Journal of Computer Science

Research Article

Hybrid Data Security Approach Based on GAN Teganographyand AES-RSA Encryption in a Distributed System

Derrick Méthode Bagaza[1], Abraham Walake[2], Mamoudou Kourouma[1] and Blaise Omer Yenke[3]

[1]*Department of Computer Science, Higher Institute of Technology, University of Bangui, Central African Republic*
[2]*Department of Mathematics and Computer Science, Faculty of Science, The University of Ngaoundere, Cameroon*
[3]*Laboratory of Applied Computer Science and Industrial Systems Intelligence, The University of Ngaoundere, Cameroon*

Article history
Received: 22-01-2026
Revised: 17-04-2026
Accepted: 12-06-2026

Corresponding Author:
Derrick Méthode Bagaza
Department of Computer Science, Higher Institute of Technology, University of Bangui, Central African Republic
Email: derrickmethodebagaza@gmail.com

Abstract: Data security in distributed environments has become a major challenge due to the dispersion of resources, the expansion of the attack surface, and the increasing sophistication of cyber threats. Although traditional mechanisms such as data encryption, multi-factor authentication, and access control systems help to strengthen information protection, these approaches still have certain limitations when faced with advanced attacks and human error. To address these challenges, this article proposes a hybrid security architecture that integrates cryptography and steganography techniques in a distributed environment. The proposed model combines AES-RSA encryption with steganography based on Generative Adversarial Networks (GANs), as well as a distributed data transmission mechanism. In this approach, the message is first encrypted using a hybrid AES-RSA scheme and then hidden in an image using a GAN-based steganography model. The steganographic image is then fragmented and transmitted through different nodes in the distributed system, which enhances the confidentiality, integrity, and resilience of the data against potential attacks. Experiments conducted on a set of images with different resolutions show that the proposed architecture offers superior performance compared to traditional methods such as LSB-DCT. Performance evaluation, based on PSNR (peak signal-to-noise ratio) and SSIM (Structural Similarity Index Measurement), demonstrates a significant improvement in visual quality, robustness to degradation, and processing efficiency. The results confirm that integrating GANs into a hybrid framework combining cryptography and distributed steganography is a promising approach to improving data security in distributed environments.

Keywords: Steganography, Cryptography, GAN, AES-RSA, PSNR, Data Security, Distributed Systems

Introduction

The swift expansion of digital data sharing across diverse distributed networks raises significant concerns regarding security, privacy, and the protection of sensitive information. Although cryptography can protect data through encryption and steganography can hide the presence of messages, conventional methods such as encryption, physical access control, and robust authentication have limitations. For example, depending solely on cryptography might make encrypted data more conspicuous, while traditional steganographic techniques, like LSB, can be susceptible to steganalysis. Contemporary hybrid solutions frequently demonstrate a lack of flexibility and face challenges in adapting to the evolving nature of dynamic distributed environments (Bagaza et al., 2023). The authors have developed and assessed a hybrid steganography approach that combines LSB and DCT techniques to enhance the security, resilience, and capacity to conceal sensitive information within images (Rajguru et al., 2024). In the paper titled "Stenographic Approaches for Enhancing Data Security in Cloud Computing," the authors investigate the application of steganographic methods to boost data

security in cloud environments. The article reviews various steganographic techniques, including LSB, DCT, and hybrid methods, which are implemented in images and multimedia files, often paired with cryptographic measures for extra protection. Furthermore, the authors discuss challenges related to latency, resistance against steganalysis attacks, and performance enhancement in a distributed setting.

In recent years, advances in artificial intelligence and deep learning have opened up new possibilities for improving steganography techniques. In particular, Generative Adversarial Networks (GANs) have demonstrated significant potential for generating realistic images and concealing information in a more robust and less detectable manner. The integration of GANs into steganography systems improves the visual quality of steganographic images while enhancing resistance to detection techniques.

In this context, several recent studies have proposed hybrid approaches combining cryptography and steganography to enhance data security in distributed environments. However, many existing models still have certain limitations, particularly in terms of robustness, transmission efficiency, and adaptability to modern distributed architectures.

They determine that hybrid techniques (merging cryptography and steganography) are especially powerful for cloud settings, as they improve privacy and lessen the chances of data interception or alteration. This study introduces a smart hybrid strategy that integrates AES-RSA encryption to guarantee confidentiality, SteganoGAN-based steganography for subtle and sturdy concealment, and a distributed framework to boost the system's durability.

Furthermore, recent studies have demonstrated the effectiveness of artificial intelligence–based approaches in strengthening the security of distributed systems. For instance, highlighted the use of hybrid deep learning models to improve anomaly detection in IoT ecosystems. In this approach, Generative Adversarial Networks (GANs) are employed to generate synthetic samples in order to balance datasets and enhance the generalization capability of security models. These results indicate that GANs can play an important role in handling imbalanced data and detecting complex cyber threats. The integration of such AI-based approaches into hybrid security architectures further strengthens the robustness and adaptability of systems against modern cyberattacks.

Related Work

Steganography based on LSB techniques, as discussed by Chi-Kwong and Cheng (2004), along with DCT methods explored by Bender et al. (1996), have been extensively utilized but are still vulnerable to compression and risks of statistical detection. Zhang et al. (2019) presented a novel approach to DNA-based cryptography that allows for the development of nanoscale molecular formations capable of encoding and preserving data. They are exploiting the programmability and complexity of folded DNA structures to develop secure communication systems at the molecular level. The process involves creating DNA origami that carries the data, transmitting it, and deciphering the information using special probes. This method provides a level of physical security that is difficult to break and paves the way for hardware encryption and data protection applications at the nanoscale. Zhang et al. (2019) published a detailed analysis of various techniques used for digital image steganography. These cover both traditional techniques such as LSB, DCT, and DWT, as well as advanced approaches that incorporate artificial intelligence such as GANs and deep neural networks. The authors consider evaluation criteria such as insert ability, imperceptibility, and resistance to attacks, and focus on current challenges such as resistance to compression, noise, and modern steganalysis techniques. They propose a structured classification of existing techniques, review recent trends, identify future prospects, and highlight the growing role of machine learning techniques in improving the security and confidentiality of hidden data.

Nechvatal et al. (1999) provided an initial evaluation of the AES standard candidate after the first round. We discuss the selection criteria used, including security, performance, flexibility, and ease of implementation. This study provides a comparative analysis of algorithms published by various researchers and industry experts and evaluates their performance on different software and hardware platforms. It also investigates possible vulnerabilities and assesses resistance to known attacks. Finally, the report identifies the algorithms selected in the second stage of the selection process that form the basis of the final AES selection.

AlKhamese et al. (2019) provided an overview of steganography techniques used in cloud computing to improve data security. The authors point out that in cloud environments, data is particularly vulnerable to interception and tampering while in transit and at rest. They review various steganography techniques (LSB, DCT, DWT, hybrid methods using encryption) and assess their effectiveness in protecting data stored in the cloud. The study concludes that adding a steganography layer on top of encryption can significantly improve privacy and attack resistance. However, this includes capacity, detection, and productivity. Kaur and Rani (2016) provide a detailed overview of various steganography techniques and discuss their principles, advantages, disadvantages, and applications. The authors classify methods based on media type (image, audio, video, text) and integration method (LSB, frequency transformation such as DCT/DWT, adaptation method, etc.). It also mentions performance criteria such as capacity, stealth, and attack

resistance. The article concludes with a discussion of current issues and future directions, including integration with artificial intelligence to improve the security and anonymity of hidden data. In this paper, the authors introduce an improved mechanism to protect embedded text in videos using steganography techniques. This approach combines encryption and steganography to protect data from interception and detection. Text data is first encrypted and then hidden within video frames using an optimized Least Significant Bit (LSB) replacement method. The authors demonstrate that their approach provides greater resistance to detection attacks while retaining the visual quality of the video as measured by PSNR. Izevbizua (2015) presented a data security model for the cloud that combines two strategies: Serpent Cipher, a block cipher algorithm known for its strength and speed used to protect data content, and distributed steganography, in which encrypted data is embedded in multiple image files scattered across different nodes in the cloud. This method prevents an attacker with access to a single image from recovering all information. Adee and Mouratidis (2022) demonstrate that this combination improves privacy and attack resistance while reducing risks associated with single points of failure. The authors propose a dynamic four-stage data protection model in cloud computing that integrates encryption and steganography to ensure confidentiality, integrity, and availability. This model includes

1) Initial data encryption
2) Steganographic encapsulation into a covering medium
3) Secure transmission via the cloud
4) Verification and extraction at the receiving end

Experimental results show that this model is effective against sniffing attacks and reduces the risk of compromise even if the data stream is intercepted. The authors note that this combination strengthens resilience against modern persistent threats. The GAN presented by Goodfellow et al. (2014) made it possible to generate images indistinguishable from real images, paving the way for more covert steganography. At the same time, the AES-RSA Hybrid Encryption by Stallings (2006) combines the speed and security of key exchange.

Recent studies from 2019 to 2025 have explored hybrid and AI-driven methods that significantly improve data hiding and security. Huo et al. (2024) proposed a high-capacity image steganography network based on chaotic mapping and GANs, improving both hiding capacity and security of images. Li et al. (2024) implemented GAN-based image steganography with style transformation, generating steganographic images that are difficult to detect and robust against steganalysis. Malik et al. (2025) demonstrated a hybrid GAN + DCT framework for secure data communication in big data environments, enhancing image quality and data transmission security. Shwaysh et al. (2024) combined AES encryption with secret sharing and steganography, improving both the confidentiality and resilience of transmitted images. Patil and Sonaje (2024) proposed AES + LSB hybrid encryption to secure hidden text in images, offering an effective layer of protection for message transmission. Tang et al. (2025) introduced reversible generative steganography with distribution-preserving, maintaining statistical consistency of images while enabling robust and stealthy information hiding.

The recent references demonstrate a clear trend toward hybrid AI-powered steganography, combining cryptography (AES-RSA) and GAN-based approaches to improve security, stealth, robustness, and flexibility in distributed systems. This highlights the fact that integrating AI-driven generative models into hybrid security frameworks represents a promising approach to protecting sensitive data in modern distributed environments.

Materials and Methods

This section presents a new approach to data security based on a three-block architecture combining encryption, steganography based on Generative Adversarial Networks (GANs), and distributed transmission. This integration simultaneously protects the confidentiality, invisibility, and resilience of the system.

Materials

The material used by researchers to experimentally validate the proposed model is a specific dataset covering all stages of the process: Encryption, steganography, and distributed transmission.

Technical Environment and Implementation Tools

The implementation of the proposed triple-block hybrid security model was carried out in a controlled technical environment designed to meet the requirements for computation, simulation, and experimental validation. This environment integrates both local and cloud resources to ensure flexibility, reproducibility, and efficiency throughout the experimental process.

Local experiments were conducted on a system equipped with an Intel Core i7-4700MQ processor (2.40 GHz, 4 physical cores, 8 threads) and 8 GB of DDR4 RAM, providing sufficient processing power for cryptographic operations and image processing tasks. The graphics calculations related to GAN training and inference were handled by an integrated Intel graphics unit for preliminary testing, while the intensive deep learning computational loads were offloaded to Google Colab, leveraging cloud-based GPU acceleration (Tesla T4) to reduce training time and improve scalability.

The software environment is primarily based on Python 3.11, chosen for its rich ecosystem in cryptography and deep learning. The GAN model was implemented using PyTorch, facilitating flexible architecture design, efficient gradient computation, and model optimization. Cryptographic operations were performed using the cryptography and PyCryptodome libraries, ensuring secure implementations that comply with AES and RSA algorithm standards. Image processing and visualization were handled by Pillow (PIL), NumPy, and Matplotlib, while Jupyter Notebook was used to organize experiments and document results.

Data Sets and Data Preparation

The experimental validation of this model was carried out using a test covering all stages of the security chain, including encryption, steganographic insertion, and distributed transmission. The data used in the experiments consisted of two main elements: The messages to be concealed and the cover images.

The messages correspond to UTF-8-encoded text data, simulating sensitive information such as confidential messages or identifiers. Before steganographic insertion, these messages were converted to binary format and then encrypted using the AES-RSA hybrid encryption framework.

The cover images were selected in PNG format to avoid any loss of information due to lossless compression. Image resolutions vary between 256×256 and 512×512 pixels, all represented in the RGB color space (three channels). The dataset includes both public, royalty-free "Simulation-Result-on-Lena" images and personal photographs taken with a smartphone, allowing the model to be evaluated on a variety of visual content.

Before being used in the GAN, all images underwent a standardized preprocessing phase that included conversion to RGB mode, resizing to a fixed resolution of 256×256 pixels, and normalization of pixel values to the range [−1, 1], which is suitable for deep learning-based image generation tasks.

Hybrid AES-RSA Encryption Framework

The first block of the proposed architecture is based on a hybrid cryptographic framework combining AES and RSA, selected for its balance between security level and computational efficiency. The AES algorithm is used in CFB (Cipher Feedback) mode with a 256-bit secret key, ensuring high confidentiality of encrypted messages. A randomly generated 16-byte Initialization Vector (IV) is used to guarantee semantic security.

To ensure secure key management in a distributed environment, the AES secret key is encrypted using RSA with a key size of 2048 bits. The OAEP padding scheme combined with SHA-256 is applied to enhance resistance to cryptographic attacks. The encrypted message and the encrypted AES key together form the payload, which is then embedded into the cover image via the GAN-based steganography module.

GAN-Based Steganography Configuration

The steganographic insertion process is implemented using a Generative Adversarial Network (GAN) composed of a generator and a discriminator. The generator architecture comprises two convolutional layers followed by ReLU and Tanh activation functions, enabling it to insert the encrypted payload while preserving the visual fidelity of the image. The discriminator consists of two convolutional layers with LeakyReLU and Sigmoid activations, enabling it to distinguish between original images and steganographic images.

The model is trained using the Binary Cross-Entropy (BCE) loss function, with optimization performed by the Adam optimizer ($\beta1 = 0.5$, $\beta2 = 0.999$) and a learning rate of 2×10^{-4}. Training is performed over 10 epochs, with images normalized in the interval [−1, 1]. This configuration ensures a balanced compromise between insertion capability, imperceptibility, and extraction reliability.

Simulated Distributed Transmission Environment

In order to evaluate the robustness of the proposed system under realistic conditions, a simulated distributed transmission environment was implemented. Each steganographed image is divided into $n = 4$ horizontal fragments, saved in lossless PNG format, and then transmitted as independent segments to simulate multi-node routing in a distributed system.

The transmission process is conceptually modeled using a secure communication protocol based on TLS, ensuring the confidentiality of exchanges. A sequential node assignment strategy, with reassignment in case of unavailability, is adopted to simulate fault tolerance. On the receiving end, the original steganographic image is reconstructed by sequentially concatenating all the fragments received, after which the message is extracted and decrypted.

Block 1: Hybrid Encryption AES-RSA

The purpose of the first block is to protect the message content using hybrid encryption. The Advanced Encryption Standard (AES) algorithm is used to quickly and efficiently encrypt data using symmetric encryption. The generated AES keys are encrypted with RSA (Rivest, Shamir, Adleman) to ensure secure key distribution in an asymmetric context. This method combines the effectiveness of AES against key sniffing attacks with the power of RSA.

Block 2: GAN-Based Steganography

The second block aims to make the encrypted message undetectable by embedding it in the cover image. Encode

the data using a GAN generator so that the resulting image maintains a realistic appearance. On the other hand, the discriminator is trained to distinguish between the original image and the modified image. This adversarial learning process improves resilience against traditional steganalysis techniques and resists degradations such as JPEG compression or added noise.

Block 3: Secure Distributed Transmission

The third block is dedicated to the transfer of steganographic images within distributed environments (cloud, peer-to-peer networks, micro services). The resulting image is split into multiple segments that are sent via different nodes, thus minimizing the risk associated with complete data interception. RSA keys are also handled in a distributed manner to avoid single points of failure. This mechanism makes the system more resilient and more resistant to targeted attacks.

In Block 3, distributed transmission is used to enhance the system's security and resilience. Following the GAN-based steganography process, the steganographic image containing the encrypted message is fragmented into several segments before being transmitted via different nodes in the distributed network.

Data Fragmentation

The steganographic image is first divided into N independent segments according to a partitioning strategy (e.g., by blocks or data fragments). Each segment is assigned a unique identifier (Segment ID) along with metadata comprising:

- The segment's order in the original image
- A session identifier
- An integrity check code

These segments are then sent to different nodes in the distributed system or the cloud, thereby reducing the likelihood that an attacker could intercept the complete image.

Transmission Via Multiple Nodes

Each segment is transmitted via a different network path or through separate nodes within the distributed architecture. This strategy offers two main advantages:

- Reduced risk of complete data interception, as an attacker would need to compromise several nodes simultaneously
- Improved fault tolerance, as the loss of a single node does not necessarily result in the loss of all data

Reassembling Segments

At the receiver end, a Reconstruction Manager is responsible for collecting and assembling the received segments.

The reassembly process involves several steps:

a. Receiving segments from the various nodes
b. Verifying the integrity of each segment using its associated hash
c. Sorting the segments according to their Segment ID
d. Reconstructing the steganographic image by reassembling the segments in their original order

Once the image has been reconstructed, the system performs:

- Steganographic extraction to retrieve the hidden message
- Followed by AES-RSA decryption to recover the original data

Node Failure Management

In a real-world cloud environment, some nodes may become unavailable due to network outages, server overload, or malicious attacks. To ensure system reliability, several mechanisms are built in:

- Segment redundancy: Certain segments may be duplicated and sent to multiple nodes
- Reception timeout: If a segment is not received within a specified time, a new transmission request is triggered
- Alternative routing: Segments can be redirected to other available nodes in the network
- Transmission logging: A central log tracks the status of each segment sent

These mechanisms enable the system to maintain a high level of availability and fault tolerance, even in large-scale distributed environments.

Summary of the Processing Flow

Therefore, the message is first encrypted with AES-RSA (block 1) and then integrated into the image using GAN (block 2). The steganographic image is then fragmented and transmitted over a secure distributed network (block 3). This processing chain provides multi-layered security through a combination of cryptography, steganography, and distributed architecture.

Figure 1 provides a comprehensive overview of the proposed data security framework. This is achieved through a mixed approach that combines AES-RSA encryption, GAN-based steganography, and cloud-based distributed configuration. This plan has two important parts: An item encryption and insertion phase, and an item retrieval and decryption phase.

The system starts working when a user first passes an authentication step, proving that only authorized people

get into the features that protect and retrieve data. This part is the first security step at the application layer.

In the encryption phase, the source file is first subjected to a hybrid encryption process, combining symmetric encryption to ensure content security and asymmetric encryption to protect the key. The encrypted message is then sent to the GAN-based data integration module, where it is optimally concealed within a multimedia medium (the cover image). This step generates a steganographic image, which discreetly contains the encrypted data.

The steganographic image thus produced is then transmitted and stored securely in a cloud infrastructure, designed as a distributed environment to ensure data availability, fault tolerance, and accessibility.

The retrieval phase begins when an authenticated user wishes to access the data. The steganographic image is extracted from cloud storage and then processed by the data extraction module using a GAN, allowing the encrypted message hidden in the medium to be retrieved.

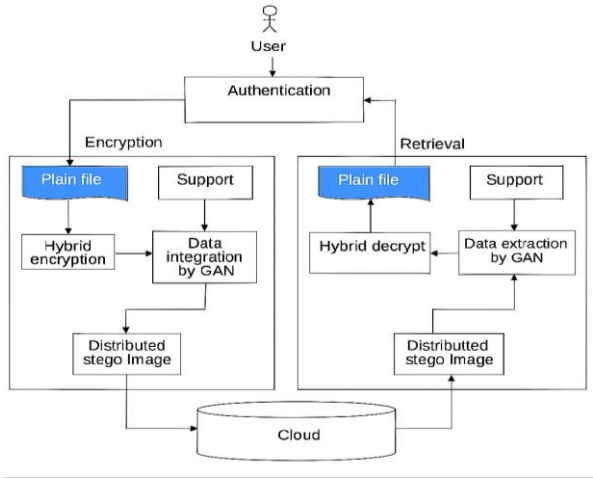

Fig. 1: The architecture has three components: encryption, GAN steganography, and secure distribution

This message is then subjected to a hybrid decryption process, which uses the appropriate keys to restore the original file to its unencrypted state.

This architectural concept implements a multi-layered security strategy, integrating authentication, encryption, artificial intelligence-based concealment, and cloud distribution mechanisms. This approach enhances data confidentiality, robustness, and resilience within a distributed environment.

Encryption and Decryption Process

This section details the encryption and decryption processes of the proposed model. First, the algorithms used within the different blocks of the model are presented.

Block 1: AES-RSA Encryption

The first step in our model aims to ensure data confidentiality and security before it is integrated into the carrier image. To this end, we use two complementary cryptographic algorithms:

- CFB (Cipher Feedback) mode with the AES-256 algorithm is used for message encryption, offering speed and efficiency in the context of symmetric encryption
- RSA-2048 is used to encrypt the AES key, thus ensuring secure key exchange through asymmetric encryption

The message is first converted into bytes and encrypted using a randomly generated AES key. This AES key is then itself encrypted using the recipient's RSA public key. The encrypted data and the encrypted AES key are then sent on to the next stage (Block 2: Steganography).

Algorithm 1: Hybrid Encryption AESRSA

Input: *M*: Message to encrypt, *Kpub*: RSA public key
Output: *C*: Message encrypted with AES,
KencAES: RSA-encrypted AES key

1. Generate a random AES-256 key: *KAES*;
2. Create a 16-byte initialization vector (*IV*);
3. Encrypt *M* using AES-256 in CFB mode with *KAES*:
4. *C* ← *AESCFB(M, KAES, IV);*
5. Encrypt *KAES* using the RSA public key:
6. *KencAES* ← *RSAenc(KAES, Kpub);*
7. Return *(C, KencAES, IV) ;*

Algorithm 1 presents a combined encryption method. It relies on symmetric and asymmetric cryptography to ensure enhanced information security. First, the initial message M is encrypted using the AES-256 algorithm in CFB mode. This approach speeds up the encryption process and remains effective, even for large volumes of data. The AES key is generated randomly, making it impossible to predict and significantly reducing the risk of brute force attacks. A 16-byte initialization vector helps to reinforce the randomness of the encryption and prevents pattern repetition within the encrypted data. To secure the exchange of the symmetric key, it is itself encrypted using an RSA public key. This method combines the speed of symmetric encryption with the security offered by asymmetric cryptography for key management. This hybrid architecture is particularly well suited to distributed systems, where data confidentiality and key protection are critical.

Block 2: GAN Steganography

The second block of the model is dedicated to secure data hiding using steganography techniques assisted by

Generative Adversarial Networks (GANs). In this phase, the Generator (G) incorporates two distinct components as input:

- The cover image: A seemingly ordinary visual representation designed to conceal information
- The encrypted message comes from Block 1, which uses a combination of AES and RSA algorithms

The generator merges these two inputs to produce a steganographic image that remains visually identical to the original image. The goal is for the modifications to be subtle enough to escape conventional steganography detection methods. At the same time, the Discriminator (D) is trained to differentiate between authentic images and steganographic images. This confrontation between G and D allows the generator to optimize its outputs in order to maximize:

- Stealth: The concealment of hidden data
- Robustness: The ability to withstand degrading treatments (compression, noise, blur)
- Adaptability: Optimized integration based on the visual characteristics of the image

Thus, this block plays a crucial role in:

1. Invisibility: Ensuring that the hidden data does not noticeably alter the image
2. Robustness: Ensuring reliable data recovery even after modifications to the image
3. Adaptability: Automatically adjusting the integration method

Principle of GAN Applied to Steganography

A GAN consists of two neural networks:

- The Generator G learns to encode data within a cover image while maintaining a visually realistic appearance
- The Discriminator D learns to differentiate between natural images and modified (stego) images

Steganography through GAN relies on adversarial learning where:

$$\min_{G}\max_{D} E_{x\sim P_{data}}[\log D(x)] + E_{z\sim P_z}[\log(1-D(G(x,z)))]$$

Where:

- x: cover image
- z: encrypted message (using AES)
- $G(x; z)$: steganographic image created from x and z
- $D(y)$: likelihood that y is a natural image

Algorithm 2: Training the GAN for steganography

Input: X: A lot of cover images, Z: A collection of encrypted messages
Output: G: Trained generator, D: Trained discriminator
For *epoch* ← 1 to N do

For each mini-batch $(x, z) \in (X, Z)$ do

Produce $y = G(x, z)$;

Calculate $D(x)$ and $D(y)$; update D by maximizing:

$\log D(x) + \log (1 - D(y))$

Update G by minimizing:

$\log (1 - D(G(x, z))) + \lambda \cdot MSE(x, G(x, z))$

Where λ is a weighting factor for visual distortion.

Algorithm 2 shows precisely how a Generative Adversarial Network (GAN) used for steganography gets trained. The setup relies on how two neural networks work together: A generator and a discriminator. A normal picture and a secret message go to the generator, which makes a hidden picture looking much like the first. But the discriminator tries to tell apart actual pictures from ones holding secret data inside them. When learning, the discriminator gets better at knowing real pictures from those the generator makes. At the same time, the generator works to minimize a cost function combining adversarial loss with reconstruction error using the MSE. The weighting parameter helps adjust if pictures look good versus how well the message stays hidden. This plan makes steganography tougher to beat with normal and automated detection techniques.

Block 3: Distributed Transmission

The steganographed image is divided into n segments and sent through various secure nodes or channels. The complete image is reassembled only after all fragments have been received. The private RSA key can be protected using the technique known as Shamir Secret Sharing in order to prevent a single point of failure. This method involves breaking up the class into multiple sections. $K \rightarrow \{k_1, k_2, \ldots, k_t\}$ where a minimum number of components $k < t$ is required to reconstruct the entire class. This makes security stronger by making it impossible to get to the key by compromising only one node.

This mechanism ensures:

- Resilience against faults and targeted attacks
- A charge distribution between different nodes
- Reinforcement of keys to ensure confidentiality

Algorithm 3: Secure Distributed Transmission

Input: *Is* : Steganographic image, *n* : Number of fragments, *{N1, …, Nn}* : Nodes
Output: Securely distributed fragments to the nodes

```
Divide Is into n fragments {f1, …, fn};
For i ← 1 to n do
  If the channel to Ni is now available, then
     Transmit fi to Ni using TLS;
  Else
     Reassign fi to another secure node.
```

Algorithm 3 shows a process for safely sending data within pictures across a network in a spread-out way. First, the picture with the hidden data gets split into many parts, lowering the risk of losing all the information if someone only grabs a few parts. After this, each part goes to a different spot in the distributed system, assuming there is a working way to talk to it.

These parts are sent using the TLS protocol, which keeps the shared data private, unchanged, and able to be checked. If a spot or way to talk stops working, the parts get moved to other safe spots right away. This way makes the system better at managing problems, stronger overall, and more secure when sending things in changing distributed environments.

Experimentation and Results

The proposed model was implemented in a technical environment suited to the requirements of computation, simulation, and experimental validation. This environment combines hardware, software, and development platforms. The experiments were conducted on a 64-bit machine equipped with four cores, each operating at 2.40 Hz. Python libraries were used to implement the model, as well as the Google Colab tool for execution on the GPU in the cloud and code sharing.

The Visual Difference between the Two Images

Figure 2 presents a comparison between the original cover image (left) and the steganographic image (right) obtained after embedding the encrypted message through the proposed steganography process.

Fig. 2: Visual distinction between the cover image and the steganography image

The starting cover image appears to the left, and the image to the right displays what happens after hiding the data.

At first glance, no perceptible difference can be observed between the two images. The steganographic image preserves the main visual characteristics of the original, including colours, textures, shapes and fine details. This visual similarity indicates that the embedding process does not introduce alterations that a human observer could detect.

This result illustrates the effectiveness of the proposed steganographic method, and in particular its imperceptibility, a key property of steganography. It shows that the hidden data can be embedded while preserving a high visual quality, which makes the proposed system more difficult to detect visually and more reliable.

Fragmentation

Figure 3 illustrates how the steganographic image is fragmented within the proposed secure transmission framework. After the encrypted message has been embedded into the cover image using GAN-based steganography, the resulting image is divided into several segments, denoted Fragment 1 to Fragment 4.

Each fragment contains only a small portion of the complete steganographic image; taken in isolation, a single fragment is not sufficient to reconstruct the full image or to recover the hidden message. This fragmentation strengthens the security of the system, since intercepting only a few fragments does not reveal any usable information.

Fragmentation is also a core element of the proposed model, in which each fragment can be transmitted or stored on a different network node or cloud location. This strategy reduces the risk of interception, tampering, or data loss, while improving the overall robustness of the secure transmission system.

Figure 3 therefore highlights the benefit of fragmentation when combined with steganography and hybrid AES-RSA encryption, reinforcing data confidentiality and security in distributed environments.

Reconstructed Image Following Fragmentation

Figure 4 shows the reconstruction of the steganographic image from the fragments distributed across the system. The fragments of the steganographic image, dispersed over different network nodes or stored in the cloud, are securely retrieved after the secure transmission stage.

The first step consists of collecting all the required fragments and arranging them in their correct order. A single fragment contains only a fraction of the data and does not allow the hidden information to be recovered. Once all the fragments are available, they are reassembled to reconstruct the original steganographic image with negligible loss of visual quality.

Fig. 3: Fragmentation of the steganographic image into four segments before distributed transmission

Fig. 4: Reconstruction of the steganographic image from the distributed fragments.

The reconstructed image is then passed to the data extraction module, which uses the GAN to recover the encrypted message embedded in the image. Finally, the recovered message undergoes hybrid AES-RSA decryption to restore the original plaintext.

The reconstruction process is essential to the security of the system, since it ensures that only authorized parties possessing all the required fragments can rebuild the complete image and access the sensitive data. This design enhances confidentiality, resistance to interception, and the overall robustness of the system in a distributed setting.

Histogram of Cover Image Versus Stego Image

Figure 5 presents the histogram of the cover image, which describes the distribution of pixel intensities and serves as the reference for data hiding. This histogram shows the intensity distribution across the image before any embedding or data-hiding operation is performed.

The grey-level distribution (or the RGB colour channels) appears regular and continuous, reflecting the natural characteristics of the image. This smooth distribution indicates the absence of artificial alterations or statistical anomalies, confirming that the image is intact and unmodified.

The histogram of the cover image is essential for assessing how effectively the steganographic scheme hides and protects data. After the secret data has been embedded, comparing the histogram of the cover image with that of the steganographic image reveals the extent to which the embedding modifies the pixel distribution.

A minimal difference between the two histograms indicates effective data hiding, which is a key requirement in steganography. Figure 5 thus provides a baseline demonstrating that the proposed method preserves the statistical properties of the cover image, making the hidden message difficult to detect.

Figure 6 presents the histogram of the steganographic image, showing the distribution of pixel intensities after the encrypted data has been embedded into the cover image using GAN-based steganography. The histogram describes how pixel intensities are distributed across the image once the hidden information has been inserted.

A close inspection of the histogram shows that the grey-level distribution and the values of the RGB channels remain very close to those of the original image. The observed variations are small, which means that the pixel modifications introduced during embedding are almost imperceptible and statistically negligible.

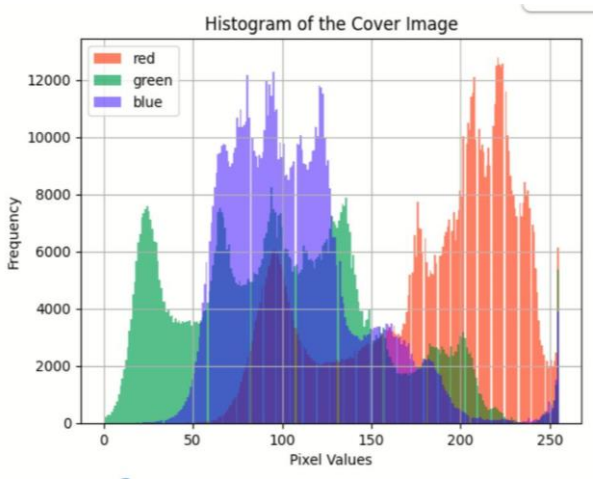

Fig. 5: Histogram of the cover image (pixel-intensity distribution before embedding)

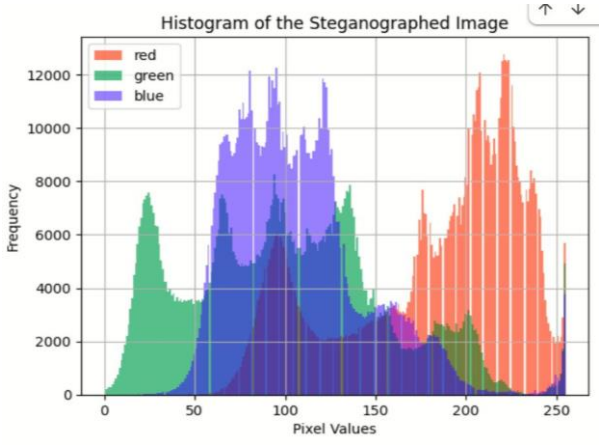

Fig. 6: Histogram of the steganographic image (pixel-intensity distribution after embedding)

The similarity between the histograms of the original and the steganographic images demonstrates how difficult it is for the proposed method to detect. The absence of significant changes in the pixel distribution makes it hard to detect any hidden information using conventional statistical analysis.

Moreover, the use of a GAN improves the embedding of the secret message while preserving the visual appearance and the statistical properties of the original image. The histogram in Figure 6 confirms that GAN-based steganography achieves a favourable trade-off between embedding capacity, visual quality, and security, while reducing the risk of detection by steganalysis.

Metrics Utilized

The measures used in this work are:

- PSNR: peak signal-to-noise ratio
- SSIM: structural similarity
- Capacity: maximum size of hidden data
- Calculation Time: integration and extraction

Values Obtained

At the end of the experiment, the performance of the proposed model was assessed using various standard metrics in steganography and image processing.

Numerical Results

The numerical results obtained in this work are:

- Average PSNR: 80.87 dB
- SSIM: 0.9519
- Recovery rate: 100 %
- Average integration time: 0.006 s
- Average extraction time: 0.002 s

Calculation Methodology

PSNR (Peak Signal-to-Noise Ratio)

The PSNR is determined based on the Mean Squared Error (MSE) between the cover image I and the steganographic image Î:

$$MSE = \frac{1}{M \times N} \sum_{i=1}^{M} \sum_{j=1}^{N} \left[I(i,j) - \hat{I}(i,j) \right]^2$$

Where *M* and *N* represent the width and height of the image. The PSNR is then expressed as follows:

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

Where MAX_I represents the maximum possible value of a pixel (255 for an 8-bit image).

SSIM (Structural Similarity Index Measure)

The SSIM measures the structural similarity between I and Î:

$$SSIM\left(I, \hat{I}\right) = \frac{\left(2\mu_I \mu_{I,\hat{I}} + C_1\right)\left(2\sigma_{I\hat{I}} + C_2\right)}{\left(\mu_I^2 + \mu_{\hat{I}}^2 + C_1\right)\left(\sigma_I^2 + \sigma_{\hat{I}}^2 + C_2\right)}$$

Where:

- μ_I, $\mu_{\hat{I}}$: average luminance levels
- σ^2_I, $\sigma^2_{\hat{I}}$: variances
- $\sigma_{I\hat{I}}$: measure of covariance
- C_1, C_2 : stabilization constants

Recovery Rate

The recovery rate indicates the percentage of bits that were accurately extracted compared to the original bits:

$$Taux\_recup = \frac{Correct\ bit\ count}{Total\ number\ of\ bits} \times 100\%$$

In our case, the recovery rate is 100%, which indicates that there are no errors.

Comparison With LSB and LSB-DCT

Table 1 displays the PSNR and SSIM values for various approaches from the literature. The proposed method yields an impressive PSNR value along with remarkable image quality. The achieved PSNR is 80.87 dB, and the SSIM is 0.9519, both of which are superior compared to LSB and LSB-DCT.

Interpretation of the Results

PSNR (Peak Signal-to-Noise Ratio)

- LSB: 36.08 dB → Acceptable quality, but a more noticeable visual decline
- LSB-DCT: 55.11 dB→Good visual quality, changes in the image are less noticeable
- Proposed Method (Hybrid AES-RSA + GAN): 80.87 dB → Excellent quality, with almost no noticeable visual difference between the original image and the steganographed image, resulting in an encrypted data size of 371 bytes

Table 1: Performance comparison based on the proposed method

Method	PSNR (dB)	SSIM	Strength
LSB	36.08	0.9872	Weak
LSB-DCT	55.11	0.9700	Average
Proposed approach	80.87	0.9519	High

The hybrid approach significantly surpasses the other two in visual quality, which is essential to avoid detection.

SSIM (Structural Similarity Index)

- LSB: 0.9872 → the structure of the image is well preserved, although not flawless
- DCT: 0.9700 → slight loss of structure due to transformation, but still high
- Proposed: 0.9519 → SSIM is slightly lower than LSB, but this is offset by a very high PSNR, indicating that the visual impact remains minimal

The hybrid method maintains a strong structural similarity while incorporating more robust encryption and coding.

Strength

- LSB: Weak → Vulnerable to compression and attacks
- DCT: Average • Handles JPEG compression and minor modifications more effectively
- Proposed: Elevated → Will withstand attacks, compression, and noise due to the integration of GAN-based steganography and encryption

Computational Latency Under Distributed High-Traffic Conditions

Beyond image quality, the practical value of the proposed architecture depends on the computational latency it introduces when deployed in a distributed environment under heavy load. To quantify this overhead, we measured the end-to-end latency of four configurations: A no-security baseline (plain image transfer), AES-RSA encryption only, LSB-DCT steganography only, and the proposed hybrid approach (AES-RSA + GAN steganography + distributed transmission). For each configuration, the number of concurrent requests was increased from 1 to 500, and we report the mean latency, the 95th-percentile (p95) latency, and the sustained throughput.

Table 2 and Figure 7 summarize the results. Two observations stand out. First, the proposed approach is dramatically faster than LSB-DCT steganography: At 500 concurrent requests, its mean latency is 208.9 ms against 6908.5 ms for LSB-DCT, i.e., roughly 33 times lower, and the gap reaches a factor of about 90 at intermediate loads. The sustained throughput of the proposed approach (around 150 req/s) is approximately 17 times higher than that of LSB-DCT (around 9 req/s).

Second, and perhaps counter-intuitively, the proposed approach also outperforms the no-security baseline under load (208.9 ms vs 1534.7 ms at 500 requests). This is explained by the distributed design: Whereas the baseline transfers each full image as a single stream, the proposed approach fragments the stego image and transmits the segments in parallel across several nodes.

This parallelism more than compensates for the cost of the GAN embedding and AES-RSA encryption. The genuine computational cost of the security layer is best seen by comparing the proposed approach with AES-RSA-only encryption.

Table 2: End-to-end latency, p95 latency, and throughput under increasing concurrent load (1–500 requests).

Configuration	Load	Mean (ms)	p95 (ms)	Thr. (req/s)
No security (baseline)	1	31.99	32.51	30.8
	10	205.88	230.73	43.6
	50	902.83	1413.62	33.6
	100	1132.69	1965.54	38.1
	200	1417.70	2374.57	37.5
	500	1534.65	2349.47	38.2
AES-RSA only	1	5.97	12.98	131.8
	10	8.17	24.01	290.7
	50	12.13	34.59	344.7
	100	7.34	19.55	540.4
	200	7.74	21.55	511.4
	500	24.20	95.38	598.3
LSB-DCT only	1	109.73	135.38	9.2
	10	1053.76	1289.78	8.5
	50	4891.64	7530.55	8.1
	100	5601.74	7841.78	8.8
	200	6543.17	9182.32	8.8
	500	6908.49	9301.66	8.8
Proposed approach	1	6.10	7.01	155.8
	10	28.89	45.96	162.3
	50	54.73	99.04	166.7
	100	61.07	107.81	165.5
	200	77.59	159.64	141.1
	500	208.86	517.02	149.7

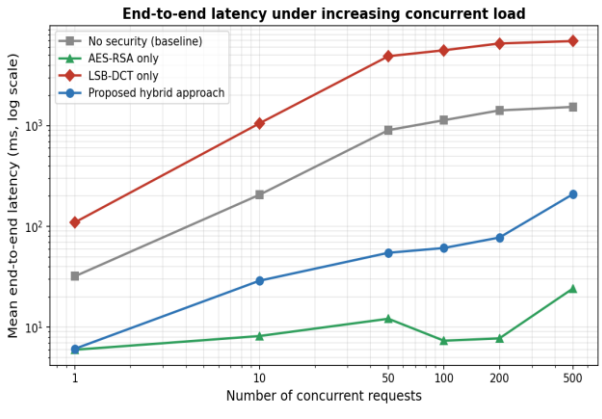

Fig. 7: End-to-end latency under increasing concurrent load (log scale)

The additional latency introduced by the GAN steganography and fragmentation stages remains modest, ranging from about 0.1 ms at low load to 185 ms at 500 concurrent requests, and the p95 latency stays bounded (517 ms at the highest load), confirming predictable behaviour under stress.

Overall, these results demonstrate that the proposed hybrid scheme adds only a limited and predictable overhead while remaining far more scalable than conventional steganographic methods. The combination of GAN-based embedding and parallel distributed transmission makes the approach suitable for high-traffic distributed deployments, where classical LSB-DCT steganography would constitute a severe bottleneck.

Resilience against Modern Steganalysis

The evaluation of the quality of the steganographic images using PSNR, SSIM, and MSE confirms a high level of perceptual imperceptibility, indicating negligible distortion between the original image and the carrier image, which remains indistinguishable to the human eye. The method therefore does not significantly alter the statistical characteristics that automated steganalysis tools typically exploit to reveal the presence of a hidden message, thus constituting an initial favourable indicator of its security. A quantitative evaluation against state-of-the-art steganalysers is currently underway and will enable this resistance to be measured by the probability of detection error.

Discussion

The experimental results demonstrate that the proposed hybrid architecture significantly improves data security and visual quality compared to traditional steganographic approaches such as LSB-DCT. High PSNR values indicate that the steganographic images remain visually very close to the original cover images, making the hidden information difficult to detect through visual inspection or basic steganalysis techniques. Similarly, high SSIM values confirm that the structural integrity of the images is preserved during the message embedding process.

These results indicate that the integration of a Generative Adversarial Network (GAN) into the steganography process significantly enhances the invisibility of hidden data compared to the traditional hybrid LSB-DCT method. This improvement is reflected not only in the high PSNR and SSIM values but also in the model's ability to maintain these performances even after the application of disturbances such as JPEG compression or noise addition**.**

The use of GANs enables the automatic optimization of information encoding, making the modifications practically imperceptible to the human eye and difficult for traditional steganalysis tools to detect. This adaptability, which is a characteristic feature of deep learning methods, represents a major advantage over deterministic approaches, as the model can learn an optimal embedding strategy that minimizes visual distortion while preserving the secrecy of the hidden data.

Furthermore, the integration of AES-RSA encryption strengthens the confidentiality of the hidden message by protecting it before the embedding stage. Even if the concealed data is extracted by an attacker, the message cannot be interpreted without the appropriate cryptographic keys. This combination of cryptography and steganography therefore creates an additional layer of security for protecting sensitive information.

In addition, the distributed transmission mechanism improves the system's resilience and fault tolerance. By fragmenting the steganographic image and transmitting it across multiple network nodes, the system reduces the risk of complete data interception. Even if a node or communication channel fails, data reconstruction remains possible, which significantly reduces the risk of information loss and enhances the overall robustness of the architecture in distributed environments.

Finally, the performance metrics related to computation time indicate that the proposed model remains competitive with conventional methods. Although the use of GANs requires more computational resources during the training phase, this cost is largely compensated by the robustness, visual quality, and level of security achieved during the operational phase.

In summary, the combination of AES-RSA encryption + GAN-based steganography + distributed transmission represents a robust and efficient approach for securing and concealing data in distributed environments while maintaining a satisfactory balance between security, invisibility, and efficiency.

Conclusion

In this article, we proposed an innovative hybrid security architecture designed to improve data protection in distributed environments. The proposed model relies on the integration of three main components: A hybrid AES–RSA encryption mechanism to ensure confidentiality and secure key management, a GAN-based steganography technique to conceal information within digital images in a discreet and robust manner, and a distributed transmission mechanism that enables the fragmentation and secure transmission of data across multiple network nodes.

The experimental results obtained from images of different resolutions demonstrate that the proposed approach provides superior performance compared to traditional steganographic methods such as LSB-DCT. The evaluation based on PSNR and SSIM metrics shows that the

steganographic images maintain high visual quality while ensuring effective data concealment. Furthermore, the integration of a distributed transmission system enhances the resilience of the model against potential attacks and reduces the risks associated with complete data interception.

The main contributions of this work can be summarized as follows:

- The proposal of a hybrid data security model combining cryptography, steganography, and distributed transmission
- The integration **of** GAN-based steganography to improve hiding capacity and resistance to steganalysis techniques
- The use of a hybrid AES–RSA encryption scheme ensuring both fast data encryption and secure key distribution
- The implementation of a fragmentation and distributed transmission mechanism aimed at improving system robustness and fault tolerance

We have introduced an innovative hybrid model that combines three key components: symmetric encryption using AES for speed and data security, asymmetric encryption with RSA for the protection and secure distribution of keys, and GAN-assisted steganography to ensure the invisibility and resilience of hidden data. Furthermore, the architecture is built on a distributed transmission system that facilitates the fragmentation of steganographed data, allowing it to be sent through various nodes and securely reassembled on the receiving end. This method enhances resilience against targeted attacks and minimizes the risks associated with a single point of failure. The experimental results revealed a strong ability to conceal information, with high PSNR and SSIM values, as well as significant resistance to compression, noise, and blurring, all while keeping processing times within a reasonable range.

Possible avenues for enhancement include:

- The incorporation of more advanced GANs, like StyleGAN or diffusion models, to create even more realistic and undetectable steganographic images
- The enhancement of distributed key sharing and storage mechanisms for RSA by investigating methods such as Shamir Secret Sharing
- Adapting the model for other types of media (audio, video) to broaden the range of use cases
- The implementation of an adaptive management system for encryption and steganography parameters based on the context and communication channel

In summary, this work illustrates the feasibility and effectiveness of a hybrid security model that combines cryptography and artificial intelligence, providing comprehensive data protection in distributed environments while maintaining a balance between security, invisibility, and efficiency.

Acknowledgment

We would like to express our sincere gratitude to the Laboratory of Applied Computer Science and Industrial Systems Intelligence (LR-IASI) at the University of Ngaoundere in Cameroon, for providing us with a suitable environment for our experiments.

Funding Information

The researchers affirm they did not obtain any monetary grants or fiscal assistance for conveying the details of this research project.

Author's Contributions

Derrick Méthode Bagaza: Prepared the initial version of the manuscript, collected the data, and conducted the experiments.

Abraham Walake: Carried out the literature review and revised the manuscript.

Mamoudou Kourouma: Contributed to the literature review and revised the manuscript.

Blaise Omer Yenke: Supervised all aspects of the study, analyzed the data, and reviewed and edited the manuscript.

Ethics

This written piece is a completely new and unique creation. The individuals who created this document affirm that putting forward this piece of work does not bring up any issues related to moral principles.

References

Adee, R., & Mouratidis, H. (2022). A Dynamic Four-Step Data Securty Model for Data in Cloud Computing Based on Cryptograpy and Steganography. *Sensors*, *22*(3), 1109. https://doi.org/10.3390/s22031109

AlKhamese, A. Y., Shabana, W. R., & Hanafy, I. M. (2019). Data Security in Cloud Computing Using Steganography: A Review. *2019 International Conference on Innovative Trends in Computer Engineering (ITCE)*, 549–558. https://doi.org/10.1109/itce.2019.8646434

Bagaza, D. M., Yenke, B. O., & M'boliguipa, J. (2024). Integration of the Triple Block Data Security Model Based on Distributed Crypto-Steganography in a Cluster. *Research in Computer Science*, *2085*, 180–190. https://doi.org/10.1007/978-3-031-63110-8_15

Bender, W., Gruhl, D., Morimoto, N., & Lu, A. (1996). Techniques for data hiding. *IBM Systems Journal*, *35*(3.4), 313–336. https://doi.org/10.1147/sj.353.0313

Chi-Kwong, C., & Cheng, L.-M. (2004). Hiding data in images by simple LSB substitution. *Pattern Recognition*, *37*(3), 469–474. https://doi.org/10.1016/j.patcog.2003.08.007

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2020). Generative adversarial nets. *Advances in Neural Information Processing Systems*, 2672–2680.

Huo, L., Chen, R., Wei, J., & Huang, L. (2024). A High-Capacity and High-Security Image Steganography Network Based on Chaotic Mapping and Generative Adversarial Networks. *Applied Sciences*, *14*(3), 1225. https://doi.org/10.3390/app14031225

Kaur, H., & Rani, J. (2016). A Survey on different techniques of steganography. *MATEC Web of Conferences*, *57*, 02003. https://doi.org/10.1051/matecconf/20165702003

Li, L., Zhang, X., Chen, K., Feng, G., Wu, D., & Zhang, W. (2024). Image Steganography and Style Transformation Based on Generative Adversarial Network. *Mathematics*, *12*(4), 615. https://doi.org/10.3390/math12040615

Malik, K. R., Sajid, M., Almogren, A., Malik, T. S., Khan, A. H., Altameem, A., Rehman, A. U., & Hussen, S. (2025). A hybrid steganography framework using DCT and GAN for secure data communication in the big data era. *Scientific Reports*, *15*(1), 19630. https://doi.org/10.1038/s41598-025-01054-7

Nechvatal, J., Barker, E., Dodson, D., Dworkin, M., Foti, J., & Roback, E. (1999). Status report on the first round of the development of the advanced encryption standard. *Journal of Research of the National Institute of Standards and Technology*, *104*(5), 435. https://doi.org/10.6028/jres.104.027

Patil, H. V., & Sonaje, V. P. (2025). Enhancing Security Through Dual Layer Techniques for Encrypting Text Messages Within Images Using LSB Image Steganography and AES Encryption Algorithms. *International Conference on Technology Advances for Green Solutions and Sustainable Development*, *56*, 249–258. https://doi.org/10.1007/978-3-031-94997-5_23

Rajguru, S. S., Singh, G., Malhi, S. S., & Kaur, G. (2024). Stenographic Approaches for Enhancing Data Security in Cloud Computing. *E3S Web of Conferences*, *556*, 01012. https://doi.org/10.1051/e3sconf/202455601012

Shwaysh, M. M., Alani, S., Saad, M. A., & Abdulhussein, T. A. (2024). Image Encryption and Steganography Method Based on AES Algorithm and Secret Sharing Algorithm. *Ingénierie Des Systèmes d Information*, *29*(2), 705–714. https://doi.org/10.18280/isi.290232

Stallings, W. (2006). *Cryptography and network security, 4/E*.

Tang, W., Rao, Y., Yang, Z., Peng, F., Cui, X., Huang, J., & Zhu, P. (2025). Reversible generative steganography with distribution-preserving. *Cybersecurity*, *8*(1), 18. https://doi.org/10.1186/s42400-024-00317-6

Zhang, Y., Wang, F., Chao, J., Xie, M., Liu, H., Pan, M., Kopperger, E., Liu, X., Li, Q., Shi, J., Wang, L., Hu, J., Wang, L., Simmel, F. C., & Fan, C. (2019). DNA origami cryptography for secure communication. *Nature Communications*, *10*(1), 5469. https://doi.org/10.1038/s41467-019-13517-3

Izevbizua, P. O. (2025). Data Security in the Cloud using Serpent Encryption and Distributed Steganography. *European Scientific Journal*, *11*(8), 347-359.