iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** VI **Month of publication:** June 2026

DOI: https://doi.org/10.22214/ijraset.2026.84020

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue VI June 2026- Available at www.ijraset.com

Multi-Modal Steganography for Secure Data Hiding across Diverse Digital Carriers

Prof. P. V. Sridevi[1], Allaka Nitin[2], Palla Yogith Yadav[3], Gandi Pavan Naga Venkatesh[4], Pillala Sridevi[5]
Department of Electronics and Communication, Andhra University College of Engineering, Andhra Pradesh, India

Abstract: ***Secure communication today must satisfy two fundamental objectives simultaneously: the content of a message must remain protected, and the communication itself should not appear suspicious. Conventional cryptographic methods such as AES, RSA, and DES are highly effective at protecting data confidentiality, but they leave encrypted traffic visible and therefore identifiable. In many environments, this visibility alone is sufficient to attract monitoring, traffic analysis, blocking, or even forced disclosure. Steganography addresses this limitation by concealing the existence of the message itself within ordinary digital carriers. This report presents a multi-modal steganography framework for secure data hiding across images, audio files, PDFs, and videos. The system follows a layered design. First, the payload is encrypted using AES-256-GCM, providing confidentiality, integrity, and authentication. Second, the encrypted bitstream is embedded using a hybrid architectural approach: spatial and temporal randomized least significant bit (LSB) substitution, driven by a pseudo-random number generator seeded from secret material, is applied to continuous media streams (images, audio, and video), whereas structural object stream parsing and metadata injection are employed for PDF document carriers. Third, the receiver extracts the embedded payload, verifies the authentication tag, and decrypts the message only when the correct key is provided. This design reduces the effectiveness of basic steganalysis because the hidden data is distributed non-sequentially, and the recovered payload remains unintelligible without the encryption key. The paper further discusses carrier suitability, implementation choices, capacity analysis, and evaluation methodology. Performance assessment is based on PSNR, SSIM, MSE, SNR, payload capacity, and detectability considerations.***

Keywords: Steganography, AES-GCM, Multi-modal Data Hiding, PBKDF2, Least Significant Bit (LSB), Cryptography.

I. INTRODUCTION

Digital communication has made the transmission of confidential information fast, cheap, and convenient, but that convenience has also created a persistent exposure problem. Files move through public networks, cloud servers, mobile devices, collaboration platforms, and shared storage systems, so sensitive information is no longer confined to closed, physically protected links. Financial records, health reports, design documents, command messages, legal drafts, and private conversations are now routinely sent through channels that may be monitored or logged. In such conditions, security must be treated as more than a question of unreadable content; it must also be treated as a question of invisibility. Cryptography solves the first part of the problem. When a message is encrypted, the data is transformed into ciphertext, which cannot be interpreted without the correct key. That protection is essential, but it does not hide the fact that a communication occurred. Encrypted traffic often has a recognizable structure, a measurable entropy profile, or a transfer pattern that may still invite suspicion. In high-risk environments, that suspicion alone may be harmful. A message that looks secret can be monitored more closely than a message that looks ordinary. Steganography offers a different property. Instead of making the message unreadable, it makes the message difficult to notice in the first place. A secret bitstream can be hidden in a carrier that already appears normal to an observer, such as an image, an audio clip, a video sequence, or even a document file. When done carefully, the carrier still looks, sounds, or behaves like the original file. This makes steganography especially useful when the goal is not merely secrecy, but discretion. Existing secure data transfer systems often depend entirely on encryption. That approach is useful but incomplete. It leaves several weaknesses: the message remains visible as suspicious ciphertext, traffic analysis may reveal patterns of communication, and a naïve hiding method may create statistical anomalies that can be detected. In addition, many systems support only one carrier type, which limits practical deployment. A user may have an image carrier available in one case, an audio carrier in another case, or a document or video file in yet another case. A security framework that is tied to one medium is therefore restrictive. Another problem lies in embedding strategy. Sequential LSB insertion is simple and efficient, but if it is used carelessly, it creates predictable bit patterns that can be attacked by histogram analysis, chi-square tests, or direct image/audio inspection. The challenge is not only to hide data; it is to hide data in a way that does not produce a clear statistical footprint and does not degrade the carrier beyond acceptable limits.

To address these issues, this work proposes a multi-modal secure data hiding framework built on a hybrid embedding architecture. The design begins with AES-256-GCM authenticated encryption so that the payload remains unreadable even if it is discovered. Depending on the structural characteristics of the selected carrier, the encrypted payload is embedded using different techniques: continuous media streams (images, audio, and video) employ randomized spatial or temporal least significant bit (LSB) placement driven by a pseudo-random number generator, whereas PDF document carriers utilize structural modifications within internal metadata fields and object streams. A capacity checker prevents overfilling a carrier, and a carrier recommender assists users in selecting an appropriate file format based on payload size and quality constraints. The framework is intentionally multi-modal rather than being restricted to a single media type, reflecting the diverse requirements of practical covert communication. Images are well suited when small file size and ease of sharing are important and audio files are preferable when the information is transmitted through multimedia systems. PDFs provide an effective medium for document-centric communication, while video files offer high embedding capacity and improved concealment through temporal redundancy, therefore the framework combines strong cryptographic protection, carrier flexibility, and hybrid embedding techniques within a unified secure data hiding pipeline.

II. CRYPTOGRAPHIC FOUNDATION

A. *Why Encryption Must Precede Steganography*

Steganography by itself hides the presence of data, but it does not secure the meaning of that data if the hidden bits are extracted. A well-designed covert communication system therefore uses a layered strategy: encryption first, hiding second. This principle is a practical example of defense in depth. Even if the embedding method is weakened, the attacker still sees ciphertext, not plaintext. In contrast, if encryption were omitted, a successful extraction attack would immediately reveal the secret message. The reverse ordering is not appropriate. If unencrypted data is embedded directly, the recovery of hidden bits becomes equivalent to message disclosure. That would defeat the purpose of using steganography in the first place. By encrypting the payload before embedding, the system ensures that each layer protects the next.

B. *AES-256-GCM*

The proposed framework uses AES-256 in where AES provides strong symmetric encryption, while GCM adds authentication through a message integrity tag. This combination is useful because it protects both secrecy and tamper detection. If a single bit is modified in the ciphertext or associated data, tag verification fails and the receiver rejects the message. This makes active attacks much harder than in encryption-only schemes. The encryption process can be summarized as follows. Let P_i represent the i-th plaintext block, IV represent the initialization vector, and $counter_i$ represent the counter value for block i. The ciphertext block C_i is produced by XORing the plaintext block with the AES-encrypted counter stream:

$$C_i = P_i \oplus E_k(IV \parallel counter_i)$$

where Pi denotes the i-th plaintext block, Ci is the corresponding ciphertext block, CTRi represents the counter block associated with block i, and Ek(·) denotes the AES encryption function under the symmetric key k. AES-GCM performs encryption using Counter (CTR) mode, while authentication is provided separately using the GHASH function. After encryption, GCM computes an authentication tag using GHASH over the ciphertext and any associated data. If the tag at the receiver does not match the recalculated value, the message is treated as invalid and is not released. This property is especially valuable in covert communication, because the system must not only hide data but also detect silent corruption. Without authentication, an attacker could alter the hidden stream in a way that changes the final decoded message. GCM closes that gap.

C. *Key Derivation*

For passphrase-based workflows, the secret key should not be generated directly from a human password. Human passwords are usually too short and too predictable. Instead, a key derivation function such as PBKDF2 is used with a salt and a large number of iterations. The salt defeats precomputed rainbow-table attacks, while the iteration count increases the cost of brute-force guessing. In practice, this means that the same password will not produce the same key across different sessions unless the same salt is intentionally reused, which should not be done here. The derived key should then be used only in the secure communication workflow. Reusing it for unrelated applications is a poor security practice because it expands the attack surface without improving the system.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue VI June 2026- Available at www.ijraset.com

TABLE I. SECURITY LAYER COMPARISON

TECHNIQUE	MAIN FUNCTION	STRENGTH	LIMITATION
Cryptography only	Protects message content	Strong confidentiality	Encrypted traffic is visible
Steganography only	Hides message presence	Low suspicion when done well	Hidden data is readable if extracted
AES-256-GCM + Steganography	Protects content and presence	Layered defense and tamper detection	Requires careful implementation

III. STEGANOGRAPHIC METHODOLOGIES

A. Least Significant Bit Embedding

The primary hiding method used for continuous media streams (images, audio, and video) in this framework is least significant bit (LSB) substitution. Digital media store samples as binary values, and the least significant bits usually have the smallest visible or audible effect when modified carefully. For example, changing a pixel sample from 10110110 to 10110111 alters only the final bit. In many cases this change is imperceptible to the human eye. That makes LSB embedding attractive for covert communication because it is simple, efficient, and easy to apply across several media types. The generic embedding equation can be expressed mathematically as:

$$x_i' = 2^k \left\lfloor \frac{x_i}{2^k} \right\rfloor + m_i$$

where x_i is the original carrier sample, $x_i'$is the modified (stego) sample, $k$denotes the number of least significant bits used for embedding, and $m_i$represents the k-bit message fragment. This formulation replaces the $k$least significant bits while preserving all higher-order bits of the original sample. For the special case of single-bit LSB embedding ($k = 1$), the equation simplifies to:

$$x_i' = 2\left\lfloor \frac{x_i}{2} \right\rfloor + m_i, \text{ where } m_i \in \{0,1\}$$

This operation replaces only the least significant bit of the carrier sample while preserving all remaining higher-order bits. When k = 1, the method prioritizes invisibility. When k is increased, payload capacity improves, but distortion also increases. The framework therefore keeps k small in most cases and allows the carrier type to determine the practical ceiling. However, this pixel- and sample-level bit replacement technique is applicable only to continuous media streams. Structured document carriers such as PDFs do not rely on LSB substitution; instead, they employ structural object manipulation and metadata-based embedding to preserve document integrity while concealing the payload. A careful implementation must also consider sample distribution. Blindly replacing the same bit position in every sample can create a detectable pattern. That is why this project does not rely on simple sequential insertion.

B. Randomized Embedding Using PRNG

Sequential insertion is a known weakness. When bits are placed in order, the payload occupies a predictable region of the carrier. Statistical steganalysis tools exploit that regularity. To reduce this risk, the framework seeds a pseudo-random number generator with secret material derived from the encryption key or a related secret. The PRNG then selects embedding locations in a randomized order. To an observer without the key, the modified positions appear irregular and far less informative. The benefit is not magic; it is unpredictability. If the attacker cannot predict where the bits are stored, localized analysis becomes much less effective. Randomized placement also helps distribute the modification load more evenly across the carrier, which can preserve visual and auditory quality better than dense insertion into one region. For security and correctness, the PRNG seed must be protected as carefully as the encryption key itself. If the seed is exposed, the embedding pattern becomes recoverable. In other words, the randomization layer is only as strong as the secrecy of the seed material.

C. Carrier-Specific Notes

Image carriers are convenient because image files are common, easy to share, and tolerant of small bit-level changes, especially in lossless formats such as PNG or BMP. Audio carriers are useful because minor sample modifications in time-domain audio can remain inaudible when kept within a tight distortion budget. PDF carriers differ fundamentally from matrix-based media because they provide a structured hiding space consisting of metadata fields, object streams, hidden references, and other non-display elements. Consequently, rather than relying on least significant bit (LSB) substitution, the framework embeds data through structural object stream parsing and metadata modification without affecting the visible rendering of the document. Video carriers provide the largest practical capacity because they combine frame-based spatial redundancy with temporal redundancy across time.

Each carrier, however, behaves differently. The same embedding strategy does not produce the same quality outcome in every format. A JPEG image may compress away naïvely embedded LSB modifications if the workflow is not designed around the compression process. A lossy audio codec may similarly alter the embedded stream. For that reason, carrier selection must be matched to the payload size, target quality, and the intended delivery path.

TABLE II. CARRIER CHARACTERISTICS AND SUITABILITY

Carrier	Strength	Best Use	Main Caution
PNG / BMP Image	Lossless and stable	Small to medium secret payloads	File size may increase
WAV Audio	High sample precision	Audio-based covert transfer	Must preserve audible quality
PDF Document	Flexible internal structure	Textual or office-based exchange	Need careful object handling
MP4 / AVI Video	High practical capacity	Large payloads and streaming use	Encoding changes can disturb bits

IV. MULTI-MODAL CARRIER IMPLEMENTATION

A. Image Steganography

Image-based embedding is the most intuitive form of steganography because the human visual system is forgiving of tiny numerical changes, particularly in textured regions. The framework supports lossless carriers such as PNG and BMP because they preserve embedded bits more faithfully than many lossy workflows. RGB channels can be treated as three separate reservoirs, which means the payload can be spread across channels to reduce local distortion. A practical implementation first converts the cover image into a matrix of pixel values, then chooses positions according to the PRNG sequence, and finally replaces the selected least significant bits. If the image is too small for the payload, the system must reject the request before embedding starts. This precheck prevents damage to the carrier and avoids partial failure states. The visual quality of the stego image can be assessed by comparing the original image, the stego image, and the corresponding error map. Such comparisons help illustrate whether the embedding process introduces noticeable visual artifacts.

B. Audio steganography

Audio-based hiding works on the same principle, but the perceptual limit is different. Humans are sensitive to abrupt distortion and narrowband noise, so the embedding depth should remain low. WAV is usually the cleanest choice because it preserves samples without additional lossy compression. In contrast, MP3 or other lossy compression formats can complicate exact bit recovery unless the embedding strategy is adapted to the codec behavior. The safest approach is to modify time-domain samples with conservative bit depth and distribute the changes across the waveform. The carrier remains perceptually unchanged when the modifications are sparse and randomized. For evaluation, the recovered audio should sound identical in casual listening, and the measured SNR should stay comfortably high. Waveform and spectrogram analyses were used to evaluate the similarity between the original and stego audio signals and to verify that the embedding process preserved perceptual audio quality.

C. PDF Steganography

PDF files offer a different type of hiding space. They are not just printed pages; they are structured containers with metadata fields, hidden object references, streams, and elements that need not appear in the visible rendering. This allows the framework to hide information in places that do not affect the visible appearance of the document. Because many users exchange PDFs in professional workflows, this carrier can be especially useful when the hidden communication must blend into ordinary document transfer. The technical challenge is to avoid breaking the document structure. Any hidden object must remain valid under PDF parsing, and the visible rendering must stay unchanged.

A broken PDF is not covert; it is simply suspicious. Therefore, extraction and insertion routines must preserve syntax and object integrity. Visual rendering analysis and metadata inspection were performed to verify that the embedded payload remained hidden within non-visible PDF structures while preserving the document's original appearance.

D. *Video Steganography*

Video carries the largest practical payload because it combines many frames over time. If the framework selects frames with high motion or textured regions, minor bit-level changes become harder to notice. Video can also hide information either in frame samples or in the audio track, which makes it the most flexible carrier in the system. This is useful when the secret payload is large and the communication channel can tolerate heavier file sizes. Video embedding must be more cautious than image embedding because a video file may be reencoded during transmission. The framework should therefore prefer settings and formats that preserve the embedded data adequately, and it should report a warning when the selected carrier is unsuitable for the payload. A successful implementation should show that the visual quality remains acceptable and that the retrieval path reconstructs the ciphertext without corruption. Frame-level comparisons were performed to assess visual transparency and confirm that the embedding process introduced negligible perceptual distortion.

V. SYSTEM ARCHITECTURE

A. *Sender Pipeline*

The sender pipeline starts with user input. The user selects the secret message or file and chooses the carrier medium. The system then checks capacity before any embedding is attempted. That step is important because embedding should never begin if the payload is too large for the carrier. A clean rejection is better than a corrupted output. Once capacity is confirmed, the payload is encrypted with AES-256-GCM. The resulting ciphertext and authentication tag are encoded into a bitstream. Randomized placement is then used to insert the bits into the carrier. The final output file appears normal in routine use, which is the central requirement of a covert channel. A good implementation also logs the carrier type, selected bit depth, payload size, and resulting output size. These metadata are useful for evaluation and debugging, though they should not be exposed in the final public stego-file.

B. *Receiver Pipeline*

The receiver takes the stego-file and applies the inverse process. It extracts the randomized bit positions using the same secret seed, rebuilds the ciphertext, verifies the authentication tag, and then decrypts the payload. If the tag verification fails, the system must reject the message immediately rather than trying to guess the content. That behavior protects integrity and reduces the risk of silently accepting altered data. The decryption stage also provides a form of plausible deniability. Without the correct key, the receiver does not obtain useful plaintext. This is an important feature in scenarios where possession of a file should not automatically prove knowledge of its hidden content. Fig. 1 illustrates the overall architecture of the proposed framework, including the sender pipeline, receiver pipeline, AES-256-GCM encryption stage, randomized embedding mechanism, payload extraction process, and authentication verification workflow.

Fig. 1. Overall architecture of the multi-modal secure data hiding framework.

VI. IMPLEMENTATION DETAILS

A. *Software Stack*

The framework prototype was constructed using Python (v3.9+) due to its native support for modular cryptographic operations via PyCryptodome and high-performance array manipulation via NumPy as Python has a vast variety of libraries for cryptography, multimedia processing, and user interface construction and other important aspects in the tool developed. A Streamlit front end keeps the workflow accessible and interactive, while PyCryptodome can handle AES-256-GCM and PBKDF2.

NumPy and OpenCV are useful for image processing, Pillow helps with image I/O, PyPDF2 supports document handling, and MoviePy, Librosa, and SciPy are convenient for video and audio operations. The exact library mix may vary according to the final implementation details, but the design should remain modular. Encryption should be isolated from carrier manipulation, and carrier manipulation should be separated from the user interface. That structure makes the system easier to test, easier to debug, and easier to extend later. Fig. 2 presents a detailed view of the sender and receiver pipelines, including encryption, randomized embedding, extraction, and authentication verification stages.

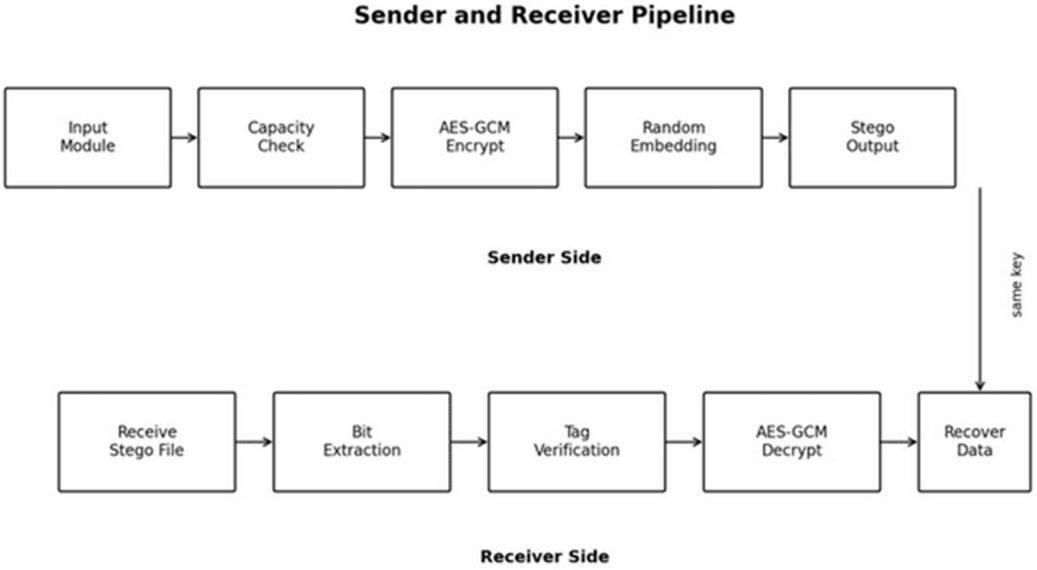


Fig. 2. Sender and receiver pipeline with encryption, randomized embedding, extraction, and verification.

B. Intelligent Carrier Recommender

The system architecture includes safeguards against optimization failures and capacity bounds. If the user tries to hide a large payload inside a small carrier, the system should not proceed blindly. Instead, it should estimate encrypted payload size, compare it against the safe carrier capacity, and recommend a better carrier if needed. This avoids corruption and improves the success rate of the workflow. For example, if the encrypted data is larger than the maximum safe embedding limit of the selected image, the system can recommend moving to a larger image, switching to an audio or video carrier, or using a more efficient payload strategy. In all cases, the key idea is to make capacity decisions before writing to the file. The result is a user-friendly system that behaves like a safeguard rather than a raw encoder. That matters in research prototypes because usability mistakes often look like algorithmic failures.

VII. RESULTS AND PERFORMANCE ANALYSIS

A. Experimental Overview

The developed framework was experimentally validated using four different carrier categories, namely images, audio signals, PDF documents, and video files. The primary objective of the evaluation was not limited to successful payload hiding alone; equal importance was given to visual transparency, extraction reliability, authentication integrity, and resistance against obvious statistical modification after embedding. During implementation, each carrier underwent AES-256-GCM encryption prior to embedding so that even if hidden data were partially exposed, the original payload would still remain computationally protected, therefore, instead of placing the encrypted bits sequentially, the framework has utilized a pseudo-randomized embedding strategy driven by a PRNG which is initialized from the secret key material. This approach distributed payload bits across non-consecutive carrier locations, reducing predictable embedding patterns that are commonly targeted during steganalysis. Randomized 1-LSB embedding was selected because it provided a practical balance between payload capacity as well as carrier preservation without introducing any noticeable degradation under normal viewing or playback conditions and to verify embedding quality, the original carriers and generated stego outputs were compared using standard fidelity and distortion metrics including PSNR, SSIM, MSE, SNR, SDR, STOI, BER, and payload recovery rate depending on the carrier modality. Alongside numerical evaluation, visual and spectral inspection was also carried out because perceptual similarity alone cannot always guarantee statistical transparency.

Histogram overlap analysis, waveform comparison, spectrogram consistency, and structural rendering inspection were therefore included as part of the overall validation procedure and the experimental observations showed that the framework maintained high carrier fidelity while successfully recovering the embedded payload with authentication verification enabled through AES-256-GCM tag validation. In almost all tested cases, the generated stego carriers remained visually or perceptually indistinguishable from their original counterparts during ordinary usage conditions.

B. Image Steganography Evaluation Metrics

The image steganography module was evaluated by embedding a compressed PDF payload within a lossless PNG carrier. The bits were distributed non-sequentially using the PRNG-driven mapping function. By using the PNG format, the system avoided any automatic quality loss, ensuring that spatial modifications remained below the human visual threshold and statistical steganalysis detection limits and specialized image metrics, such as Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM), confirmed that the image's structure, colors, and sharp edges remained virtually identical to the original, leaving behind no visual flaws or clues of any kind. Table III details the quantitative embedding and extraction performance parameters metrics.

TABLE III. Image Steganography Embedding and Extraction Results

S. No.	Parameter	Value	Good Range	Status
1	Carrier File	ImageCarrier.png	PNG lossless carrier	Valid
2	Stego File	secure_output_ImageCarrier.png	Successful output generation	Generated
3	Payload File	PDFpayload.PDF	Supported payload format	Embedded
4	Original Size	3614 KB	Large carrier preferred	Good
5	Stego Size	3631 KB	Minimal increase	Good
6	Payload Size	15.86 KB	< 703.12 KB	Safe
7	Safe Capacity	703.12 KB	> Payload size	Excellent
8	Embedding Technique	Randomized 1-LSB	Randomized preferred	Secure
9	Encryption	AES-256-GCM	AES-256 standard	Strong
10	PSNR	67.62 dB	> 40 dB	Excellent
11	SSIM	0.9999	≈ 1.0	Excellent
12	MSE	0.01	≈ 0	Excellent
13	Extraction Status	Successful	Successful required	Passed
14	Authentication Status	Verified	Verification required	Passed
15	Overall Result	Secure High-Fidelity Embedding	High transparency expected	Excellent

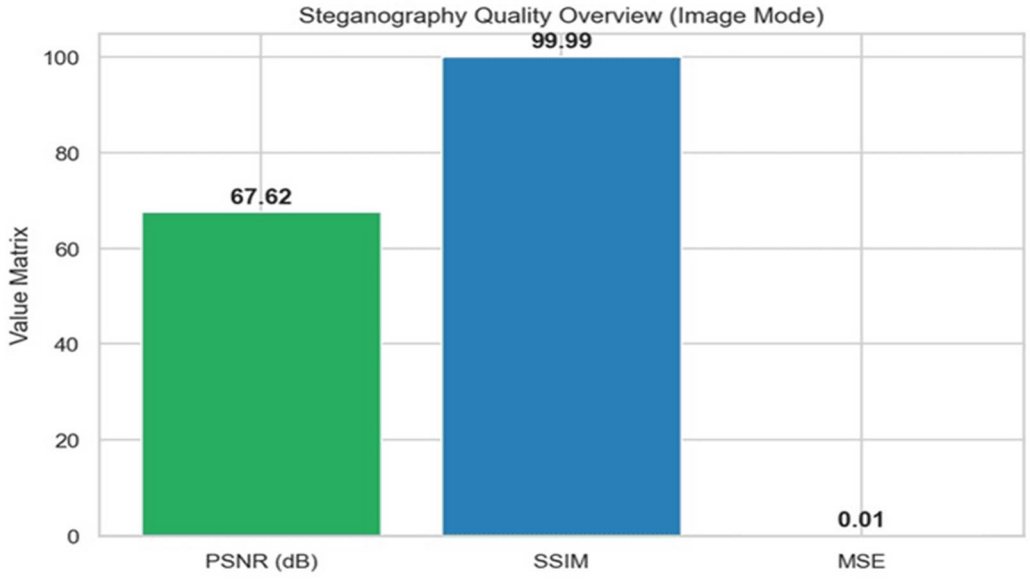


Fig. 3. Steganography Quality Overview for Image Carrier

The graph in Fig. 3 summarizes the quantitative quality measurements obtained after embedding the encrypted payload inside the file named ImageCarrier.png using randomized 1-LSB method of steganography. The measured PSNR value of 67.62 dB reflects extremely small pixel-level modification after embedding. Likewise, the SSIM score of 0.9999 indicates that the structural composition of the stego image remained almost identical to the original carrier. Since the MSE value remained close to zero as it should, the overall embedding distortion introduced into secure_output_ImageCarrier.png stayed practically imperceptible or very small in magnitude during normal viewing conditions.

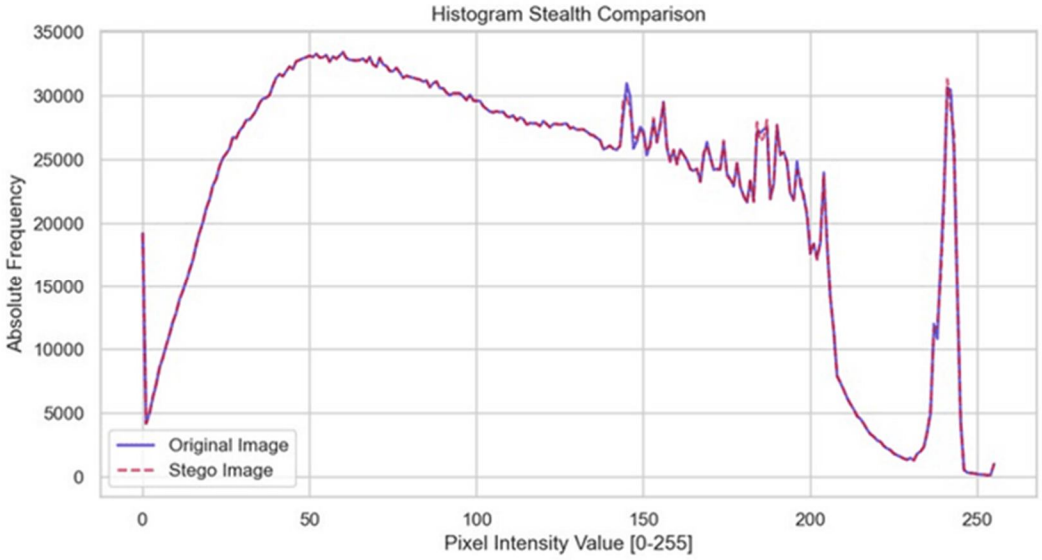


Fig. 4. Histogram Stealth Comparison between Original and Stego Image

The histogram comparison in Fig. 4 presents the pixel intensity distributions of the original carrier image file named ImageCarrier.png and the generated stego image file named secure_output_ImageCarrier.png. Both distributions exhibit strong overlap across most intensity levels, suggesting that the randomized embedding operation introduced only minor statistical variation. This behavior is important because large histogram deviations can expose hidden embedding patterns during steganalysis. In this case, the very close histogram similarity indicates that the embedding process preserved the natural statistical characteristics of the carrier image while securely storing the encrypted payload within the file.

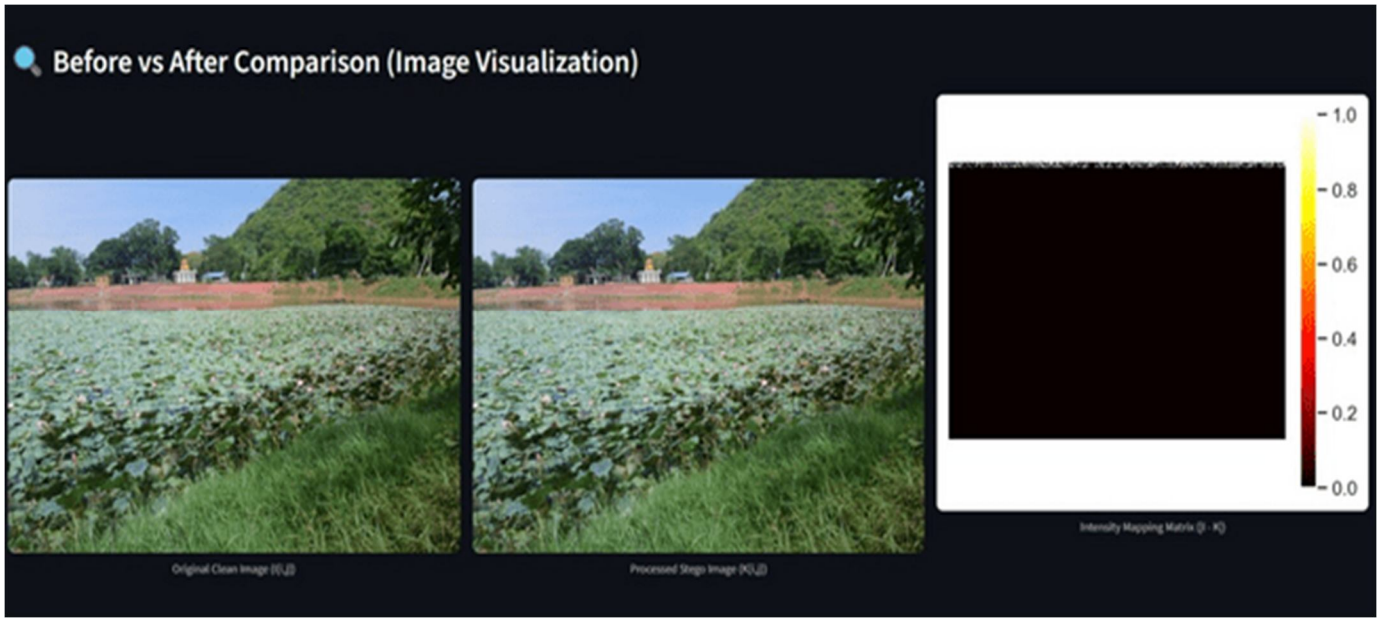


Fig. 5. Before vs After Image Comparison and Structural Error Mapping

This figure 5 illustrates the visual comparison between the original carrier image I[i,j], which ImageCarrier.png and the generated stego image K[i,j] which is secure_output_ImageCarrier.png and the corresponding intensity error map |I-K|. The visible similarity between both images confirms that the payload embedding operation did not introduce perceptible visual degradation into secure_output_ImageCarrier.png and because we hid information randomly, the resulting errors are scattered and barely noticeable. This makes it much harder for spatial analysis tools to detect that any modifications were made.

C. Audio Steganography Results

The audio steganography module was evaluated using the WAV carrier file AudioCarrier.wav, while the encrypted payload used during testing was ImagePayload.jpg. After successful embedding, the generated stego output was stored as secure_output_AudioCarrier.wav. WAV format was intentionally selected because uncompressed audio carriers allow direct sample-level manipulation without compression-induced modification that could corrupt embedded information and the embedding operation utilized randomized 1-LSB substitution across pseudo-random audio sample positions generated from secret key material. Instead of modifying adjacent samples sequentially, the payload bits were distributed throughout the carrier signal to reduce predictable alteration patterns and improve embedding stealth of the secret file. The evaluation focused on preserving perceptual audio quality while ensuring accurate extraction and successful AES-256-GCM authentication verification after payload recovery and signal quality analysis was performed using SNR, SDR, PSNR, STOI, BER, recovery rate, and spectral energy preservation metrics. The obtained results indicated that the embedded payload introduced only minimal noise relative to the original carrier power. In addition to numerical measurements, waveform inspection and spectrogram analysis were carried out to observe time-domain and frequency-domain transparency after embedding. Table IV below has the Audio Steganography Embedding and Extraction Results.

TABLE IV Audio Steganography Embedding and Extraction Results

S. No.	Parameter	Value	Good Range	Status
1	Carrier File	AudioCarrier.wav	WAV lossless carrier	Valid
2	Stego File	secure_output_AudioCarrier.wav	Successful output generation	Generated
3	Payload File	ImagePayload.jpg	Supported payload format	Embedded
4	Original Size	2.5 MB	Large carrier preferred	Good
5	Stego Size	2.53 MB	Minimal increase preferred	Good
6	Payload Size	135.94 KB	< 162.76 KB	Safe
7	Safe Capacity	162.76 KB	> Payload size	Excellent
8	Embedding Technique	Randomized 1-LSB	Randomized preferred	Secure
9	Encryption	AES-256-GCM	AES-256 standard	Strong
10	SNR	46.3 dB	> 30 dB	Excellent
11	SDR	46.3 dB	> 30 dB	Excellent
12	PSNR	90.4 dB	> 40 dB	Excellent
13	STOI	1.00	Close to 1.0	Excellent
14	BER	0.0%	Close to 0%	Excellent
15	Recovery Rate	100%	100% preferred	Successful
16	Spectral Energy Preservation	1.00	Close to 1.0	Excellent
17	Extraction Status	Successful	Successful recovery required	Passed

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue VI June 2026- Available at www.ijraset.com

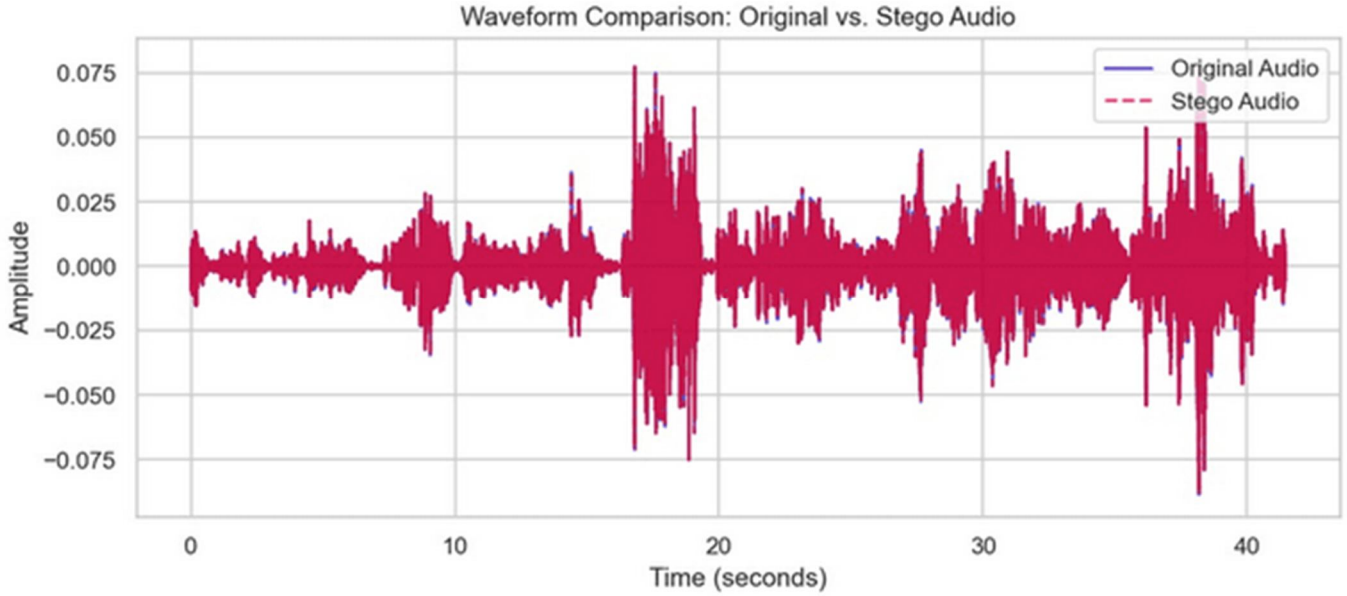


Fig. 6. Waveform Comparison of Original and Stego Audio

The waveform comparison in the Fig. 6, displays the time-domain representations of AudioCarrier.wav and secure_output_AudioCarrier.wav and across the entire signal duration, both waveforms remain nearly indistinguishable as they are almost identical, indicating that the randomized embedding process introduced extremely small amplitude variation into the carrier audio. The measured SNR and SDR values of 46.3 dB further support this observation because the embedded noise remained substantially lower than the original signal energy hence as a result, the stego audio preserved its natural playback characteristics without introducing noticeable audible distortion.

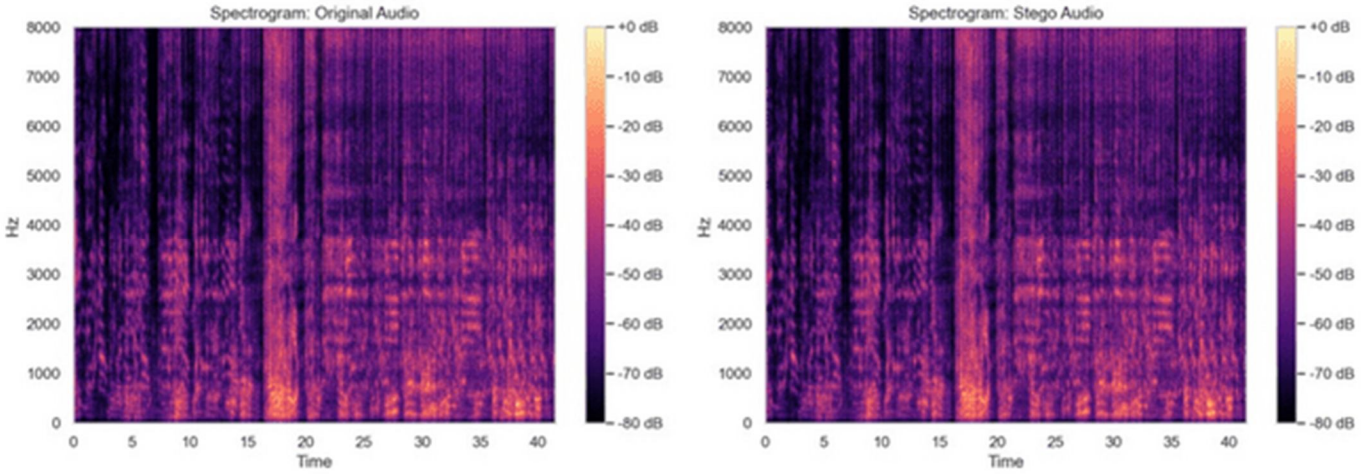


Fig. 7. Spectrogram Comparison of Original and Stego Audio

The spectrogram analysis in the Fig. 7. compares the frequency-domain behavior of the original and stego audio signals over time. Both spectrograms retained nearly identical spectral energy distributions across low-, mid-, and high-frequency regions. No abnormal frequency spikes or visible spectral discontinuities were observed after embedding as observed from the spectrogram. This suggests that the randomized payload insertion process preserved the acoustic structure of AudioCarrier.wav media file while successfully concealing the encrypted payload within the carrier samples.

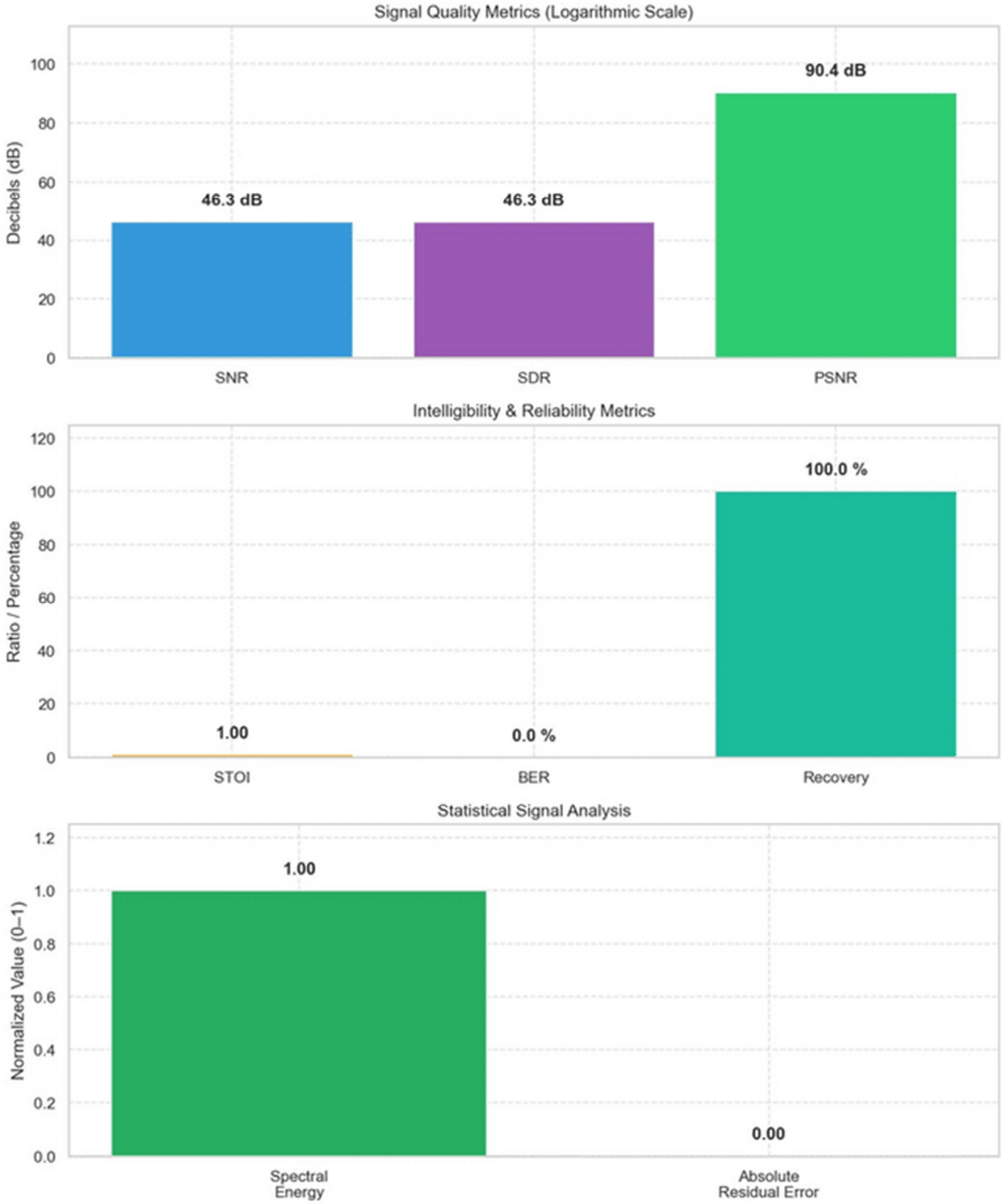


Fig. 8. Comprehensive Evaluation Graph for Audio Steganography

The comprehensive evaluation graph, above summarizes the overall performance of the proposed audio steganography framework using quality, intelligibility, reliability, and extraction-related metrics, therefore the framework achieved a PSNR value of 90.4 dB together with 46.3 dB SNR and SDR values, indicating strong signal preservation with extremely low embedding distortion. At the same time, the STOI value of 1.00, zero BER, no or zero Absolute Residual Error and complete payload recovery confirmed that the hidden information remained fully recoverable without corruption. Authentication verification through AES-256-GCM also succeeded during extraction, confirming payload integrity and protection against unauthorized modification.

D. PDF Steganography Results

The PDF steganography module was evaluated using the carrier document PDFcarrier.PDF with an original size of 3740.3 KB (3.72 MB). The embedded payload file was PDFpayload1.PDF, whose encrypted form required 15.86 KB after AES-256-GCM encryption. The generated stego output was stored as secure_output_PDFcarrier.PDF. The encrypted payload was embedded inside internal metadata fields and PDF object stream structures rather than visible page content. This approach preserved document appearance while allowing secure payload storage and authenticated recovery. Before embedding, a carrier capacity analysis was performed. The calculated safe embedding capacity of the selected PDF carrier was 64.00 KB, which remained well above the encrypted payload requirement of 15.86 KB; hence, this ensured that the embedding process did not introduce excessive structural modification or abnormal file growth, and the original and stego PDF documents were evaluated using file size comparison, entropy analysis, metadata inspection, and rendering verification. Both documents opened successfully in standard PDF viewers without formatting corruption or rendering inconsistencies. Table V shows us the PDF Steganography Embedding and Extraction Results

TABLE V. PDF Steganography Embedding and Extraction Results

S. No.	Parameter	Value	Good Range	Status
1	Carrier File	PDFcarrier.PDF	Standard PDF carrier	Valid
2	Stego File	secure_output_PDFcarrier.PDF	Successful output generation	Generated
3	Payload File	PDFpayload1.PDF	Supported payload format	Embedded
4	Original Size	3740.3 KB	Large carrier preferred	Good
5	Stego Size	3760.4 KB	Minimal increase preferred	Good
6	Absolute Size Increase	20.03 KB	Low overhead preferred	Excellent
7	Overhead Integration Ratio	0.54%	Minimal structural overhead	Excellent
8	Payload Requirement	15.86 KB	< Safe capacity	Safe
9	Maximum Safe Capacity	64.00 KB	> Payload requirement	Excellent
10	Embedding Location	Metadata/Object Streams	Non-visible structures preferred	Secure
11	Encryption	AES-256-GCM	AES-256 standard	Strong
12	Shannon Entropy (Original)	7.319	Stable entropy preferred	Good
13	Shannon Entropy (Stego)	7.327	Minimal entropy variation	Excellent
14	Entropy Deviation ((\Delta H))	0.0072	Near-zero variation preferred	Excellent
15	Total Pages Verified	22	Structural preservation required	Passed
16	Visual Rendering	Unchanged	No visible distortion expected	Excellent
17	Structural Integrity	Preserved	Valid PDF syntax required	Excellent
18	Extraction Status	Successful	Successful recovery required	Passed

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue VI June 2026- Available at www.ijraset.com

PDF steganography module was evaluated using PDF document carriers with encrypted payload embedding inside metadata fields and internal object structures of the carrier PDF file named PDFcarrier.PDF. The evaluation mainly focused on preserving document rendering quality, maintaining structural integrity, ensuring successful payload recovery, and verifying AES-256-GCM authentication after extraction; along with that, visual comparison and internal metadata analysis of the stego file were also performed to confirm secure and transparent payload embedding without affecting normal PDF functionality.

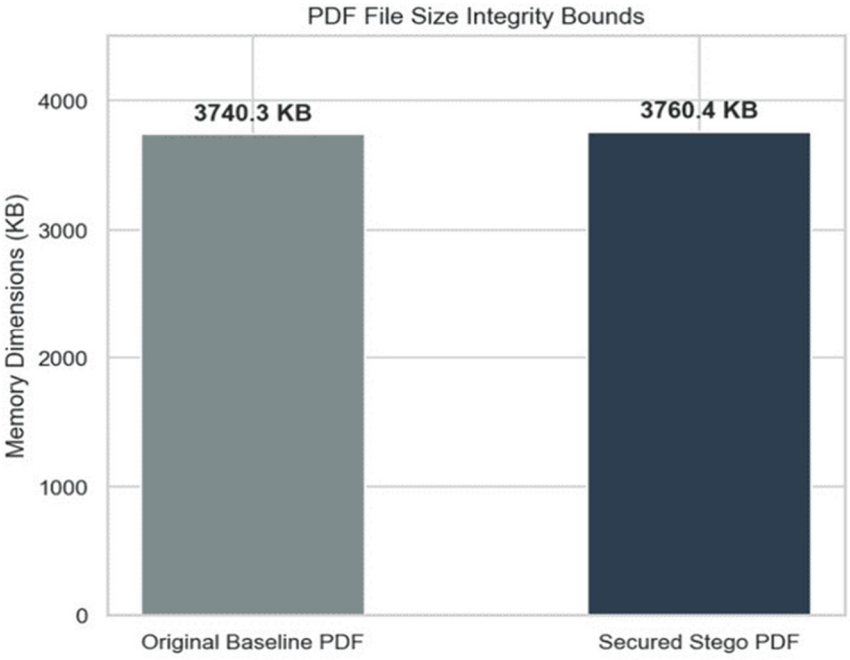


Fig. 9. File Size Integrity Comparison for PDF Embedding

The Fig. 9 shows the file size comparison between PDFcarrier.PDF and secure_output_PDFcarrier.PDF. After embedding the encrypted payload, the document size increased from 3740.3 KB to 3760.4 KB, resulting in a structural increase of only 20.03 KB in the PDF size. This small increase confirms that the embedding operation introduced only minimal overhead while preserving overall document integrity in total.

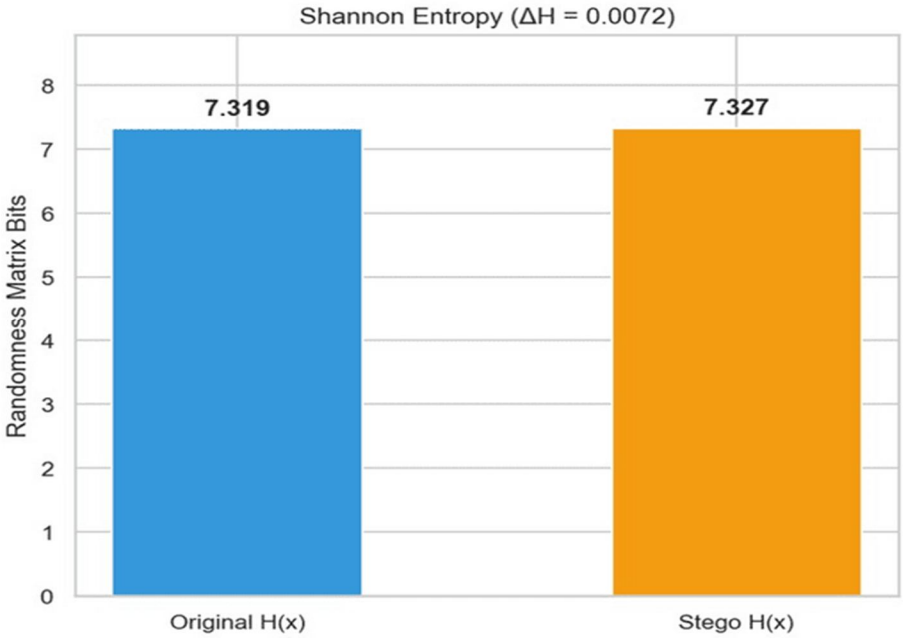


Fig. 10. Entropy Analysis of Original and Stego PDF Documents

The Fig. 10 presents the entropy comparison between the original and stego c PDF documents. The Shannon entropy changed slightly from 7.319 to 7.327 when the two files are compared meaning, producing an entropy deviation of only 0.0072. Such a small variation/difference indicates that the embedding process preserved the statistical characteristics of the original document without introducing any abnormal randomness patterns.

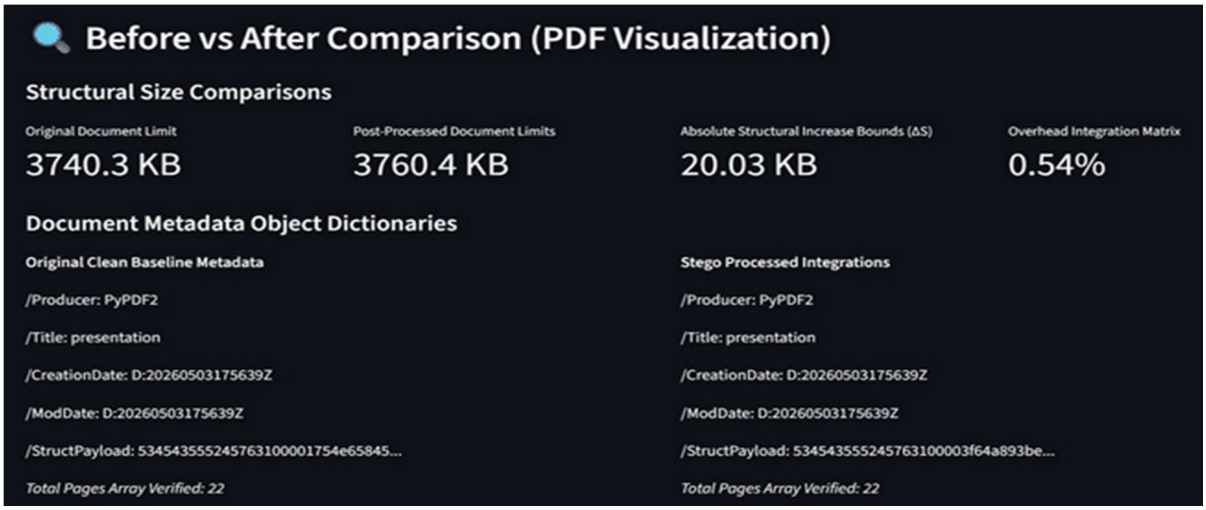


Fig. 11. Metadata and Object Stream Payload Embedding Analysis

The Fig. 11 shows us that the encrypted payload embedded within the metadata and object stream structures of stego PDF file named secure_output_PDFcarrier.PDF. The hidden ciphertext remained stored inside non-visible PDF components while preserving document rendering, structural consistency and integrity. The metadata structure remained valid after embedding, and the document retained all 22 pages without corruption or accessibility issues.

E. Video Steganography Results

The video steganography module was evaluated using MP4 video carriers with randomized 1-LSB embedding across selected video frames. The evaluation mainly focused on preserving playback quality of the file, maintaining structural frame similarity of the file, ensuring successful payload recovery, and verifying AES-256-GCM authentication after extraction. The original and stego videos were compared using PSNR, SSIM, MSE, SNR, and BER metrics to analyse embedding distortion and extraction reliability. Frame-level visual analysis and sequence-based statistical evaluation were also performed to verify embedding transparency and video integrity preservation. Table VI clearly summarizes the overall video steganographic embedding and extraction performance.

TABLE VI. Video Steganography Results

S. No.	Parameter	Value	Status
1	Carrier File	VideoCarrier.mp4	Valid
2	Carrier Size	4.3 MB	Good
3	Payload File	PDFpayload1.PDF	Embedded
4	Payload Size	15.86 KB	Safe
5	Safe Capacity	115480.58 KB	Excellent
6	Embedding Method	Randomized 1-LSB	Secure
7	Video Resolution	848 × 566	Preserved
8	Total Frames	657	Preserved
9	PSNR	89.79 dB	Excellent
10	SSIM	1.00	Excellent
11	MSE	0.00	Excellent
12	SNR	82.71 dB	Excellent
13	BER	0.0000	Ideal
14	Extraction Status	Successful	Passed

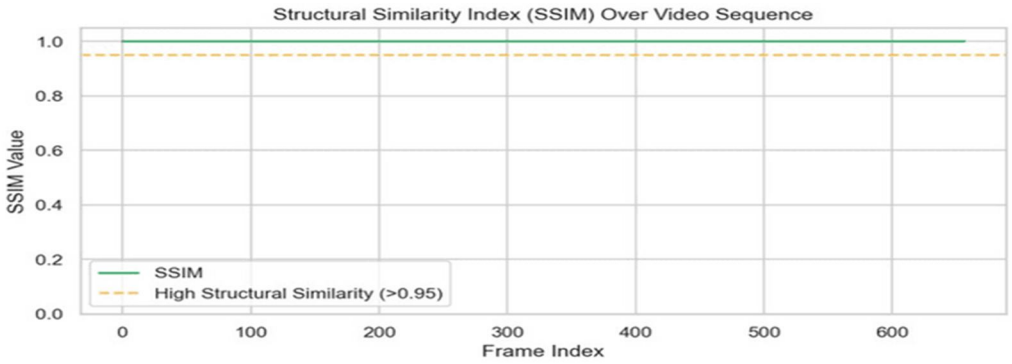


Fig. 12. Structural Similarity Index (SSIM) Across Video Frame Sequence

The graph in the Fig. 12 shows us the SSIM, i.e. Structural Similarity Index Measure variation across the complete embedded video frame sequence. The SSIM values remained consistently close to 1.0 and above the high structural similarity threshold of 0.95, confirming that the stego video preserved the structural and visual characteristics of the original video frames with negligible or hardly any amount of perceptual distortion after embedding.

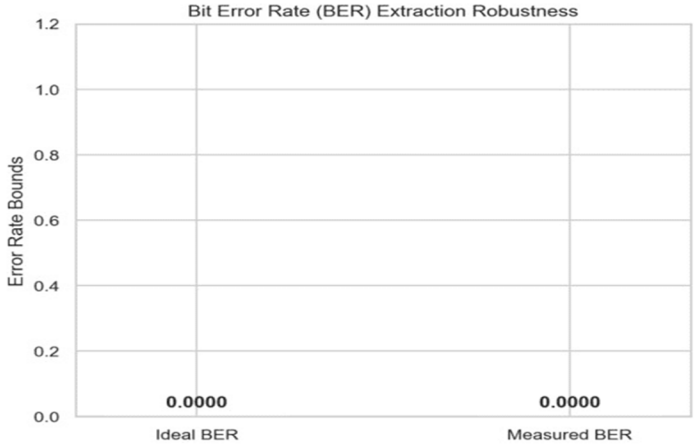


Fig. 13. Bit Error Rate (BER) Extraction Robustness

The graph in the Fig. 13. presents the Bit Error Rate (BER) evaluation for payload extraction robustness. The obtained 0.0000 BER exactly matched the ideal BER condition which is 0.0, indicating completely error-free payload recovery after extraction therefore this totally confirms successful ciphertext reconstruction and reliable AES-256-GCM authentication verification without any data corruption hence good recovery.

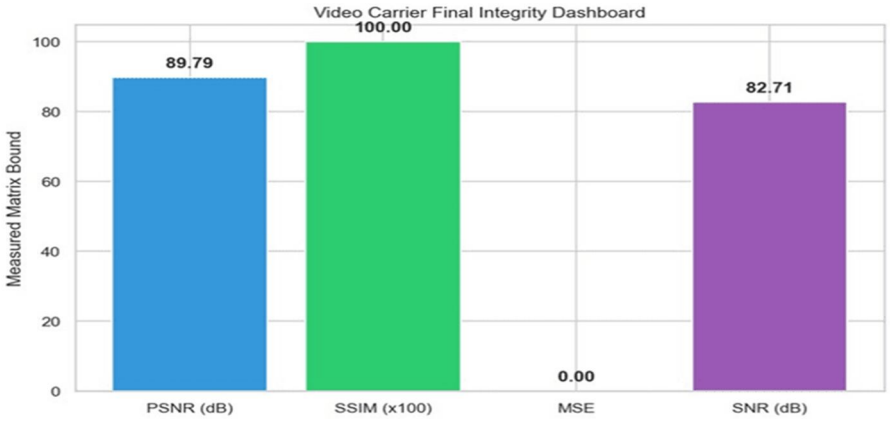


Fig. 14. Video Carrier Final Integrity Dashboard

The integrity dashboard in the Fig.14 has graphs which summarize the overall performance metrics obtained during video steganographic evaluation of the given carrier and stego files. The proposed framework has achieved a high 89.79 dB PSNR, 1.00 SSIM, 82.71 dB SNR, and near-zero MSE, indicating extremely low embedding distortion and excellent visual transparency all along. The obtained results confirm that the randomized 1-LSB embedding process preserved playback quality, frame integrity, and structural similarity while securely concealing the encrypted payload safely.

VIII. CONCLUSION AND FUTURE SCOPE

A. Conclusion

This report presents a structured multi-modal steganography framework for secure data hiding across diverse digital carriers. The design combines authenticated encryption, randomized embedding, carrier-aware capacity checking, and flexible support for images, audio, PDFs, and videos. The resulting system does more than conceal content; it conceals communication itself while still preserving recovery integrity. The major strength of the framework is its layered approach. AES-256-GCM protects the payload even if it is found. Randomized LSB insertion reduces predictable patterns. Carrier selection prevents misuse. Together these features create a practical covert communication model that is suitable for real-world secure data transfer scenarios where simple encryption is not enough. The contribution of the work is therefore both technical and practical. Technically, it combines encryption and steganography in a clean pipeline. Practically, it allows the same security logic to be reused across multiple file formats, which increases deployment flexibility.

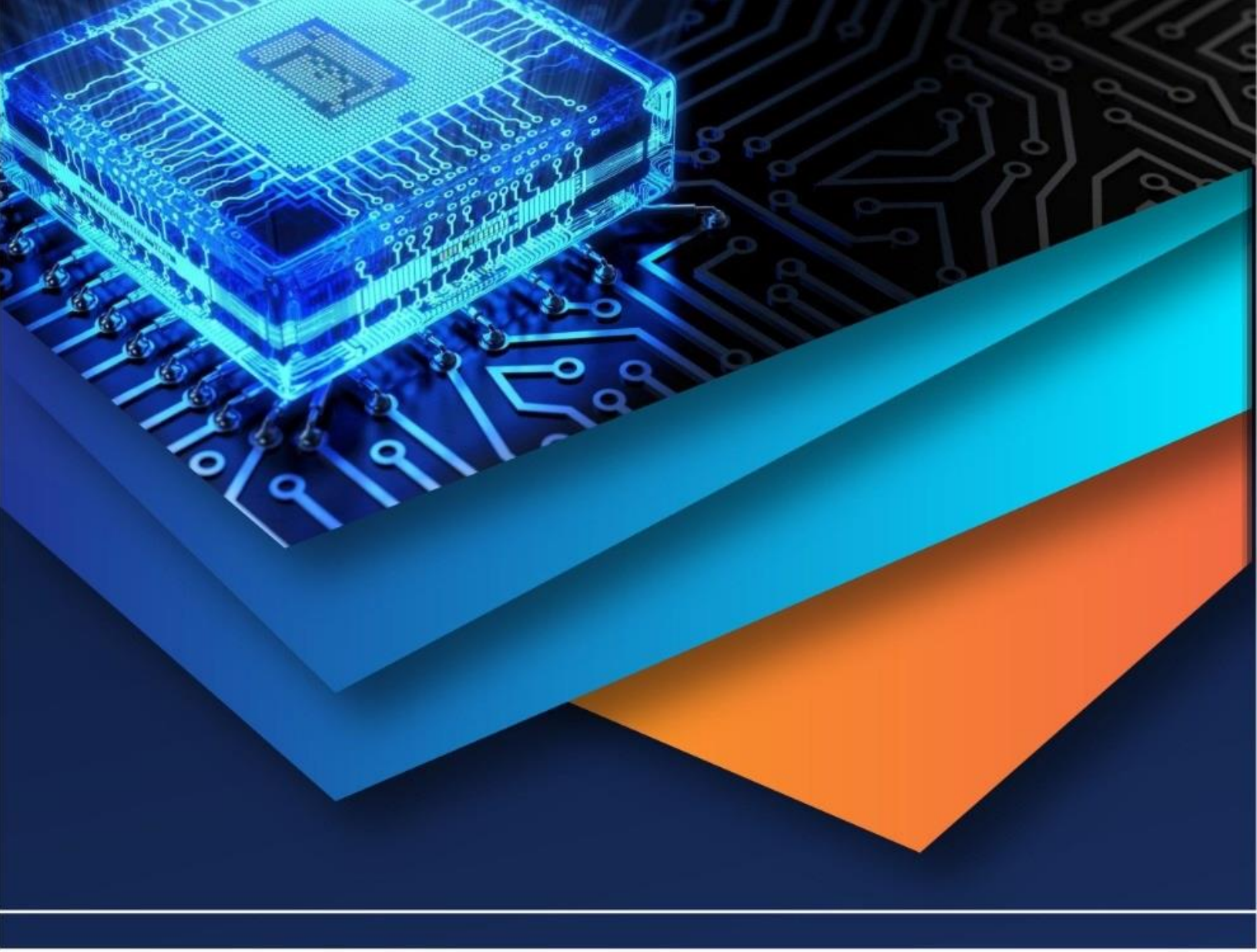
B. Future Scope

Several extensions are possible. A major one would be real-time steganography for live communication tools such as VoIP or conferencing systems. Blockchain-backed integrity logging could support tamper-evident audit trails. Machine-learning-based anti-steganalysis resistance could also be studied, provided that the learning system is used for defensive detection analysis rather than for concealment of malicious activity. Another promising direction is cloud-native deployment. A secure web application could allow users to upload a carrier, embed a payload, and retrieve the stego output through a browser interface while preserving local key secrecy. That would make the system easier to test at scale and easier to integrate into enterprise workflows. Finally, a stronger experimental section can be built by comparing this framework against a baseline sequential LSB system. Such a comparison would show the practical benefit of randomization and authenticated encryption more clearly than absolute performance numbers alone.

REFERENCES

[1] I. Haverkamp and D. K. Sarmah, “Evaluating the merits and constraints of cryptography–steganography fusion: A systematic analysis,” Int. J. Inf. Secur., vol. 23, pp. 2607–2635, May 2024.
[2] R. Klemm and B. Chen, “Hiding sensitive information using PDF steganography,” arXiv preprint arXiv:2405.00865, May 2024.
[3] S. Rahman, J. Uddin, H. Hussain, S. Shah, A. Salam, F. Amin, I. de la Torre Díez, D. L. Ramírez Vargas, and J. C. Martínez Espinosa, “A novel and efficient digital image steganography technique using least significant bit substitution,” Sci. Rep., vol. 15, Art. no. 107, Jan. 2025.
[4] T. Radivilova, I. Dobrynin, A. Snihurov, S. Stanhei, and S. Bulba, “Image steganography method using LSB and AES encryption algorithm,” in Proc. CQPC 2025, CEUR Workshop Proceedings, Aug. 2025.
[5] D. T. Firdaus, N. J. D. C. Ntivuguruzwa, T. Ahmad, D. Mukanyiligira, and L. Sibomana, “Steganographic model to conceal the secret data in audio files utilizing a fourfold paradigm: Interpolation, multi-layering, optimized sample space, and smoothing,” J. Saf. Sci. Resil., vol. 6, no. 2, pp. 138–149, Jun. 2025.
[6] D. Kumar, V. K. Sudha, N. Manikandan, and K. Ramaswamy, “Efficient three-layer secured adaptive video steganography method using chaotic dynamic systems,” Sci. Rep., vol. 14, Art. no. 18301, Aug. 2024.
[7] M. ud Din, I. A. Shoukat, E. Arif, M. Amjad, M. Mohsin, M. Mumtaz, and M. A. Rauf, “Randomized frame selection-based video steganography method for secure embedding of secret data,” J. Comput. Biomed. Inform., vol. 8, no. 2, 2025.
[8] O. Omego and M. Bosy, “Multichannel steganography: A provably secure hybrid steganographic model for secure communication,” arXiv preprint arXiv:2501.04511, Jul. 2025.
[9] H. El-Taj, “A secure fusion: Elliptic curve encryption integrated with LSB steganography for hidden communication,” Int. J. Comput. Exp. Sci. Eng., vol. 10, no. 3, pp. 434–460, 2024, doi: 10.22399/ijcesen.382.
[10] M. A. Nasr, W. El-Shafai, E.-S. M. El-Rabaie, A. S. El-Fishawy, H. M. El-Hoseny, F. E. Abd El-Samie, and N. Abdel-Salam, “A robust audio steganography technique based on image encryption using different chaotic maps,” Sci. Rep., vol. 14, Art. no. 22054, 2024, doi: 10.1038/s41598-024-70940-3.
[11] S. Z. Banoori, W. Khan, S. Rahman, F. Masood, A. Salam, F. Amin, I. de la Torre, M. G. Villar, H. Garay, and G. S. Choi, “An improved hybrid image steganography method using AES algorithm,” Sci. Rep., vol. 15, Art. no. 44515, 2025, doi: 10.1038/s41598-025-28140-0.
[12] J. Singh, A. Kumar Singh, and S. S. Chauhan, “Enhanced edge-based steganography using image segmentation and random LSB substitution for secure data hiding,” Int. J. Smart Sens. Intell. Syst., vol. 18, no. 1, 2025, doi: 10.2478/ijssis-2025-0049.
[13] M. Helmy and H. Torkey, “Secured audio framework based on chaotic-steganography algorithm for Internet of Things systems,” Computers, vol. 14, no. 6, Art. no. 207, 2025, doi: 10.3390/computers14060207.

[14] A. Aravind Krishnan, Y. Ramesh, U. Urs, and M. Arakeri, "Audio-in-image steganography using analysis and resynthesis sound spectrograph," IEEE Access, pp. 75180–75189, 2025, doi: 10.1109/ACCESS.2025.3563781.
[15] S. N. Al-Rekaby, M. A. Khodher, and L. K. Adday, "Video steganography using a 9-D chaotic system and an encryption algorithm for data hiding," Int. J. Saf. Sec. Eng., pp. 1007–1018, 2025, doi: 10.18280/ijsse.150514.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY