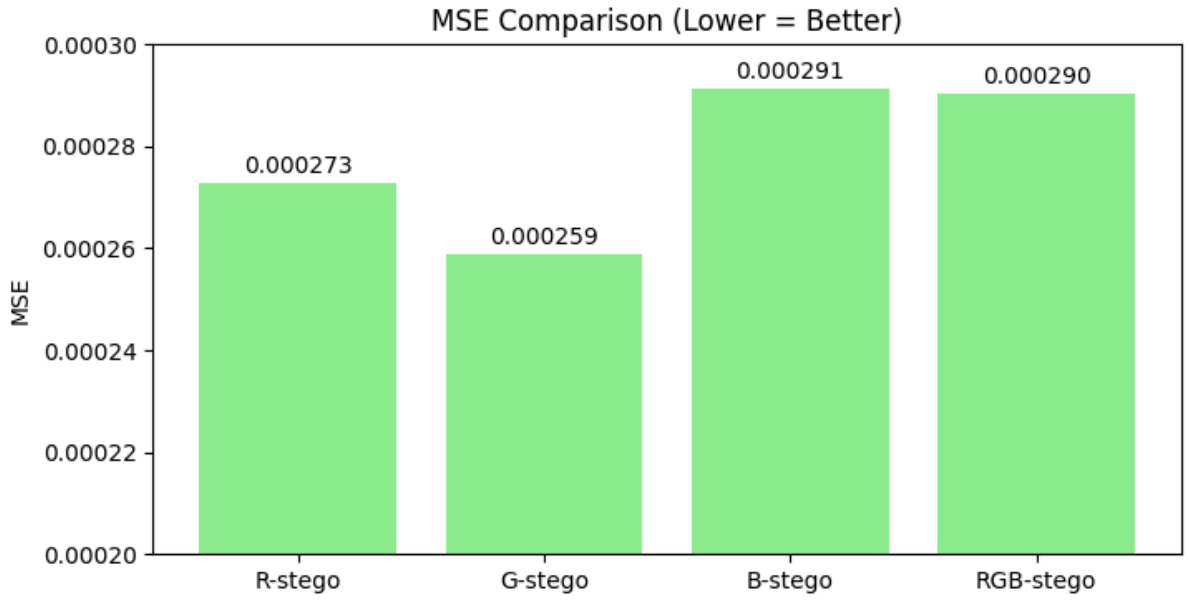
Dull Image:



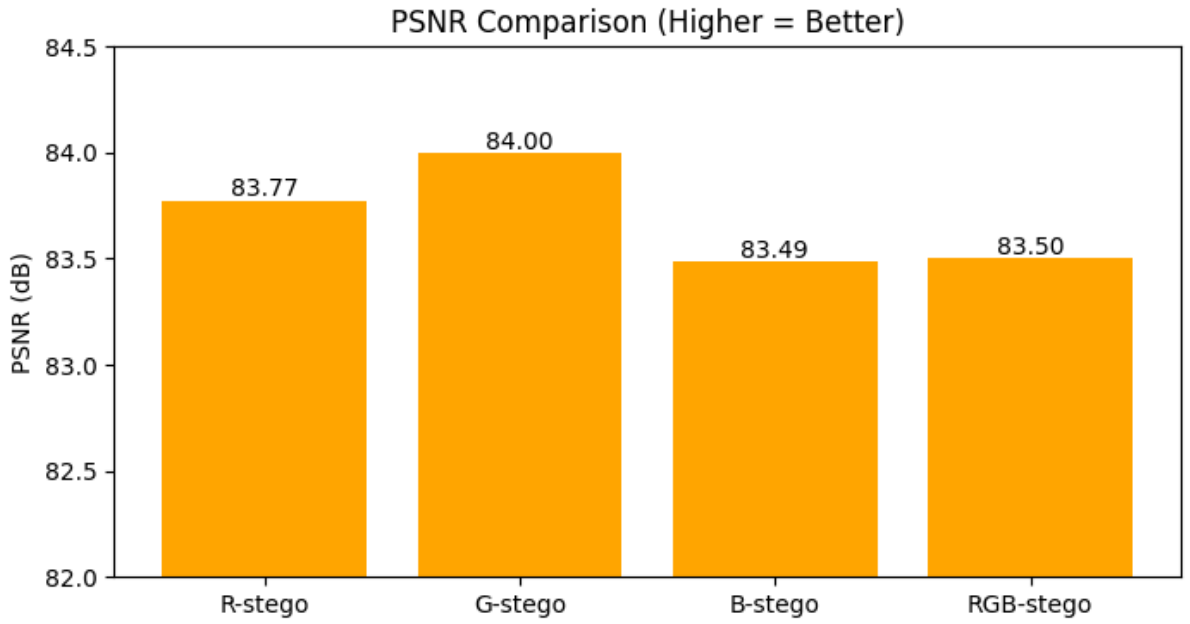
4.2.1 AES Encryption in Bright Image:

	A	B	C	D	E
1	Image	Filename	MSE	PSNR (dB)	SSIM
2	R-stego	image_with_message.png	0.0002727237	83.77357479616735	0.9999997872326508
3	G-stego	image_with_message_G.png	0.00025893984	83.99881493142615	0.9999997426744898
4	B-stego	image_with_message_B.png	0.00029143036	83.48545564709562	0.9999995155569955
5	RGB-stego	image_with_message_RGB.png	0.0002904458	83.50015246999561	0.9999997348976256
6					
7					

In the case of bright images processed with AES encryption, the embedding process achieves remarkable efficiency.

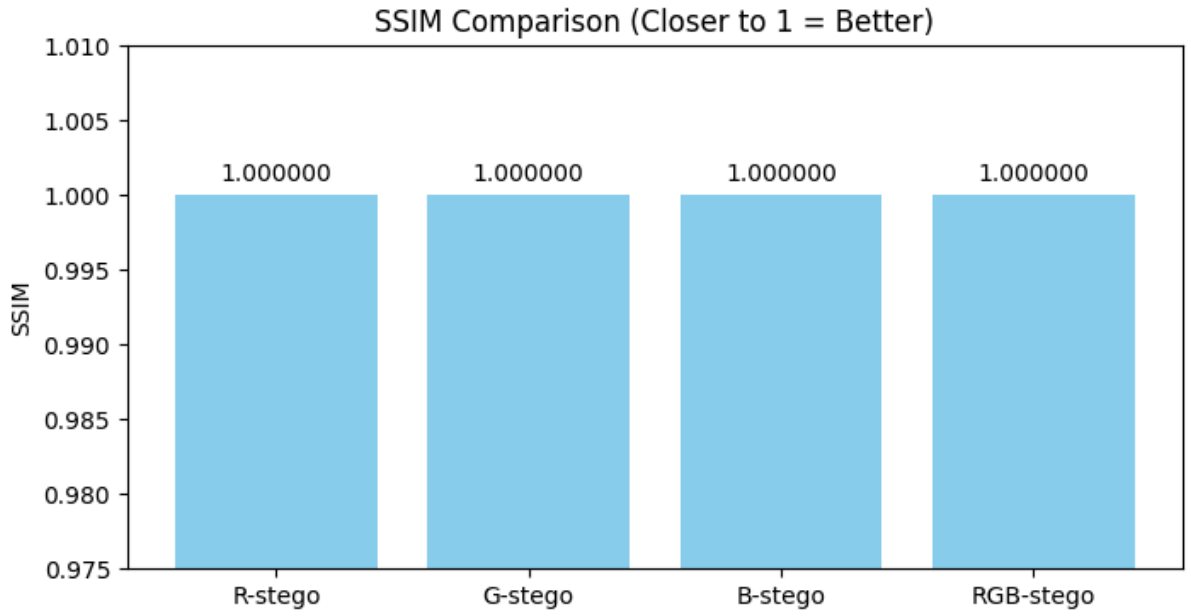


The Mean Squared Error (MSE) values remain very small, falling within the range of 0.00025 to 0.00029. Such low values indicate that the pixel-level modifications introduced by the embedding process are almost negligible and do not compromise the original structure of the image.



Furthermore, the Peak Signal-to-Noise Ratio (PSNR) values for this case consistently remain high, approximately 83.4 to 84 dB. These values highlight the minimal distortion that occurs during embedding and confirm that the perceptual quality of the stego image is maintained at

an excellent level. This ensures that any hidden information remains well concealed without degrading the visible clarity of the cover image.

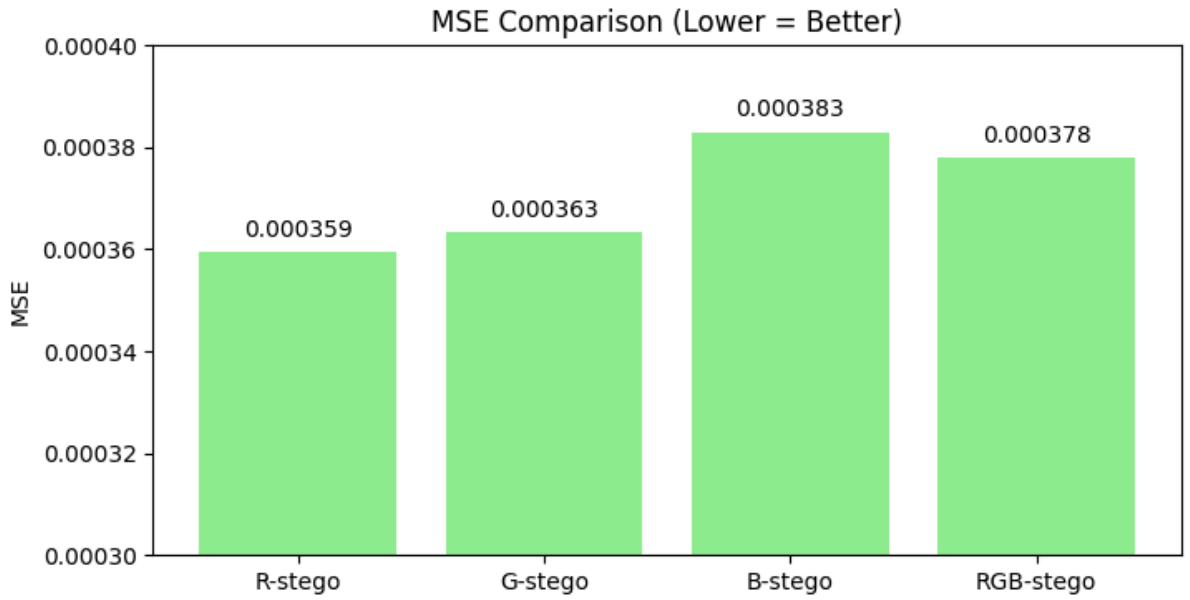


The Structural Similarity Index (SSIM) values further strengthen these findings, as they remain extremely close to 1 (ranging from 0.9999995 to 0.9999997). This result indicates that the structural similarity between the original and the stego images is nearly perfect. Overall, these observations suggest that AES encryption, when applied to bright images, offers both high security for embedded data and superior imperceptibility of changes.

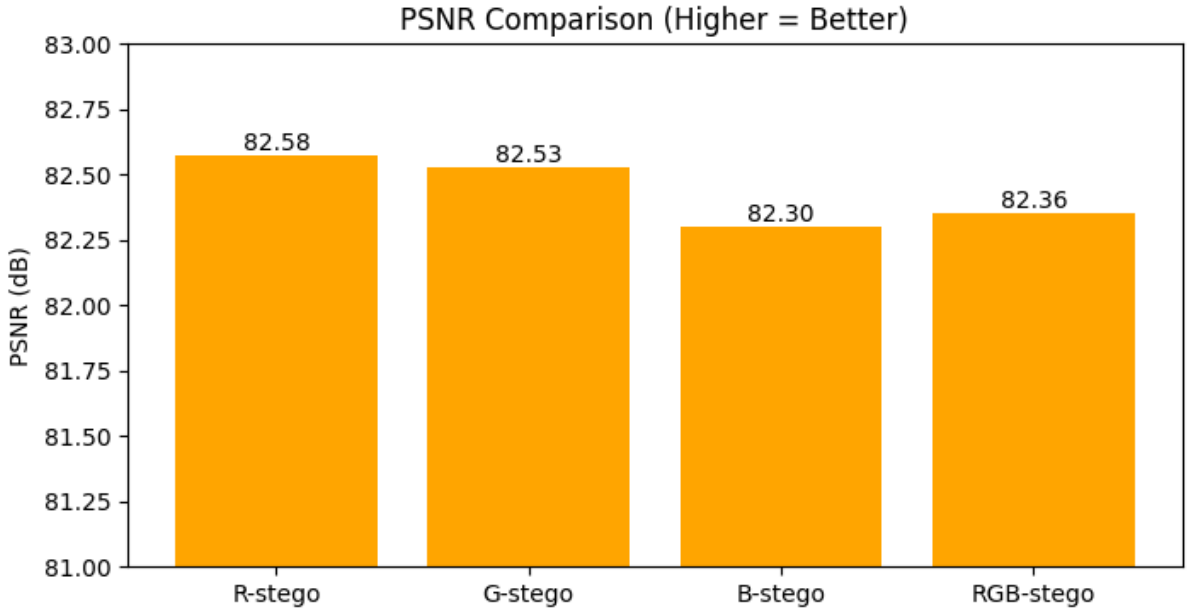
4.2.2. RSA Encryption in Bright Image:

	A	B	C	D	E
1	Image	Filename	MSE	PSNR (dB)	SSIM
2	R-stego	image_with_message (1).png	0.00035936516	82.57544386002363	0.9999995905266147
3	G-stego	image_with_message_G (1).png	0.0003633034	82.52810890686041	0.9999995204321558
4	B-stego	image_with_message_B (1).png	0.00038299465	82.29887652478547	0.9999993039998302
5	RGB-stego	image_with_message_RGB (1).png	0.00037807183	82.35506046883154	0.9999996386467099
6					
7					

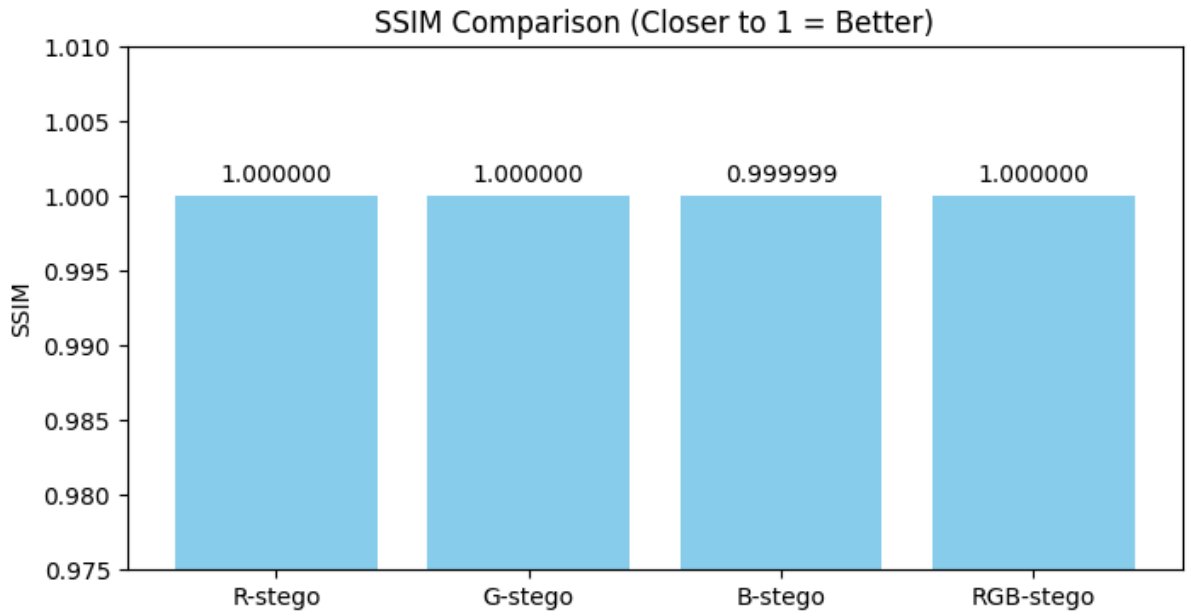
When RSA encryption is applied to bright images, the results, though still strong, are slightly weaker in comparison to AES.



The MSE values rise to around 0.00035–0.00038, which reflects a slightly greater degree of pixel-level distortion introduced during embedding. Although still small, this increase suggests that RSA has a comparatively higher impact on bright images.



In line with this, the PSNR values show a modest decline, ranging from 82.2 to 82.5 dB. While these values remain in the high-quality range, the reduction compared to AES demonstrates that RSA introduces more noticeable distortion. This implies that the perceptual quality of stego images with RSA is somewhat less preserved, though still visually acceptable.

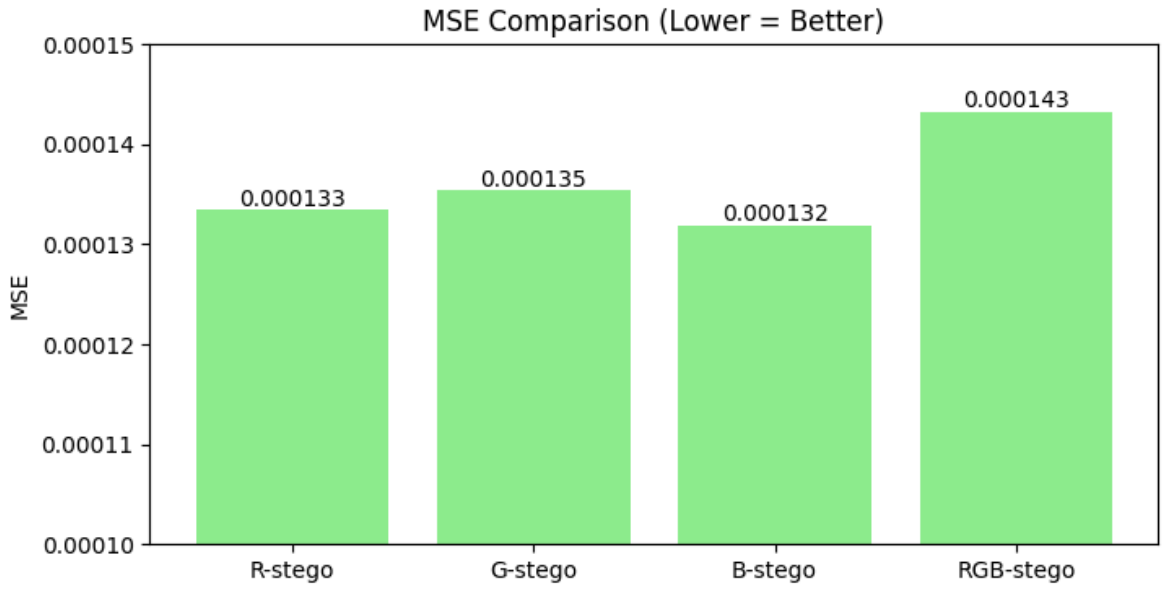


Despite these quantitative differences, the SSIM values remain nearly perfect (0.9999993–0.9999996). This ensures that the stego images remain visually indistinguishable from the originals. Nevertheless, the combined analysis of MSE and PSNR clearly suggests that AES is the more suitable option for bright images, as it minimizes embedding distortion more effectively than RSA.

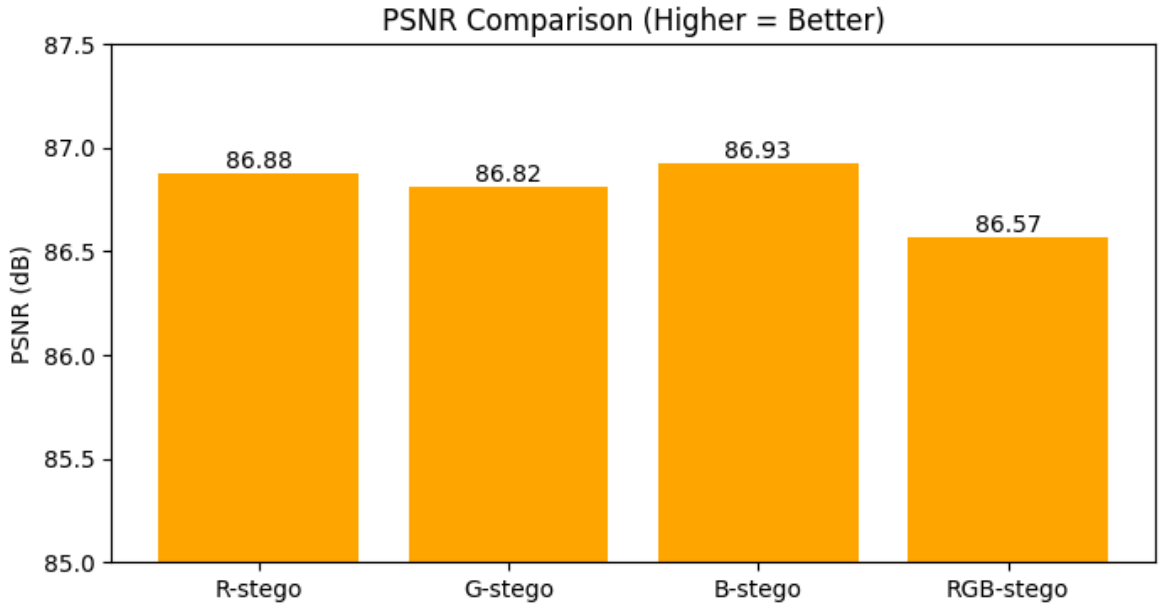
4.2.3 AES Encryption in Dull Image:

	A	B	C	D	E
1	Image	Filename	MSE	PSNR (dB)	SSIM
2	R-stego	image_with_message.png	0.00013340816	86.87897964073886	0.9999996827504898
3	G-stego	image_with_message_G.png	0.00013537728	86.81534569558866	0.9999996771574714
4	B-stego	image_with_message_B.png	0.00013193132	86.92732466399106	0.9999996869920215
5	RGB-stego	image_with_message_RGB.png	0.00014325378	86.56974261729832	0.9999996608011624
6					
7					

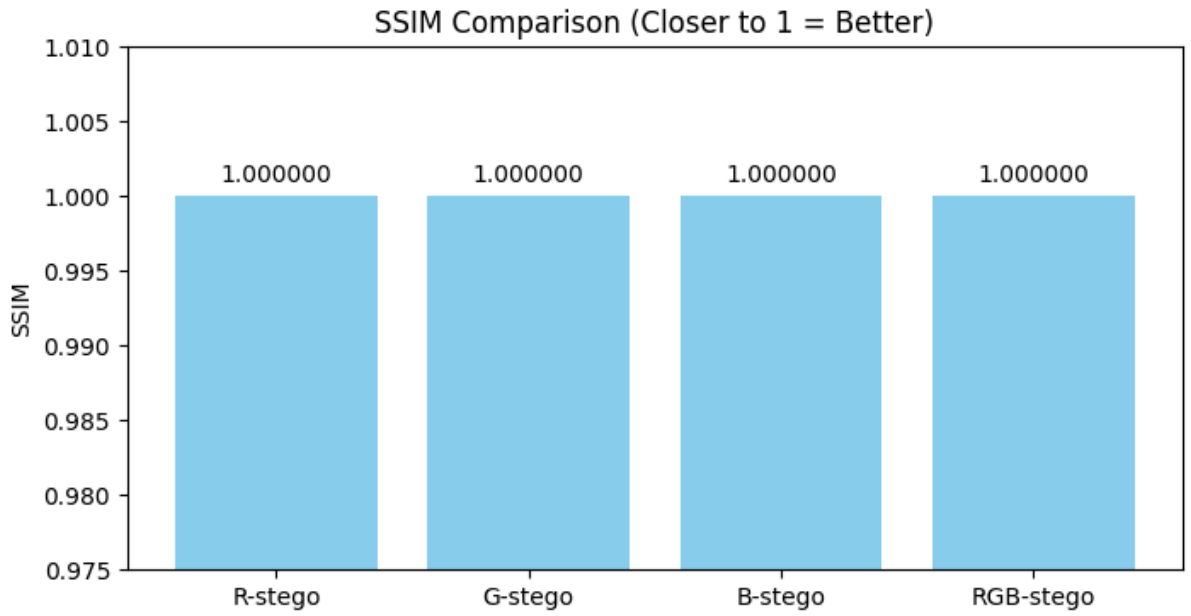
For dull images combined with AES encryption, the results are even more favorable than with bright images.



The MSE values decrease significantly, ranging between 0.00013 and 0.00014. These extremely small values reflect the minimal level of pixel modification, demonstrating the superior adaptability of dull images in concealing information.



Additionally, the PSNR values rise notably, reaching levels of about 86.5 to 86.9 dB. Such high values indicate that the visual quality of the stego image is exceptionally well preserved after embedding. This further emphasizes the strength of AES encryption when paired with dull images, as it ensures imperceptibility while maintaining outstanding image fidelity.

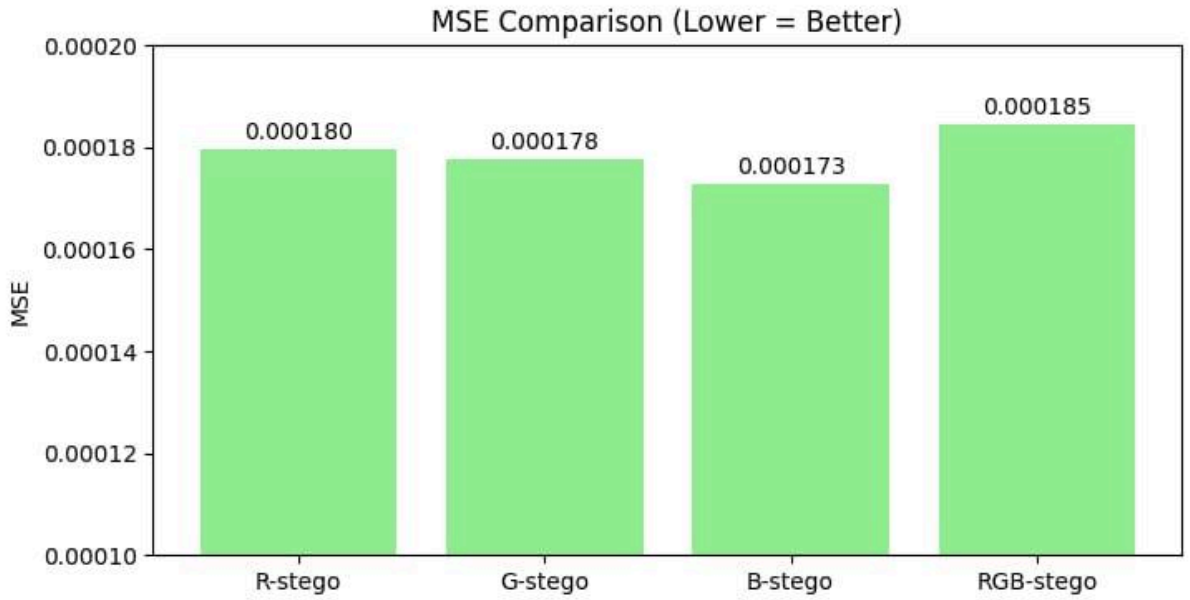


As with the other cases, the SSIM values remain close to 1 (0.9999996–0.9999997), confirming that the structural similarity between cover and stego images is virtually unchanged. These results suggest that dull images, combined with AES encryption, provide one of the most effective mediums for steganography, enabling secure data hiding with minimal distortion.

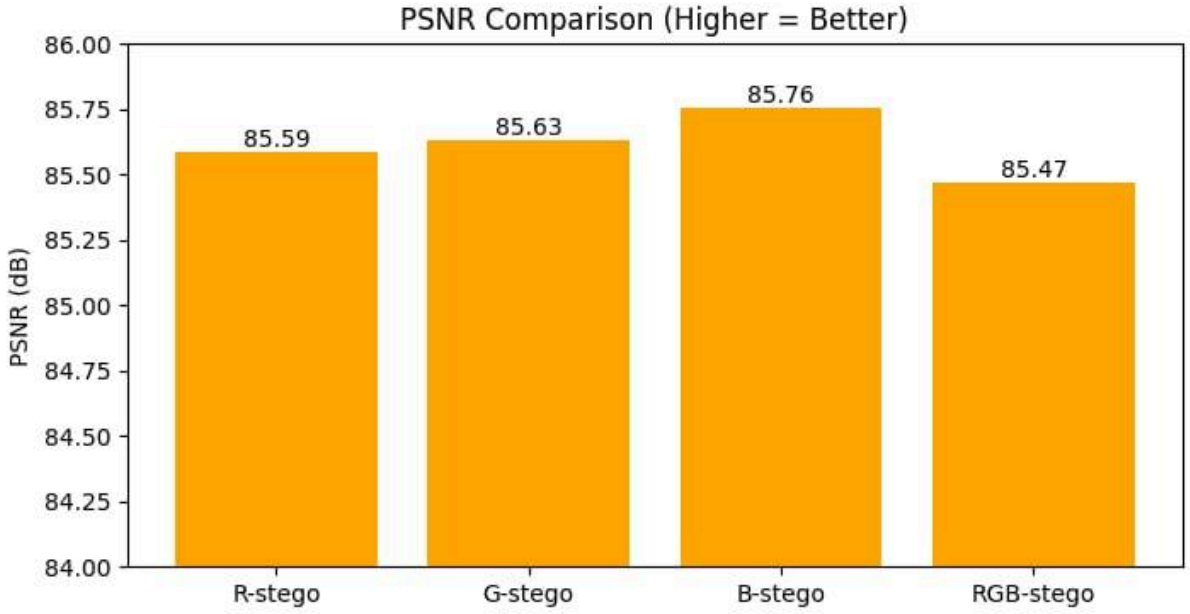
4.2.4 RSA Encryption in Dull Image:

	A	B	C	D	E
1	Image	Filename	MSE	PSNR (dB)	SSIM
2	R-stego	image_with_message (1).png	0.00017968258	85.58574381666344	0.9999996071958414
3	G-stego	image_with_message_G (1).png	0.00017771346	85.63360037689371	0.9999996150417719
4	B-stego	image_with_message_B (1).png	0.00017279065	85.7556012444631	0.9999996356229196
5	RGB-stego	image_with_message_RGB (1).png	0.0001846054	85.46835976988214	0.9999995642381116
6					
7					

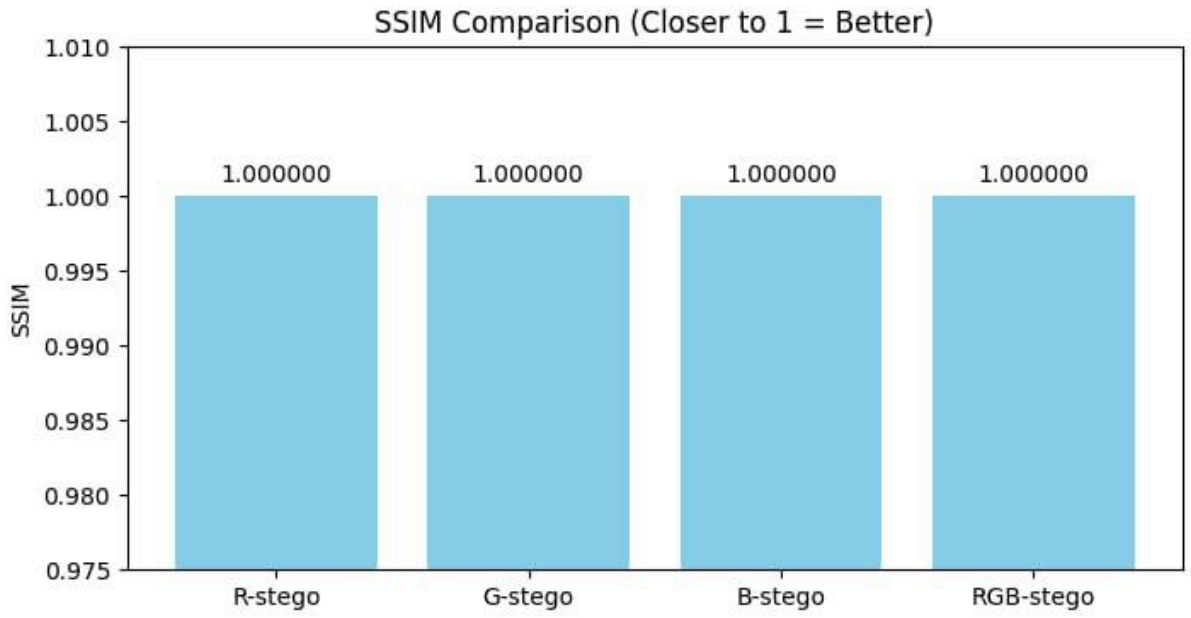
When RSA encryption is applied to dull images, the results remain strong but once again fall slightly short of AES performance.



The MSE values rise to around 0.00025–0.00029, suggesting a higher degree of distortion compared to AES but still maintaining very low levels overall. This demonstrates that RSA remains effective but less efficient in minimizing pixel-level modifications.



The PSNR values correspondingly decline, ranging from 83.4 to 83.9 dB. While this represents a drop from the dull image AES case, the values remain sufficiently high to guarantee that the stego images maintain very good perceptual quality. Compared to bright images, dull images still offer a stronger medium for embedding, even under RSA.



Finally, the SSIM values continue to hold close to 1 (0.9999995–0.9999997), ensuring that the structural integrity of the image remains intact. Although RSA encryption produces high-quality stego images, the comparative reduction in PSNR and increase in MSE make it evident that AES is superior. This confirms that AES encryption is more effective than RSA in retaining image quality, especially in dull images.

Comparative Analysis of Black & White Images in Crypto-Steganography

Black and white Image Version of Bright Image:



Black and white Image Version of Dull Image:

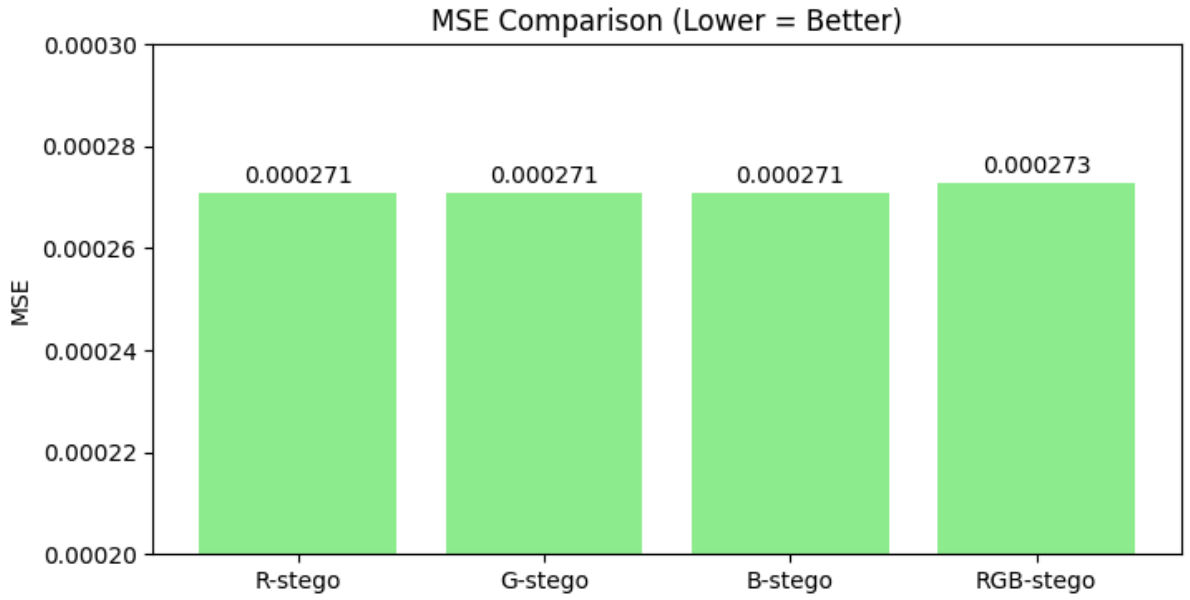


In this work, black & white versions of both bright and dull cover images were used to embed encrypted secret messages. Two encryption algorithms, Advanced Encryption Standard (AES) and Rivest–Shamir–Adleman (RSA), were applied separately before data embedding to compare their performance. The quality of the stego images was evaluated using three standard image quality metrics: Mean Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR), and Structural Similarity Index Measure (SSIM).

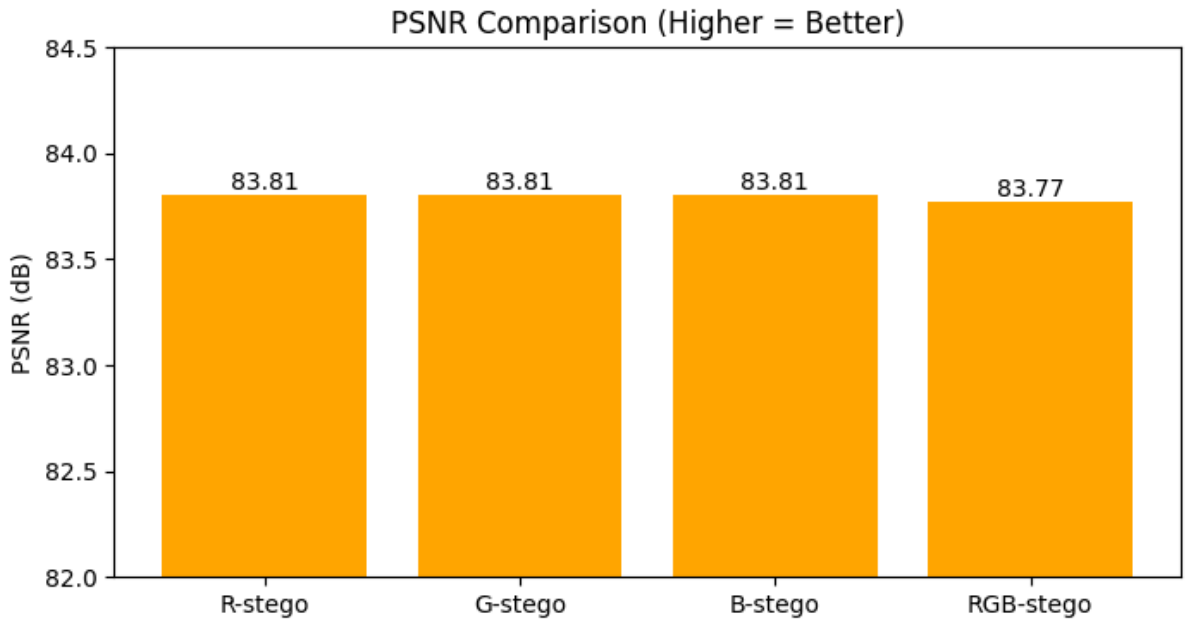
4.2.5 Black & White Bright Image with AES:

	A	B	C	D	E
1	Image	Filename	MSE	PSNR (dB)	SSIM
2	R-stego	image_with_message (2) (1).png	0.00027075456	83.80504573894885	0.9999997235368704
3	G-stego	image_with_message_G (2) (1).png	0.00027075456	83.80504573894885	0.9999997235368704
4	B-stego	image_with_message_B (2) (1).png	0.00027075456	83.80504573894885	0.9999997235368704
5	RGB-stego	image_with_message_RGB (2) (1).png	0.0002727237	83.77357479616735	0.9999997771025621
6					
7					

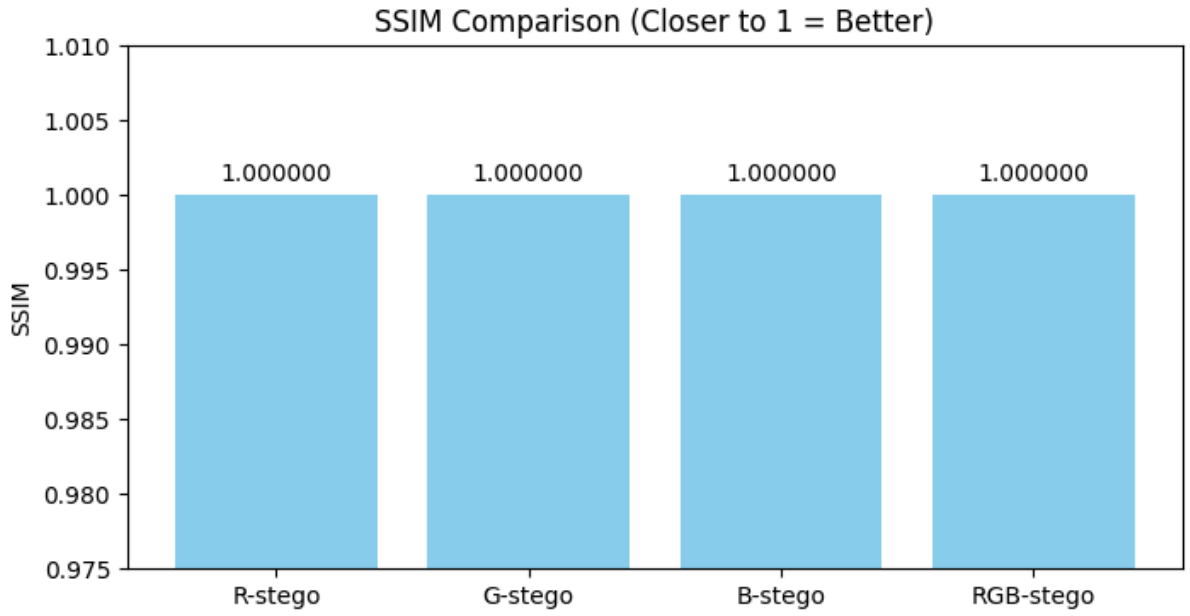
When AES encryption is applied to black & white bright images, the results demonstrate strong performance in maintaining image quality.



The MSE values for individual color channels (R, G, and B) remain very low at approximately 0.00027075, while the combined RGB channel slightly increases to 0.00027272, indicating minimal pixel-level distortion.



Correspondingly, the PSNR values are consistently high, with R, G, and B channels each at 83.805 dB and RGB at 83.774 dB, ensuring that the stego images retain excellent perceptual quality.

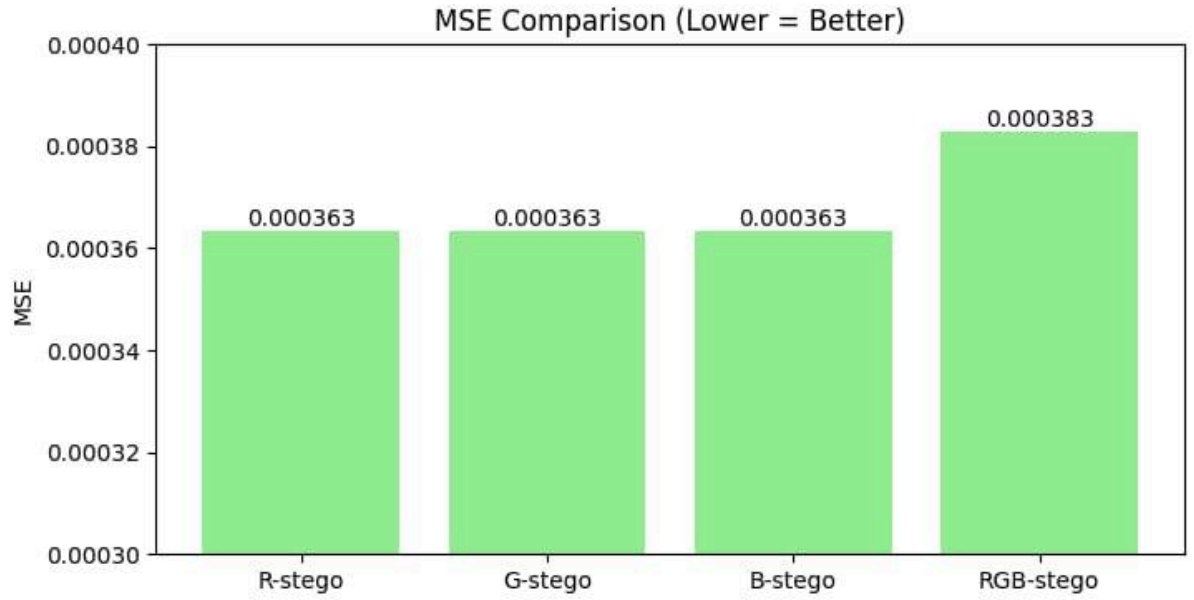


The SSIM values also remain extremely close to 1, ranging from 0.99999972 for individual channels to 0.99999977 for RGB, confirming that the structural integrity of the image is effectively preserved. These results indicate that AES encryption is highly effective for black & white bright images, maintaining both visual fidelity and structural consistency, with only negligible degradation in image quality.

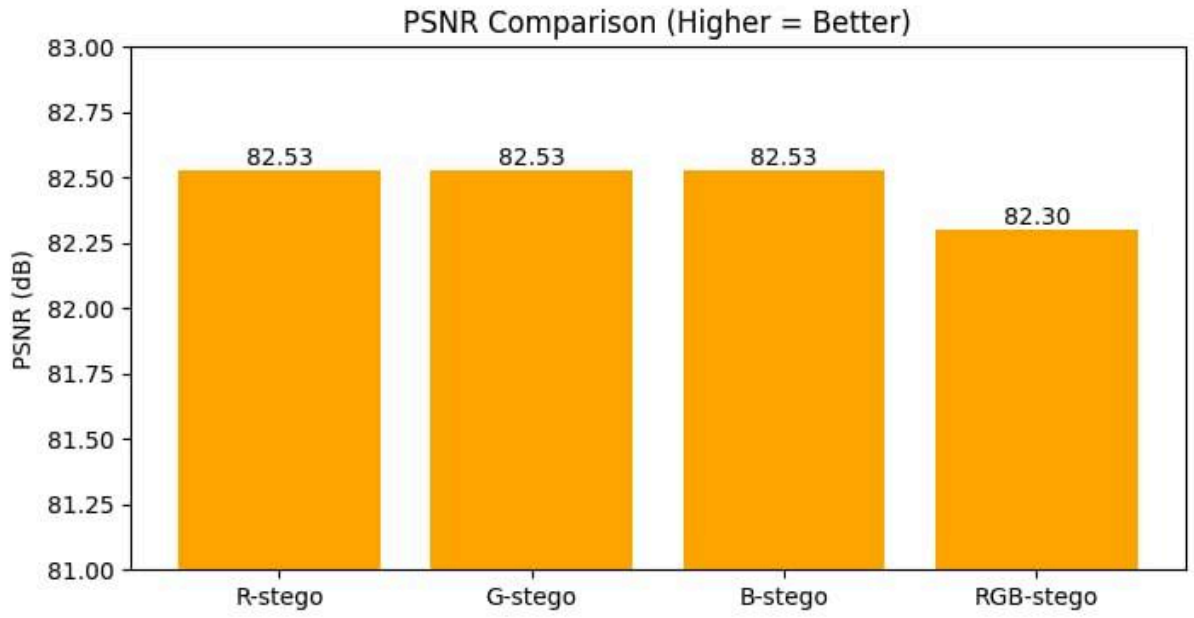
4.2.6 Black & White Bright Image with RSA:

	A	B	C	D	E
1	Image	Filename	MSE	PSNR (dB)	SSIM
2	R-stego	image_with_message (1).png	0.0003633034	82.52810890686041	0.999999512643314
3	G-stego	image_with_message_B (1).png	0.0003633034	82.52810890686041	0.999999512643314
4	B-stego	image_with_message_G (1).png	0.0003633034	82.52810890686041	0.999999512643314
5	RGB-stego	image_with_message_RGB (2).png	0.00038299465	82.29887652478547	0.9999996785858923
6					
7					

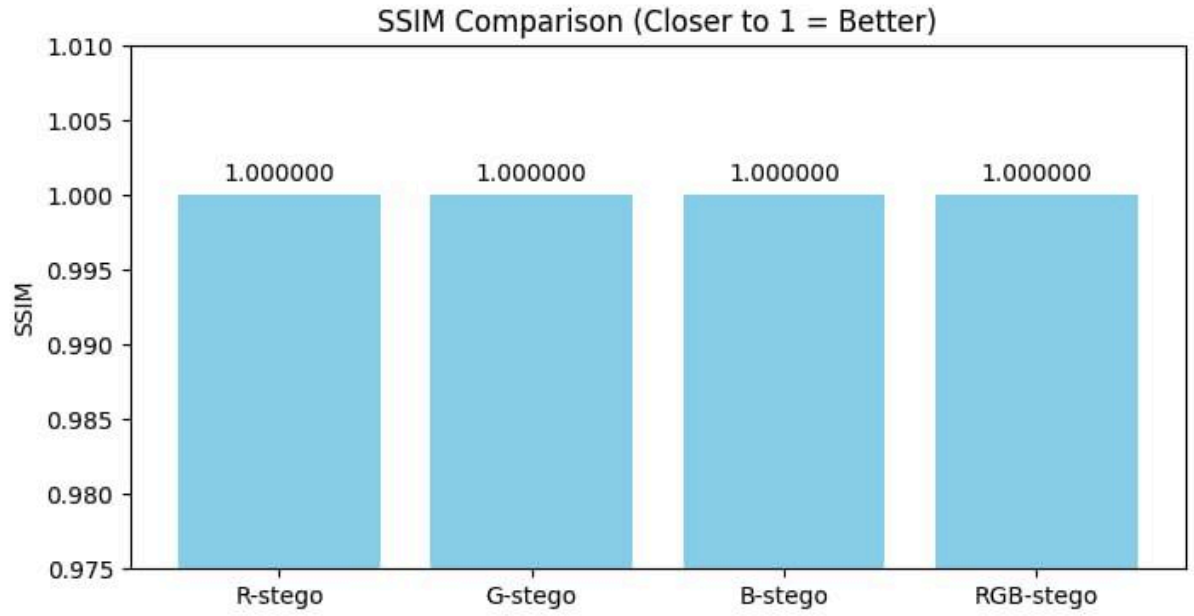
When RSA encryption is applied to black & white bright images, the results remain strong but again fall slightly short of AES performance.



The MSE values rise to around 0.0003633 for the individual channels and 0.0003830 for the combined RGB channel, indicating a higher level of pixel-level distortion compared to AES. This shows that RSA is effective but less efficient in minimizing changes to the image.



The PSNR values correspondingly decline, with individual channels at 82.528 dB and the RGB channel slightly lower at 82.299 dB. While this represents a noticeable drop from the AES case, the values are still high enough to ensure that the stego images maintain good perceptual quality.

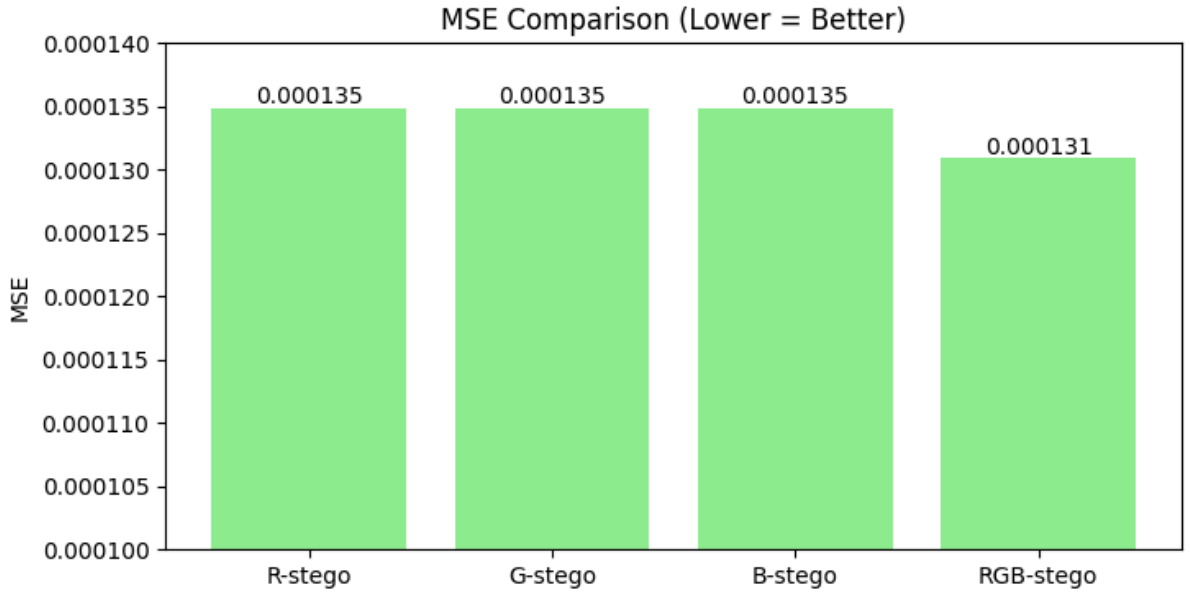


The SSIM values also remain close to 1 (0.99999951–0.99999968), confirming that the structural integrity of the images is preserved. Although RSA encryption generates high-quality stego images, the comparative reduction in PSNR and increase in MSE make it clear that AES performs better. This confirms that AES encryption is more effective than RSA in maintaining image quality, especially in bright black & white images.

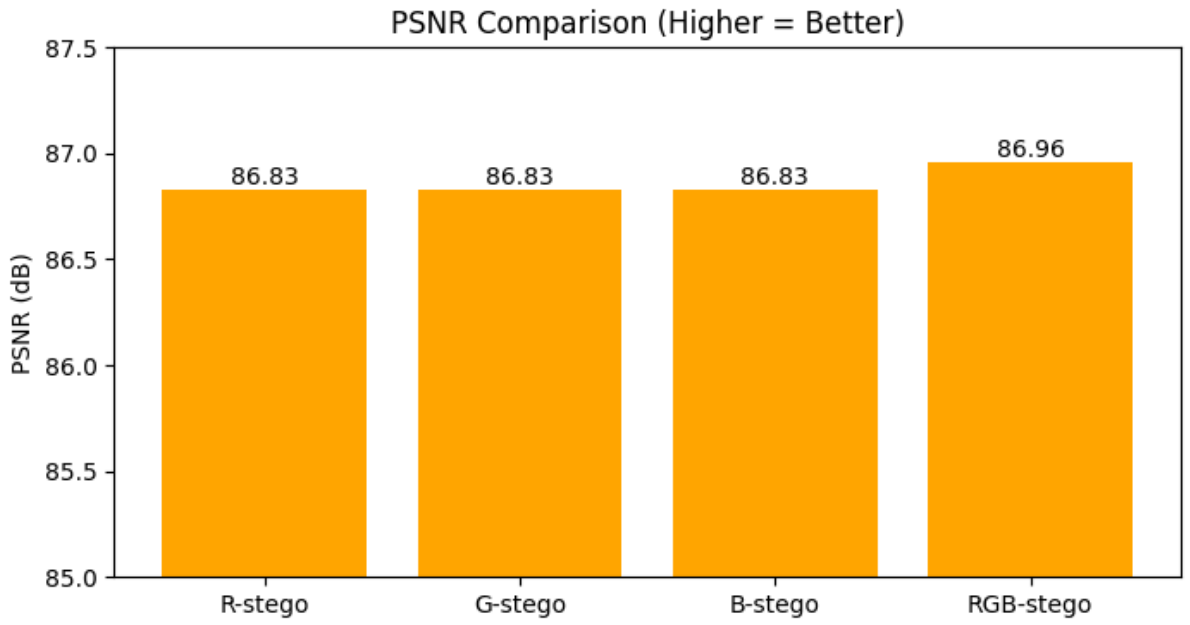
4.2.7 Black & White Dull Image with AES:

	A	B	C	D	E
1	Image	Filename	MSE	PSNR (dB)	SSIM
2	R-stego	image_with_message (3).png	0.000134885	86.83116686768294	0.9999996801223391
3	G-stego	image_with_message_G (3).png	0.000134885	86.83116686768294	0.9999996801223391
4	B-stego	image_with_message_B (3).png	0.000134885	86.83116686768294	0.9999996801223391
5	RGB-stego	image_with_message_RGB (3).png	0.00013094675	86.95985643606059	0.9999996901786967
6					
7					

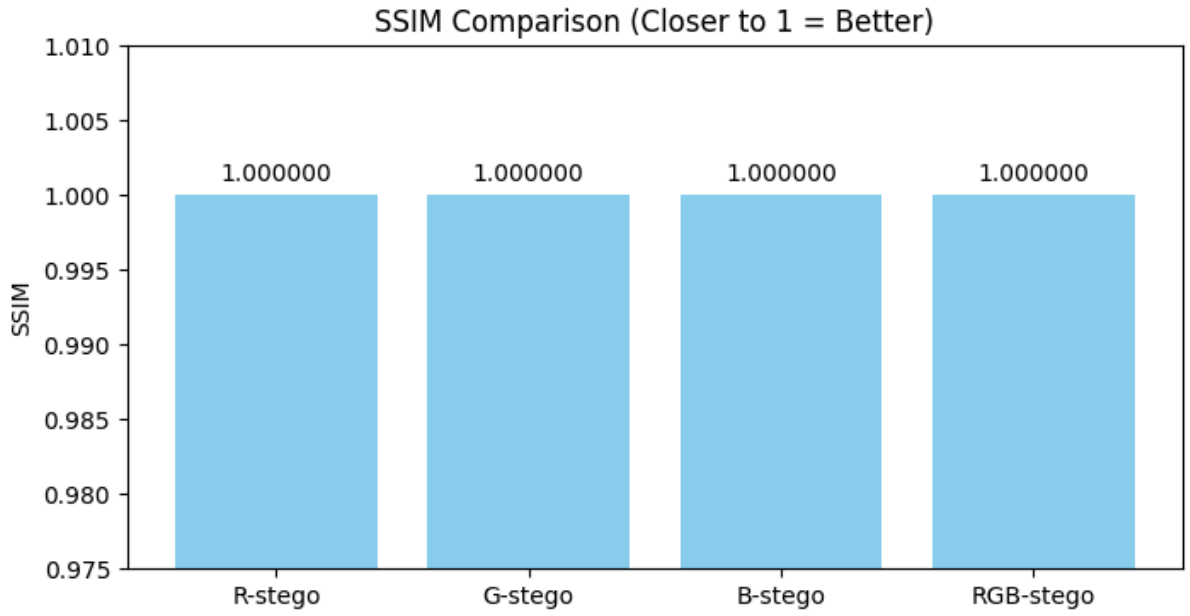
When AES encryption is applied to black & white dull images, the results remain very strong, showing minimal distortion.



The MSE values for the individual channels are consistently 0.000134885, while the combined channel slightly decreases to 0.00013094675, indicating an even lower level of pixel-level modification compared to bright images.



The PSNR values correspondingly remain very high, with individual channels at 86.831 dB and the combined channel at 86.960 dB, ensuring that the stego images retain excellent perceptual quality.

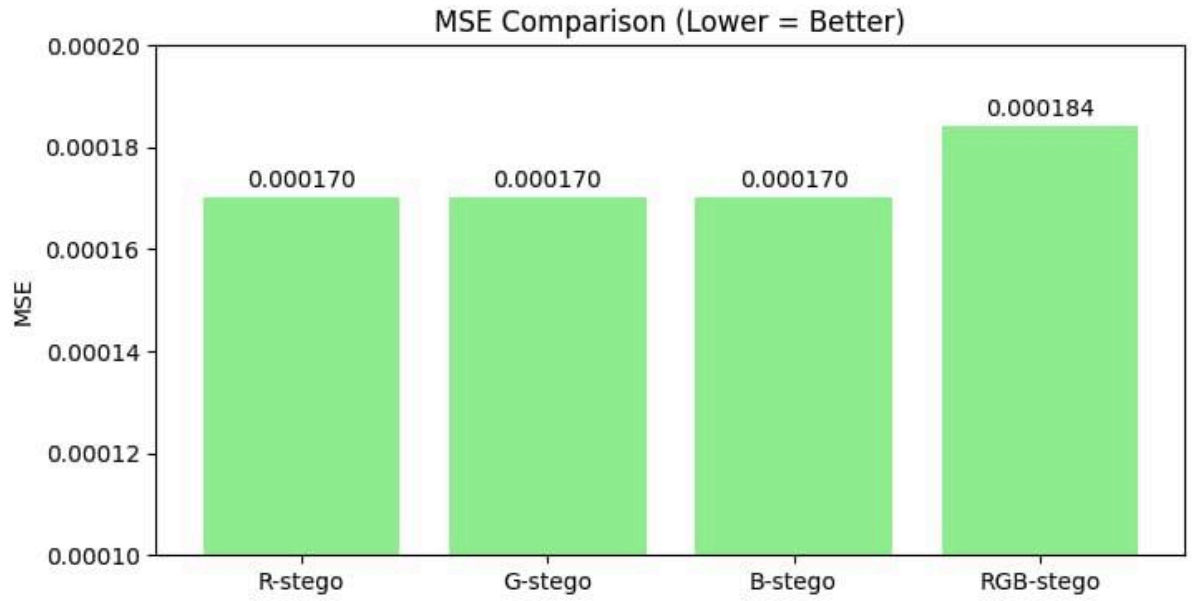


The SSIM values also stay extremely close to 1 (0.99999968–0.99999969), confirming that the structural integrity of the images is effectively preserved. Overall, AES encryption produces high-quality stego images for black & white dull images, maintaining both visual and structural fidelity with negligible degradation, highlighting its effectiveness for embedding in low-brightness images.

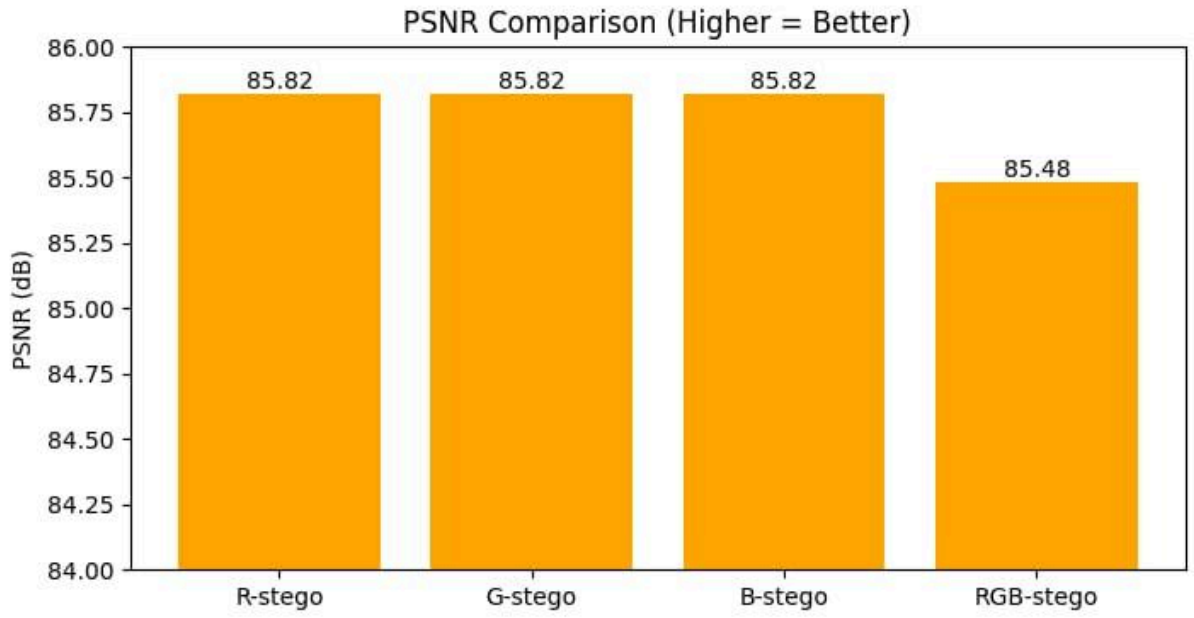
4.2.8 Black & White Dull Image with RSA:

	A	B	C	D	E
1	Image	Filename	MSE	PSNR (dB)	SSIM
2	R-stego	image_with_message (1).png	0.00017032924	85.81791161358225	0.9999996263600557
3	G-stego	image_with_message_B (1).png	0.00017032924	85.81791161358225	0.9999996263600557
4	B-stego	image_with_message_G (1).png	0.00017032924	85.81791161358225	0.9999996263600557
5	RGB-stego	image_with_message_RGB (1).png	0.0001841131	85.47995666680765	0.9999995661984015
6					
7					

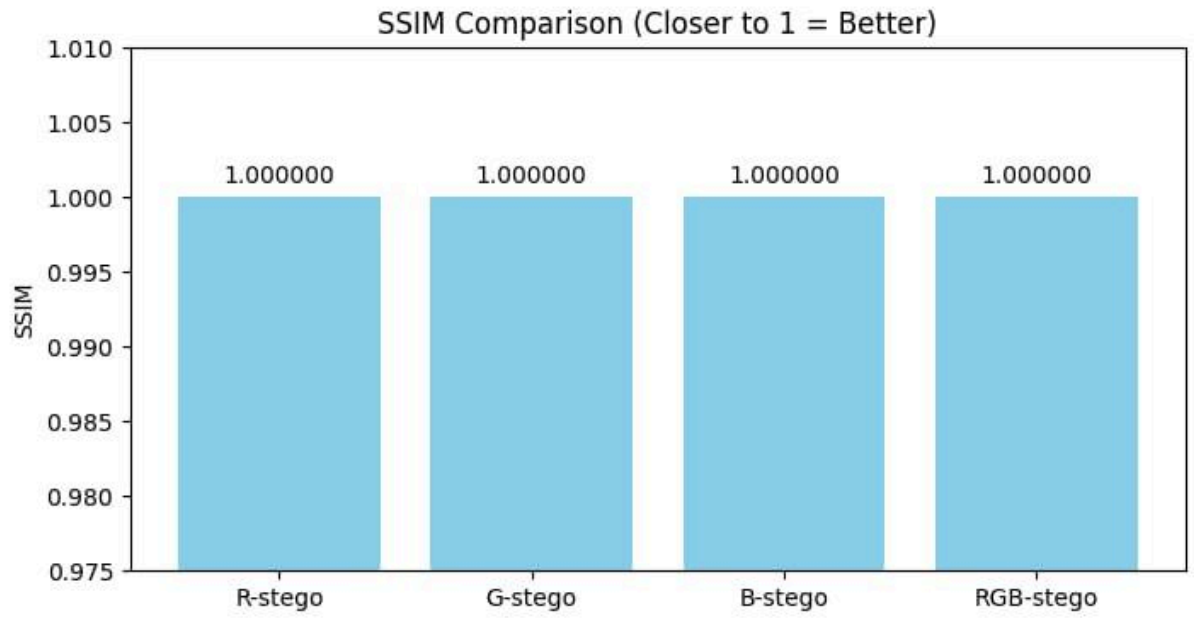
When RSA encryption is applied to black & white dull images, the results remain strong but once again fall slightly short of AES performance.



The MSE values rise to around 0.0001703 for the individual channels and 0.0001841 for the combined RGB channel, suggesting a higher degree of distortion compared to AES but still maintaining very low levels overall. This demonstrates that RSA remains effective but less efficient in minimizing pixel-level modifications.



The PSNR values correspondingly decline, with individual channels at 85.818 dB and the RGB channel slightly lower at 85.480 dB. While this represents a drop from the dull image AES case, the values remain sufficiently high to ensure that the stego images maintain very good perceptual quality.



The SSIM values continue to stay close to 1 (0.99999963–0.99999957), confirming that the structural integrity of the images remains intact. Although RSA encryption produces high-quality stego images, the comparative reduction in PSNR and increase in MSE make it evident that AES is superior. This confirms that AES encryption is more effective than RSA in retaining image quality, even for black & white dull images.

4.3 SUMMARY AND COMPARATIVE INSIGHTS

From all these eight cases, the **Best** Combination to use for Crypto-Steganography is AES encryption with dull images (both in original and in black & white). This combination produced the lowest MSE (~0.00013–0.00014) and the highest PSNR (~86.5–86.9 dB) while maintaining SSIM values extremely close to 1. The reason is twofold. AES produces compact ciphertext, which means less data is embedded compared to RSA, reducing pixel modifications and distortion. Dull images provide a smoother intensity profile than bright ones, so embedding data introduces less noticeable change, especially when converted to black & white, where fine variations are less detectable. Thus, AES with dull images achieves both strong imperceptibility and high image quality.

The **Worst** Combination to use is RSA encryption with bright images (in both original and black & white). Here, MSE was highest (~0.00035–0.00038) and PSNR lowest (~82.2–82.5 dB), though SSIM still remained close to 1. This happens because RSA ciphertext is much longer than AES, which increases the payload size, forcing more modifications in the cover image. And also, bright images are less tolerant to embedding since variations stand out more in high-intensity areas, making distortion slightly more visible. This explains why RSA with bright images consistently gave weaker performance compared to the other cases.

Dull images performed better than bright images in both AES and RSA cases. This is because dull images have lower contrast and intensity, allowing embedding changes to be better masked. Bright images showed slightly more distortion since embedding changes are more noticeable in higher intensity regions. AES is consistently superior to RSA. Its shorter ciphertext length means fewer embedding modifications, leading to lower MSE and higher PSNR. RSA, while secure, produces longer ciphertexts that require more embedding space, introducing greater distortion. Black & white images do not necessarily outperform originals, but they showed similar trends—AES still performed better than RSA, and dull images still outperformed bright ones. Conversion to black & white mainly reduces color redundancy, but it does not eliminate the inherent issue of RSA's longer ciphertext or the advantage of dull images.

In addition to brightness, encryption type, and image version, the choice of embedding channel significantly influences steganographic performance. Among the individual channels, the Blue (B) channel consistently delivered the best results, with lower distortion (low MSE) and higher imperceptibility (high PSNR), since the human visual system is least sensitive to variations in blue intensity. The Green (G) channel offered moderate performance, while the Red (R) channel showed the weakest results, as the eye is more sensitive to red variations, making distortions more detectable. However, the RGB combined channel outperformed all individual channels, as the embedding load was distributed across all three color planes, thereby minimizing distortion in any single channel and achieving the highest PSNR and SSIM values. This demonstrates that while blue is the best option for single-channel embedding, the RGB channel is the most effective overall, ensuring a balance between imperceptibility and data-hiding efficiency.

Overall, the experiments clearly establish that the choice of encryption method, cover image characteristics, and embedding channel all significantly affect the performance of crypto-steganography. AES consistently produced higher quality stego images than RSA because of its shorter ciphertext length, which reduces embedding distortion. Among cover images, dull images proved to be better carriers than bright images, as they mask modifications more effectively. Black & white conversion does not fundamentally improve results but maintains the same trend: dull images remain superior, and AES remains more efficient than RSA.

In addition to brightness and encryption type, the choice of embedding channel also strongly influences steganographic performance. Among individual channels, the Blue (B) channel consistently delivered the best results, with lower distortion (low MSE) and higher imperceptibility (high PSNR), since the human visual system is least sensitive to variations in blue intensity. The Green (G) channel offered moderate performance, while the Red (R) channel showed the weakest results, as the eye is more sensitive to red variations, making distortions more detectable. However, the combined RGB channel outperformed all individual channels, as the embedding load was distributed across all three color planes, minimizing distortion in any single channel and achieving the highest PSNR and SSIM values.

Therefore, the optimal configuration for crypto-steganography is AES encryption with a dull cover image and embedding in the RGB channel. This combination ensures minimal distortion, maximum imperceptibility, and the highest overall quality. Conversely, the least effective option is RSA encryption with a bright image using the Red channel, which introduces the most distortion due to RSA's longer ciphertext length, the lower tolerance of bright images, and the eye's sensitivity to red variations. This comparative analysis underscores the importance of carefully selecting encryption scheme, cover image, and embedding channel for effective and secure steganographic applications

References:

[1] Abid, Noman. "Evolution of Cryptographic Techniques: Overview of the Existing Approaches and Trends of the Development." Bullet: Jurnal Multidisiplin Ilmu, vol. 1, no. 3, June–July 2022, pp. 523–538.

[2] Abdulla, Alan Anwer. Exploiting Similarities between Secret and Cover Images for Improved Embedding Efficiency and Security in Digital Steganography. PhD thesis, The University of Buckingham, 2015.

[3] Abdelkader, Moumen, and Hocine Sissaoui. "Images Encryption Method Using Steganographic LSB Method, AES and RSA Algorithm." Nonlinear Engineering, vol. 6, no. 1, Jan. 2017, pp. 1–8. De Gruyter, doi:10.1515/nleng-2016-0010.

[4] Al-Najjar, Yusra. "Comparative Analysis of Image Quality Assessment Metrics: MSE, PSNR, SSIM and FSIM." International Journal of Science and Research (IJSR), vol. 13, no. 3, Mar. 2024, pp. 110–114. DOI:10.21275/SR24302013533.

[5] Bennett, Krista. Linguistic Steganography: Survey, Analysis, and Robustness Concerns for Hiding Information in Text. CERIAS Technical Report TR 2004-13, Purdue University, May 2004.

[6] Cheddad, Abbas, Joan Condell, Kevin Curran, and Paul Mc Kevitt. "Digital Image Steganography: Survey and Analysis of Current Methods." Signal Processing, vol. 90, no. 3, Mar. 2010, pp. 727–752. Elsevier, doi:10.1016/j.sigpro.2009.08.010.

[7] Hussain, Mehdi, and Mureed Hussain. "Survey of Image Steganography Techniques." International Journal of Advanced Science and Technology, vol. 54, May 2013, Shaheed Zulfiqar Ali Bhutto Institute of Science & Technology (SZABIST), Islamabad, Pakistan.

[8] Hussain, M., and M. Hussain. "Survey of Image Steganography Techniques." International Journal of Advanced Science and Technology, vol. 54, May 2013.

[9] Holdsworth, Jim, and Matthew Kosinski. "What Is Information Security (InfoSec)?" IBM Think; IBM.

[10] Hashim, Mohammed Mahdi, et al. "Performance Evaluation Measurement of Image Steganography Techniques with Analysis of LSB Based on Variation Image Formats." International Journal of Engineering & Technology, vol. 7, no. 4, 2018, pp. 3505–3514.

[11] Lu, Chun-Shien. Multimedia Security: Steganography and Digital Watermarking Techniques for Protection of Intellectual Property. IGI Global, 2005.

[12] Petitcolas, Fabien A. P., Ross J. Anderson, and Markus G. Kuhn. "Information Hiding—A Survey." Proceedings of the IEEE, vol. 87, no. 7, 1999, pp. 1062–1078.

[13] Sara, Umme, Morium Akter, and Mohammad Shorif Uddin. "Image Quality Assessment through FSIM, SSIM, MSE and PSNR—A Comparative Study." Journal of Computer and Communications, vol. 7, no. 3, Jan. 2019, pp. 8–18. doi:10.4236/jcc.2019.73002.

[14] Singh, R., and S. Singh. "An Improved LSB Based Image Steganography Technique for RGB Images." International Journal of Computer Applications, vol. 140, no. 7, 2016, pp. 1–5.

