



International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: <https://www.svedbergopen.com/>



Research Paper

Open Access

Quantum-Based Steganography on AI-Generated and Real Images: A Comprehensive Experimental Analysis and Security Evaluation

Asha Durafe¹, Manisha Mane², Mamta Tikaria³, Dr. Pravin Shinde⁴, Anita Nalawade⁵

¹Electronics and Computer Science Department, Shah & Anchor Kutchhi Engineering College, Mumbai, MH, India. Email: asha.durafe@sakec.ac.in,

²Electronics and Computer Science Department, Shah & Anchor Kutchhi Engineering College, Mumbai, MH, India. Email: manisha.mane@sakec.ac.in

³Electronics and Telecommunication Department, Shah & Anchor Kutchhi Engineering College, Mumbai, MH, India. Email: mamta.tikaria@sakec.ac.in

⁴Artificial Intelligence & Data Science Department, Shah & Anchor Kutchhi Engineering College, Mumbai, MH, India. Email: pravin.shinde@sakec.ac.in

⁵Electronics and Computer Science Department, Shah & Anchor Kutchhi Engineering College, Mumbai, MH, India. Email: anita.nalawade@sakec.ac.in

Abstract

Steganography provides an important mechanism for secure covert communication by concealing sensitive information within digital media. However, conventional steganographic approaches may be affected by predictable key-generation mechanisms, statistical detectability, and vulnerabilities to steganalysis. This paper presents a quantum-based steganographic framework that integrates quantum key generation with Least Significant Bit (LSB) image steganography and evaluates its performance on both real and AI-generated images. The quantum keys are created based on the quantum circuit design which uses the Hadamard transform. In particular, a single qubit initialized to $|0\rangle$ is transformed into a uniform superposition using a Hadamard gate, followed by quantum measurement to generate unbiased binary key bits. The required (L)-bit quantum key is obtained by collecting (L) individual measurement outcomes using the Qiskit framework and Qiskit AerSimulator. Then, the keys are utilized for the embedding and extraction procedures. The proposed approach was tested on a dataset which consisted of 200 RGB images. There were 100 real images and 100 AI-generated images in the test set. First, a fixed payload of 2,296 bits was embedded into each image and evaluated using image quality metrics, extraction accuracy, embedding efficiency, steganalysis, and compression robustness. Scalability was then evaluated by increasing the payload size from 6,000 to 96,000 bits, representing a sixteen-fold increase in embedded data. As a result of experiments, the images with the embedded data were found to be visually similar to the original images because their Peak Signal-to-Noise Ratio (PSNR) values are higher, Mean Squared Error (MSE) values are negligibly small and Structural Similarity Index (SSIM) values are almost equal to one. Therefore, it can be stated that the embedding introduces almost no distortion to the image and the proposed method performs equally well on real and AI-generated images. Moreover, a number of additional tests were performed to check how easily the embedded data could be detected and recovered from images processed by different means. The experimental results show that the proposed method achieves high visual fidelity, reliable hidden message recovery and consistent computational performance on both kinds of images. This study shows the possibilities of the integration of quantum key generation and image steganography for the purposes of secure covert communication and can be considered as a starting point for further investigations with physical quantum systems and more complex steganalysis techniques.

Keywords: Quantum Computing; Quantum Steganography; Hadamard Transform; Quantum Superposition; Quantum Key Distribution; AI-Generated Images; LSB Embedding; Steganalysis; Image Security; Qiskit.

I. Introduction

Steganography comes from the Greek words "steganos," which means "covered" and "graphein," meaning "writing." It refers to hiding a message inside what looks like an ordinary carrier, in such a way that the recipient should not be able to detect any secret message [1]. In contrast, cryptography encrypts the message

to protect the content. While in steganography, the objective is to hide the very presence of a message, which introduces an extra layer of protection [2]. Due to the popularity of digital images, image steganography became widely used for concealing messages. Images are transmitted through current networks all the time and small variations in pixel values go unnoticed by the human eye [3].

There are many conventional algorithms designed to be used in image steganography, among them are LSB [4], DCT-based [5] and spread spectrum techniques. However, despite being widely investigated and applied, the conventional algorithms have a number of drawbacks as follows:

- Traditional systems use Pseudorandom Number Generators (PRNG) as key generators. Being deterministic, PRNG can generate predictable numbers, which is a potential danger in case of the attacker having enough computational power and information about the generation process [6].
- Steganalysis based on statistical analysis and machine learning can detect alterations in the distribution of pixel values and other patterns generated by hidden information [7][8].
- Most classical cryptography and steganography schemes rely on computational problems hard to solve by a classical computer. With the emergence of quantum computing, some of these underlying security assumptions may no longer hold [9].

Quantum computing allows generating a secure key using quantum mechanical phenomena such as superposition, entanglement and no-cloning theorem [10][11]. Quantum Random Number Generators (QRNG) generate random values through quantum measurements, which are inherently probabilistic and thus allow overcoming the determinism of classical generators [12]. The basic quantum gate used in the process is Hadamard gate, creating a superposition of a qubit and allowing random results on the measurement [13].

Along with the development of quantum computing, the advances in generative AI introduced a new type of realistic images created artificially. Generative Adversarial Networks (GAN) [14], Variational Autoencoders (VAE) and diffusion models [15] can produce images that look like photographs taken in reality. This leads to an important research question: whether images generated by generative artificial intelligence (GenAI) can effectively serve as carrier images alongside real images in image steganography.

II. Related Work

A. Classical Steganography

There have been tremendous advancements made in the area of image steganography over time, with a variety of techniques proposed by different scholars to improve the quality of the embedding process and increase resistance to detection. Fridrich et al. [16] have shown that applying LSB embedding directly may lead to statistical properties detectable using chi-square tests. This problem can be solved through LSB matching, which entails adjusting the value of pixels by either adding one to or subtracting one from them so as to make the modification more difficult to detect statistically [17]. There are also methods that work in the transform domain, with DCT being the most common method used to embed information in JPEG images [18], and DWT methods are used to embed information in wavelet coefficients in order to preserve the visual quality of images [19]. Boroumand et al. [20] have applied deep neural networks for both image steganography and steganalysis recently.

B. Quantum Cryptography and Quantum Random Number Generation

Quantum cryptography created a new model of secure communication, and one of the first models in this area was BB84 by Bennett and Brassard [21]. It uses basic principles of quantum mechanics to ensure the high level of security in key distribution. After that, Quantum Key Distribution (QKD) technology has developed not only within laboratory settings but also in commercial applications [22, 29]. Herrero-Collantes and Garcia-Escartin [23] give a thorough overview of different designs of QRNG and their methods of implementation. The Hadamard gate is also an important part of quantum computing and is used extensively to create superposition states in quantum computations and cryptographic algorithms [10].

C. AI-Generated Images

Generative AI has transformed the way that synthetic images are made. Generative adversarial networks (GANs) were introduced by Goodfellow et al. [14] as two neural networks competing against each other to generate increasingly realistic images. On this front, StyleGAN [24] achieved very high quality results, especially in face generation. More recently, diffusion based models such as Stable Diffusion [25] and DALL-E [26] have also improved upon the quality of text-to-image generation, allowing for realistic images to be generated from

textual descriptions. However, the use of AI-generated images as carrier images for steganography has been relatively less explored, even with these advances in image generation.

III. Proposed Quantum Steganography System

A. System Architecture Overview

The proposed framework is depicted as four stages of the experimental pipeline in Fig. 1. The stages consist of input preparation and quantum key generation, encryption and steganographic embedding, security and robustness assessment, and finally verification and statistical analysis. The general workflow aims to check the effectiveness of quantum-generated keys to secure LSB steganography with real and AI-generated images.

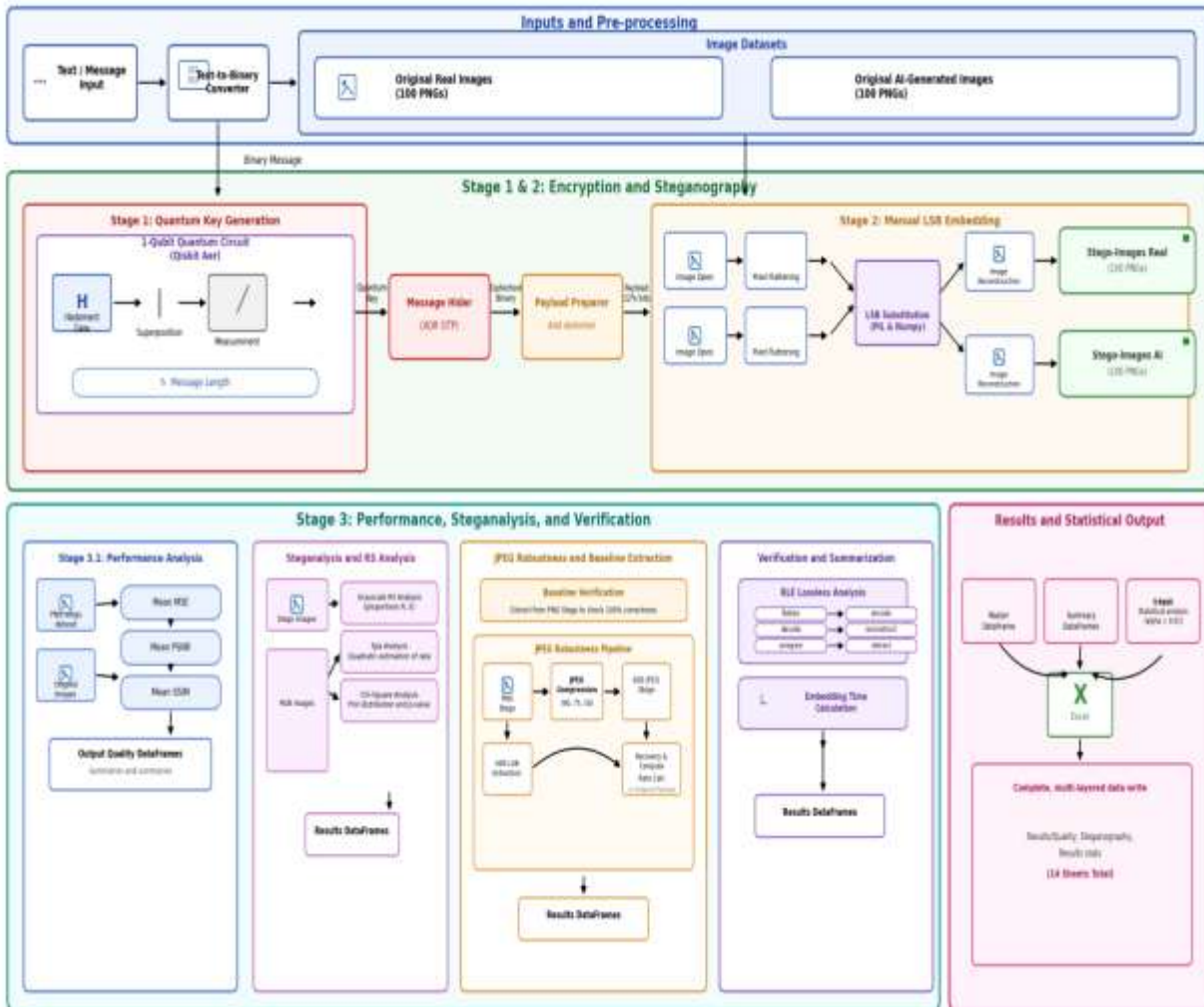


Fig. 1 — System Architecture Overview

Stage 1: Input Preparation and Cryptographic Payload Generation

3.1 Input Acquisition and Pre-processing

The experimental framework is based on two types of inputs: a secret message in plaintext and a set of cover images. The image dataset consists of 200 PNG images, of which 100 are genuine photographs and 100 are AI-generated images. The two groups are kept separate throughout the experiment so that the proposed method can be evaluated and compared on images with different statistical characteristics. Initially, the plaintext secret message is transformed into its binary equivalent, giving us the bitstream necessary for the following

cryptographic operations. The binary message is then transferred to the quantum key-generation module for further processing.

3.2 Quantum Key Generation

The quantum random key generation mechanism is implemented using Qiskit with Aer simulator. The key-generation circuit applies a Hadamard gate to a single qubit, resulting in an equal superposition of the computational basis states, followed by measurement to obtain a classical binary outcome. The circuit is run repeatedly until a key of the desired length is produced. The number of key bits generated is equal to the length of the binary plaintext, so the cryptographic key is sufficiently long for the desired One-Time Pad (OTP) operation. The generated quantum derived bit sequence is passed to the encryption module.

3.3 Encryption via Quantum One-Time Pad

A bitwise XOR operation of the plaintext bitstream with the quantum generated key in its binary representation is applied to encrypt the secret message. Since the key is generated with the same length as the plaintext and is for one-time use, the resulting ciphertext is following the operational principle of a One-Time Pad. A fixed 16-bit delimiter is appended to the encrypted bitstream to provide reliable payload extraction. The sequence generated is the final payload that is embedded into the cover images in the later steganographic stage.

Stage 2: Steganographic Embedding

3.4 Payload Embedding

A customized LSB substitution technique is used to hide the encrypted payload independently into all 200 cover images. The implementation uses numerical image processing operations to manipulate the pixel representation, while maintaining the spatial dimensions and format of the original images. For each cover image the pixel data is converted to a one dimensional representation and the least significant bit of each successive pixel value is substituted with the corresponding payload bit. After the embedding process is completed, the modified pixel array is reconstructed to the original image size and saved as a stego image. The generated images are stored in two datasets separately as Real Stego-Images and AI-Generated Stego-Images. Maintaining these data sets separately allows for later comparative analysis of image fidelity, statistical detectability, robustness and computational performance.

Stage 3: Security, Fidelity, and Robustness Evaluation

3.5 Fidelity and Performance Evaluation

The influence of the payload embedding on the perceptual and the structural quality of the images is investigated by comparing each stego image with the corresponding cover image. Three most popular image-quality metrics are considered: MSE that measures pixel-level distortion, PSNR that evaluates the overall image fidelity and SSIM that measures the extent to which the structural information of the original image is preserved. These metrics are calculated separately for the Real and AI-generated image subsets and then aggregated to evaluate how effectively the embedding process maintains the visual and structural characteristics of the images.

3.6 Steganalytic Evaluation

The statistical detectability of the embedded payload is evaluated using three complementary steganalysis techniques: Regular-Singular (RS) analysis, Chi-Square analysis, and Sample Pair Analysis (SPA). RS analysis examines the distribution of regular and singular pixel groups to detect statistical changes that may result from LSB embedding. Chi-Square analysis focuses on the frequency distribution of paired pixel values and uses the calculated statistical measure to determine whether there is evidence of data embedding. SPA provides another way to estimate the embedding rate by analyzing the relationship between neighboring pixel values. Using these three steganalysis methods together gives a broader view of the detectability of the proposed steganographic approach compared with using only a single statistical detector.

3.7 JPEG Robustness Analysis

The robustness of the employed payload during lossy compression is assessed in two stages. Initially, a baseline extraction process is done directly on the lossless PNG stego images in order to validate proper payload recovery before any modifications have been applied to the images. Subsequently, all stego images are subjected to JPEG compression with three different values of quality levels ($Q = 90, 75$, and 50) after which the process of LSB extraction is applied once more on the processed 600 JPEG stego images. Recovery rate is

determined by comparing the retrieved bitstream with the original payload for each level of compression, thus quantifying loss of payload integrity according to the values of compression.

Stage IV: Verification, Computational Analysis, and Statistical Validation

3.8 Independent Reconstruction and Verification

The verification stage is integrated into the procedure for verifying the integrity of the image processing system. Lossless intermediate format, namely Run-Length Encoding (RLE), is utilized in a process comprising a flatten-encode-decode-reconstruct sequence. The rebuilt image is compared against the original one to confirm the lossless process of reconstruction. Upon the verification completion, the embedded payload is also extracted to ensure that the additional procedure has not caused any damage to the experimental data. Finally, the time required for embedding images is also recorded for determining the efficiency of the proposed framework.

3.9 Results Consolidation and Statistical Analysis

In the course of the investigation, the outcomes generated from fidelity, steganalysis, robustness, verification, and performance experiments were compiled into a comprehensive experimental database. Separate summary datasets for the real images and AI-generated images have been created to enable a systematic comparison of all the examined parameters. The application of a paired statistical significance test with a significance level equal to $\alpha = 0.01$ serves to find out whether the differences between real images and AI-generated images can be deemed statistically significant. This provided quantitative proof necessary to assess whether the origin of images affects the parameters of performance and security of the proposed method.

B. Quantum Key Generation Implementation

The quantum key generation module was implemented using the IBM Qiskit framework [27] and the Qiskit Aer simulator. A single-qubit quantum circuit containing one qubit and one classical bit was constructed. The qubit was initialized in the computational basis state $|0\rangle$, followed by the application of a Hadamard gate to generate an equal superposition of the two computational basis states. The qubit was subsequently measured and the measurement result was mapped to the classical bit.

Let L denote the required key length in bits. Instead of employing an L -qubit quantum register, the proposed implementation uses the same single-qubit circuit and executes it for L measurement shots. The simulator was configured with $\text{shots} = L$ and $\text{memory} = \text{True}$, enabling the individual measurement outcome of every shot to be retrieved. Each shot produces a binary outcome, $k_i \in \{0,1\}$. The L measurement outcomes are subsequently concatenated to construct the quantum-generated key: $K = k_1 \parallel k_2 \parallel \dots \parallel k_L$, where $\parallel$ denotes concatenation. Consequently, a key of arbitrary length can be generated using a single-qubit circuit, with the required key length determined by the length of the binary representation of the secret message. In the experimental implementation, the generated key and binary message have identical lengths and are subsequently supplied to the XOR-based encryption stage.

Algorithm 1: Single-Qubit Quantum Key Generation

Input: Required key length L

Output: Quantum-generated binary key K

1. Create a quantum circuit QC containing 1 qubit and 1 classical bit.
2. Initialize the qubit in the state $|0\rangle$.
3. Apply the Hadamard gate H to qubit 0.
4. Measure qubit 0 and store the measurement result in classical bit 0.
5. Initialize the Qiskit AerSimulator.
6. Execute the constructed circuit using: $\text{shots} = L$, $\text{memory} = \text{True}$.
7. Return the L -bit quantum-generated key K .

V. Experimental Setup

The proposed quantum steganography system is systematically evaluated across real and AI-generated images using a comprehensive experimental framework.

A. Computational Environment

The proposed quantum steganography framework was implemented and experimentally evaluated using a dedicated computational environment to ensure consistent and reproducible results. The research work was carried out on an ASUS TUF Gaming F16 (FX607VBR/FX677VBR) laptop equipped with an Intel Core 5 210H

processor operating at 2.20 GHz with 8 cores and 12 threads, 16 GB RAM, and an NVIDIA GeForce RTX 3050 Laptop GPU with 4 GB VRAM. The system was configured with a 512 GB SSD and operated using Windows 11 Home Single Language, 64-bit.

The proposed framework was developed using Python 3.11.15 within an Anaconda environment. Quantum circuit implementation and simulation were performed using Qiskit 2.5.2 and Qiskit Aer 0.17.2, respectively. NumPy 2.4.6, Pandas 3.0.5, SciPy 1.17.1, scikit-image 0.26.0, Pillow 12.3.0, and OpenPyXL 3.1.5 were used for numerical computation, data processing, statistical analysis, image processing, and experimental-result management. The research implementation and experimental runs were executed through Jupyter Notebook in Visual Studio Code. The same computational and software environment was maintained throughout the implementation, embedding, extraction, robustness, steganalysis, and performance-evaluation stages to ensure consistency and reproducibility of the reported results.

B. Dataset Used

1) Real and AI-Generated Images

The data set that was used in this research was collated from the "AI-Generated Images v/s Real Images" data set that is publicly available on Kaggle [29]. The dataset consists of both authentic and artificially created images, and it was used to check the performance of the quantum steganography method for different sources of image data. The subset of the dataset that is meant for experimental evaluation consisted of 100 real images and 100 artificial images, giving a total of 200 images. The images were initially stored in JPEG format. Because JPEG format is subject to lossy compression, therefore, the images were converted to PNG format before being used for steganography tests. This has created an equal basis for all images involved in the experiments, thus allowing for uniform conditions of embedding and evaluation. All images were processed to have an equal size of 512×512 pixels, so that equal conditions of computations are created.

Overall, the final experimental data set consists of two classes with a similar number of inputs in each class, 100 authentic images and 100 artificial images. Thus, a total of 200 RGB PNG images with a resolution of 512×512 pixels were used for quantum steganography embedding and subsequent performance evaluation.

Table I — Dataset Composition and Characteristics [28]

Dataset Class	No. of Images	Original Format	Processed Format	Image Resolution	Image Type
Real	100	JPEG (.jpg)	PNG (.png)	512×512	RGB
AI-Generated	100	JPEG (.jpg)	PNG (.png)	512×512	RGB
Total	200	JPEG	PNG	512×512	RGB

Table I presents the composition and preprocessing details of the dataset used for the experimental evaluation. An equal number of real and AI-generated images were selected from the test set. To maintain a consistent image representation and ensure uniform experimental conditions, all selected images were converted from JPEG to PNG format and resized to 512×512 pixels.

C. Experimental Parameters

The following experimental parameters were maintained consistently across the evaluation to ensure a fair and reproducible assessment of the proposed quantum steganography method.

- **Message Size:** The proposed framework was evaluated across five payload sizes of 6000, 12000, 24000, 48000, and 96000 bits. For each target size, plaintext was drawn from a fixed narrative corpus, truncated to the required length (accounting for a 16-bit end-of-message delimiter), and converted to binary using 8-bit ASCII encoding. In addition, a fixed payload of 2296 bits per image was used as a baseline for the comparative evaluation of real and AI-generated cover images.
- **Quantum Key Length:** Rather than using a fixed-length key, the quantum key was generated dynamically for each experimental run, with its length set equal to the length of the plaintext message being encrypted (a one-time-pad scheme). The key was produced by placing a single qubit in an equal superposition state using a Hadamard gate, measuring it, and repeating this process for a number of shots equal to the required key length on Qiskit's AerSimulator backend with each shot contributing one truly random bit to the key.
- **Embedding Mode:** Sequential LSB embedding was used, in which message bits are embedded sequentially into the selected least significant bits of the cover image.

- **Quantum Simulator:** Quantum key-generation experiments were performed using the AerSimulator backend of IBM Qiskit. The number of measurement shots per experimental run was equal to the required key length in bits. A single qubit was prepared in superposition using a Hadamard gate and then repeatedly measured, each shot yielding one random bit, so that the overall number of shots (and thus random bits collected) was directly proportional to the payload size being tested.
- **Test Messages:** In all experiments, the source of the plaintext message was a single narrative text in English.
- **Cover Images:** The experimental evaluation used 200 RGB PNG images of size 512×512 pixels, comprising 100 real images and 100 AI-generated images. The same preprocessing and embedding configuration was applied to both classes.
- **Performance Metrics:** The proposed method was evaluated using PSNR, SSIM, MSE, embedding time, and extraction accuracy. Higher PSNR, SSIM, and extraction accuracy indicate better performance, whereas lower MSE and embedding time indicate reduced distortion and computational overhead, respectively.

D. Evaluation Methodology

The proposed quantum steganography method was thoroughly evaluated independently on the Real and AI-generated image datasets to analyze the impact of the method on the image imperceptibility, steganalysis performance, resistance to image compression, the accuracy of secret message extraction as well as the computational performance of the method.

Imperceptibility Assessment

For each cover image selected from the datasets, the secret message was embedded using the proposed quantum-key-based embedding procedure. Then, the resulting stego-image was analyzed using the following procedure:

- **Image quality evaluation:** the quality of the generated stego-images was assessed by comparing them with the originals using three well-known image quality assessment indices, namely PSNR, MSE and SSIM. Using the values of these indices, one can estimate how much information was lost during the embedding process. Particularly, PSNR and SSIM were used to assess the similarity between the original and stego-images on a perceptual level, whereas MSE represented the pixel-wise difference between the compared images. Thus, higher values of PSNR and SSIM and lower values of MSE indicated higher quality of the resulting stego-image.
- **Steganalysis:** the statistical detectability of the embedded information was assessed using the state-of-the-art steganalysis methods, including Chi-square attack, RS analysis and SPA. Using these methods, we estimated to which extent the embedding process altered statistical properties of the cover images that could be used in steganalysis.
- **Robustness Evaluation:** The robustness of the proposed method was estimated by analyzing the quality of message extraction after JPEG and RLE compression of the generated stego-images. Particularly, the following procedure was applied for both lossy (JPEG) and lossless (RLE) compression: The generated stego-images were compressed with quality factors of 90%, 75% and 50% (for JPEG compression) or encoded using RLE. Then, the compressed or encoded stego-images were used to extract the embedded secret messages, and the message recovery rate was calculated. This allowed us to estimate the impact of the compression on the performance of the proposed embedding method. For RLE encoding/decoding, since this procedure is lossless, it was also interesting to see if the pixel information was fully preserved during this process.
- **Extraction Accuracy:** The accuracy of the secret message extraction was evaluated by comparing the resulting message with the originally embedded one. Particularly, the extraction accuracy was calculated as the percentage of correctly extracted message bits. Thus, 100% accuracy meant that the entire message was retrieved.
- **Computational Performance:** The computational performance of the proposed method was estimated by analyzing the time needed to perform the embedding procedure. For each cover image from the Real and AI-generated image datasets, the embedding time was calculated and used for the comparison. To ensure sufficient statistical power, the experiments were conducted using the same hardware and software configurations.
- **Statistical Significance Analysis:** For each evaluation metric, the study performed a two-sample t-test to identify whether the difference in the values of this metric was statistically significant for the Real and AI-generated cover images. For all the tests, the significance level was set to $\alpha = 0.01$. The null hypothesis H_0 for these tests assumes that there is no statistically significant difference between the means of the distributions of the measurement values for the Real and AI-generated cover images, while the alternative hypothesis H_1 states the opposite. The statistical significance of the obtained results was analyzed using the corresponding t-statistic, degrees of freedom and p-value. If the p-value was below 0.01, the alternative hypothesis was accepted; otherwise, the null hypothesis was retained.

• **Repetition and Statistical Reporting:** In order to ensure that the obtained results were consistent across multiple experiments, each experiment was repeated five times using the same cover image but different quantum keys. In other words, for each cover image and each payload, the embedding procedure was executed five times with different quantum keys. The resulting values were then averaged, and the performance of the method was reported as the mean value $\pm$ standard deviation.

VI. Results and Discussion

The proposed steganographic method was evaluated using two categories of cover images: real-world images (Real) and AI-generated images (AI), as presented in Table II. For each category, 100 images were used with a fixed payload of 2296 bits per image. The performance of the method was evaluated using PSNR, SSIM, MSE, embedding time, and extraction accuracy.

Table II — Performance Comparison of the Proposed Steganographic Method on Real and AI-Generated Images

Dataset	No. of Images	Payload (bits)	PSNR (dB)	SSIM	MSE	Embedding Time (s)	Extraction Accuracy (%)
Real	100	2296	76.49	0.999997	0.00146	0.06023	100.0
AI	100	2296	76.51	0.999997	0.00145	0.05339	100.0

The experimental results demonstrate that the proposed method exhibits robustness and effectiveness in terms of high imperceptibility and precise message recovery for both real and AI-generated images. In particular, the PSNR, SSIM, and MSE metrics for the real-image dataset were 76.49 dB, 0.999997 and 0.00146, respectively, while those for the AI-generated image dataset were 76.51 dB, 0.999997 and 0.00145, respectively. This implies that the embedding process did not cause any noticeable distortions. In both cases, the payload extraction accuracy was 100%, which demonstrated that the 2296-bit payload was completely and accurately recovered without any single bit being lost or erroneously extracted. Besides, the proposed method shows a high speed of processing since the mean values of the payload embedding time for the real and AI-generated images were only 0.06023 s and 0.05339 s, respectively. Even though the time taken to embed the payload in the AI-generated images was slightly lower than that for the real images, this difference hardly impacted the overall result.

It can be concluded that the results were comparable for the two datasets, which indicates that the proposed method can be effectively applied to both real and AI-generated cover images. It is worth noting that the two datasets yielded nearly the same results in terms of high PSNR (76.49 dB or higher), SSIM (0.999997), and payload extraction accuracy (100%). Thus, the embedding process did not affect the quality of the cover images, and the method's performance was stable for different cover images.

6.1 Extraction Reliability Under Lossless vs. Lossy Compression

The experiments were carried out to evaluate the impact of two compression methods that differ markedly in their properties and approaches to information removal on the reliability of the payload extraction. One of the methods was RLE, a lossless compression algorithm, and the other was JPEG recompression, which is lossy. According to the results provided in Fig. 2(a), when RLE was used to compress the payload before embedding, the Real and AI groups both demonstrated an outstanding result in terms of the mean payload extraction accuracy and lossless reconstruction rate, which were 100% for both. Also, no noticeable differences were found between the two groups in any of the considered metrics across the entire set of 200 images. This can be explained by the fact that RLE is a lossless compression method, meaning that no information is lost during the compression process. In other words, RLE allows reconstructing the exact same bits as the source bits after decompression. Therefore, the entropy of the cover images, which plays a critical role in steganography, was not altered.

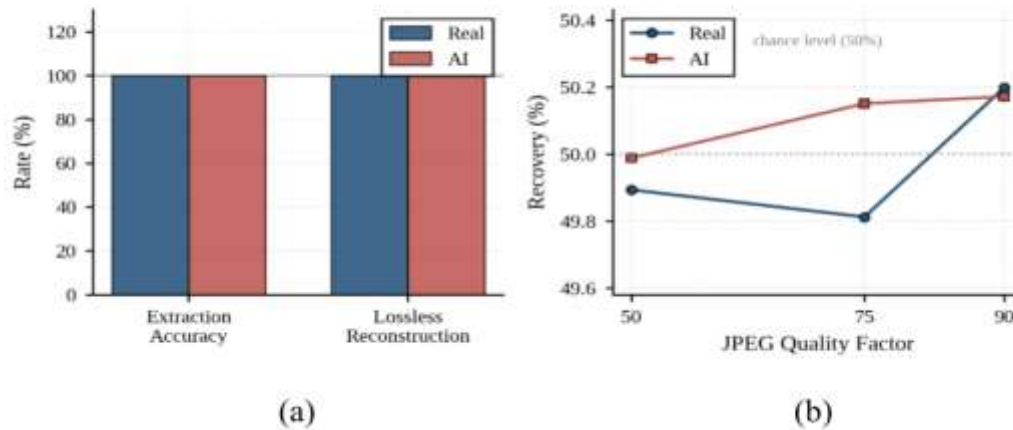


Fig. 2 — Extraction Reliability Under Lossless vs. Lossy Compression: (a) Lossless compression (RLE) and (b) Lossy compression (JPEG)

Figure 2(b) presents the contrasting case of JPEG recompression, applied to the stego image itself after embedding, at quality factors of 50, 75, and 90. In sharp distinction to the RLE result, recovery percentage remains clustered at the chance level of 50% across the entire range of quality factors tested, for both cohorts, with zero images in either group achieving complete and correct extraction. This divergence is likewise consistent with information-theoretic expectation: JPEG is a lossy codec that discards perceptually redundant information through block-wise discrete cosine transform quantization, a process that is agnostic to and therefore destructive of the least-significant-bit structure in which the payload is encoded. From the comparison of panels (a) and (b), it is clear that the failure to extract occurs as a direct result of the properties of the compression algorithm as far as information preservation, and not because of any problem within the embedding process or the source of the cover image itself, since the embedding process is deterministic regardless of the transform being lossless or not.

6.2 Computational Efficiency: AI Covers are Marginally Faster

Mean embedding time per image is shown in Fig. 3, is lower for AI (0.053 s) than Real (0.060 s), and AI shows a slightly better mean RLE compression ratio (0.572 vs 0.545) with comparable encode/decode times. Inference can be made that there is no meaningful computational penalty (if anything, a small efficiency edge) when using AI-generated covers.

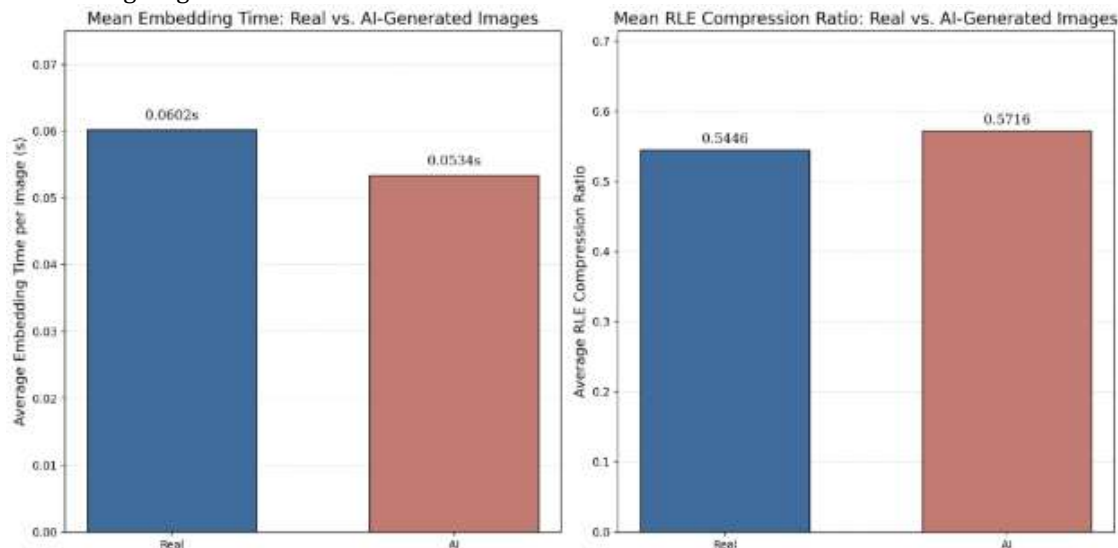


Fig. 3 — Mean Embedding Time and Mean RLE Compression Ratio: Real vs. AI-Generated Images**Table III — Statistical Test (t-test)**

Metric	Real Mean	AI Mean	AI – Real	T-Statistic	P-Value	Alpha
MSE	0.001458	0.001454	-4.53e-06	1.116	0.2658	0.01
PSNR (dB)	76.4923	76.5058	0.0134	-1.112	0.2676	0.01
SSIM	0.999997	0.999997	-1.83e-07	0.830	0.4078	0.01

To determine whether the choice of cover image source influences embedding-induced distortion, an independent two-sample t-test (Welch's t-test) was conducted comparing Real and AI-generated stego images across three image quality metrics: MSE, PSNR, and SSIM, at a significance level of $\alpha = 0.01$. As summarized in Table III, the mean MSE values for the Real and AI datasets were nearly identical (0.001458 vs. 0.001454), yielding a negligible difference of -4.53×10^{-6} and a p-value of 0.2658. Similarly, the mean PSNR values (76.4923 dB for Real and 76.5058 dB for AI) differed by only 0.0134 dB, corresponding to a p-value of 0.2676, while the mean SSIM values (0.999997 for both datasets) produced a p-value of 0.4078. Since all three p-values substantially exceed the chosen significance threshold of $\alpha = 0.01$, the null hypothesis cannot be rejected for any of the metrics, indicating that the observed differences between Real and AI-generated images are not statistically significant.

These results suggest that the proposed quantum-assisted LSB steganographic scheme introduces a statistically equivalent level of distortion regardless of whether the cover image originates from a real photograph or an AI-generated source. The consistently high PSNR (> 76 dB) and near-unity SSIM (0.999997) values across both datasets further confirm that the embedding process preserves visual and structural fidelity to a near-lossless degree. Taken together, these findings support the conclusion that the proposed method is robust and source-agnostic, performing with comparable imperceptibility on both natural and synthetic cover images.

As the payload increases from 6,000 to 96,000 bits, the mean MSE increases from approximately 0.0038 to 0.0610, while PSNR decreases from approximately 72.3 dB to 60.3 dB for both real and AI-generated images, however, SSIM remains above 0.9994 throughout, indicating that the proposed method as per Table IV maintains high structural similarity even at larger payloads. The closely matched results between real and AI-generated images across all payload levels further demonstrate the consistent and image-source-independent embedding performance of the proposed approach.

Table IV — Different Payload Comparison

Payload Size	Mean MSE (Real)	Mean PSNR (dB) (Real)	Mean SSIM (Real)	Mean MSE (AI)	Mean PSNR (dB) (AI)	Mean SSIM (AI)
6,000 bits	0.003805	72.3271	0.999987	0.003822	72.3084	0.999986
12,000 bits	0.007630	69.3059	0.999952	0.007640	69.2999	0.999949
24,000 bits	0.015263	66.2946	0.999872	0.015270	66.2925	0.999866
48,000 bits	0.030543	63.2817	0.999715	0.030531	63.2835	0.999711
96,000 bits	0.061047	60.2742	0.999421	0.061027	60.2756	0.999431

The results show a progressive increase in MSE and a corresponding decrease in PSNR and SSIM as the payload increases from 6,000 to 96,000 bits, confirming the expected trade-off between embedding capacity and image quality. The trends remain highly consistent between real and AI-generated images, with only marginal differences across all payload levels.

The MSE increases from approximately 0.0038 to 0.0610 for both real and AI-generated images, indicating gradually increasing embedding distortion with payload size as shown in Fig. 4a and b.

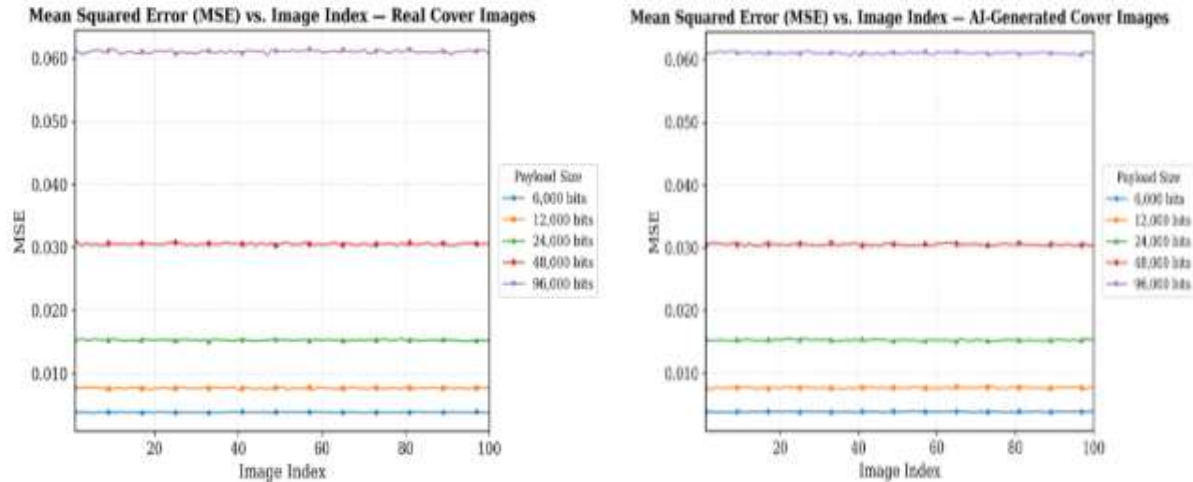


Fig. 4 — MSE vs. Image Index for (a) Real and (b) AI-Generated Cover Images

PSNR decreases from approximately 72.33 dB to 60.27 dB for real images and from 72.31 dB to 60.28 dB for AI-generated images, demonstrating reduced image fidelity at higher payloads (Fig. 5a and b).

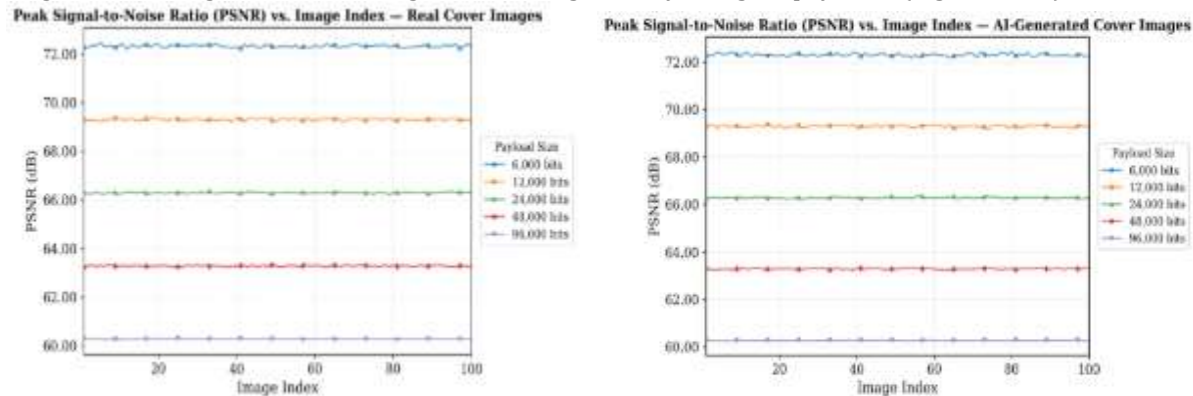


Fig. 5 — PSNR vs. Image Index for (a) Real and (b) AI-Generated Cover Images

SSIM remains very close to unity across all payload levels, although it gradually declines from approximately 0.99999 to 0.99942, indicating that structural similarity is largely preserved even at higher embedding capacities (Fig. 6a and b).

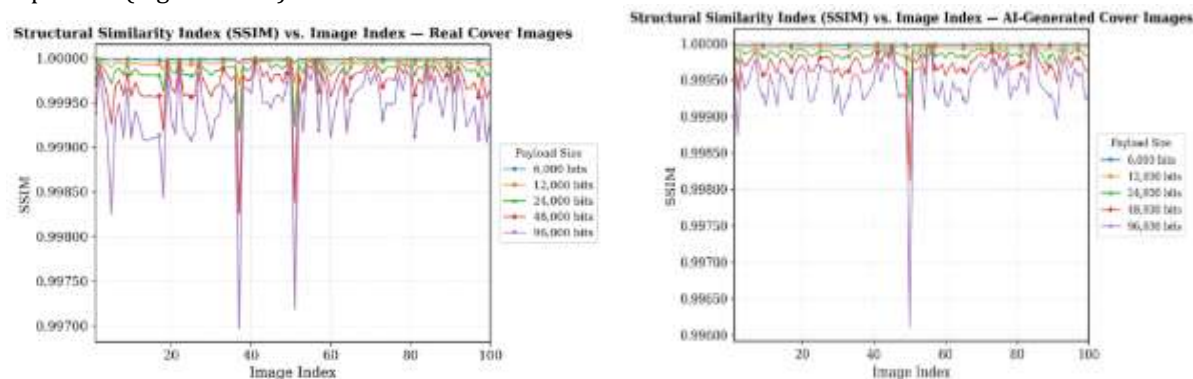


Fig. 6 — SSIM vs. Image Index for (a) Real and (b) AI-Generated Cover Images

In general, the almost overlapping tendencies obtained for the real and artificial images prove the similarity in the behavior of the proposed method on two types of images despite the increment in the payload capacity.

Table V presents the results obtained from the Chi-Square, RS, and SPA analyses used for the estimation of the statistical detectability of the embedded data in terms of various payload sizes. The Chi-Square test demonstrated large test statistic values with the p-values much lower than 0.05 for both real and AI-generated images at all considered payload sizes. It implies that the impact of the embedding is detectable statistically. On the other hand, the relatively low difference between the regular and singular groups in the RS test points out low detectability based on this technique. The estimates obtained in the SPA test show high estimated embedding rate values especially for the real-image dataset.

At the highest payload size (96,000 bits), the estimated embedding rate becomes equal to 0.777 for the real images and 0.613 for the AI-generated ones. Thus, the results show that there are more statistical artifacts detected when the payload capacity is increased. In general, the obtained results show that while the proposed method provides relatively low detectability according to the RS analysis, Chi-Square and SPA analyses demonstrate certain residual signatures.

Table V — Statistical Steganalysis Results Across Varying Payload Sizes

Parameter	6,000 bits	12,000 bits	24,000 bits	48,000 bits	96,000 bits
Chi-Square (Real)	20616.74	20020.91	19049.98	17366.93	14656.45
Chi-Square (AI)	16529.09	16209.10	15513.05	14349.19	12657.60
P-Value (Real)	2.74e-14	6.80e-14	3.61e-13	6.93e-12	1.17e-09
P-Value (AI)	4.26e-27	2.06e-25	2.21e-24	5.50e-22	1.72e-15
RS: R_M (Real)	0.282571	0.282737	0.282778	0.283362	0.283668
RS: R_M (AI)	0.276433	0.276740	0.276880	0.277740	0.278547
RS: S_M (Real)	0.281200	0.281349	0.281560	0.281826	0.282560
RS: S_M (AI)	0.274693	0.274909	0.275346	0.275977	0.277195
RS: R_m (Real)	0.282441	0.282609	0.282639	0.283139	0.283565
RS: R_m (AI)	0.276062	0.276354	0.276450	0.277247	0.277973
RS: S_m (Real)	0.281125	0.281263	0.281418	0.281766	0.282282
RS: S_m (AI)	0.274886	0.275118	0.275590	0.276176	0.277463
RS: R_M – S_M (Real)	0.001371	0.001387	0.001217	0.001537	0.001108
RS: R_M – S_M (AI)	0.001740	0.001831	0.001534	0.001763	0.001353
RS: R_m – S_m (Real)	0.001316	0.001346	0.001220	0.001373	0.001283
RS: R_m – S_m (AI)	0.001176	0.001236	0.000860	0.001071	0.000510
SPA: Est. Embedding Rate (Real)	0.756877	0.756743	0.756804	0.756872	0.776694
SPA: Est. Embedding Rate (AI)	0.529396	0.528904	0.528110	0.566140	0.613326
SPA: LSB Transition Rate (Real)	0.468232	0.468576	0.469001	0.469905	0.471479
SPA: LSB Transition Rate (AI)	0.438571	0.439261	0.440454	0.442978	0.446814
SPA: Mean Pair Difference (Real)	7.383014	7.383991	7.385728	7.389288	7.395778

Parameter	6,000 bits	12,000 bits	24,000 bits	48,000 bits	96,000 bits
SPA: Mean Pair Difference (AI)	7.821521	7.822884	7.825398	7.830522	7.839230

VI. Conclusion

The proposed framework combines quantum Hadamard transformation-based key generation with the conventional LSB substitution for information hiding. This paper investigates whether quantum key sequences can be embedded into a steganography framework preserving the visual quality of images, reliability of message extraction and performance consistency for both real images and AI-generated images. For the investigation, 200 RGB images, 100 of which are real and 100 are AI-generated images, were used, having gone through the uniform pre-processing and being converted to lossless PNG format.

The results of the experiment demonstrate that the proposed approach causes a negligible level of distortion of the cover images. With the 2296 bits payload size, PSNR for the real images is equal to 76.49 dB, whereas it is 76.51 dB for the AI-generated images. For the same payload size, MSE for the real images is equal to 0.00146 and it is 0.00145 for the AI-generated images. The SSIM is equal to 0.999997 for both sets of images, which means that the visual properties of the cover images are preserved during the embedding process. Moreover, both categories of the images reached 100% message extraction rate (lossless). In addition, the embedding times of 0.06023 s for real images and 0.05339 s for AI-generated images are measured, which indicates relatively low computational costs under the current experimental conditions.

To further assess the scalability of the proposed framework, the payload size was varied across a wider range (6,000, 12,000, 24,000, 48,000, and 96,000 bits) representing a sixteen-fold increase in embedded data. The results show a predictable and graceful degradation in image fidelity as payload size grows: PSNR decreases from 72.33 dB (Real) and 72.31 dB (AI) at 6,000 bits to 60.27 dB (Real) and 60.28 dB (AI) at 96,000 bits, corresponding to an approximate 3 dB reduction for every doubling of the payload consistent with the theoretical relationship between MSE and PSNR, since MSE scales approximately linearly with the number of embedded bits. Correspondingly, MSE increases from roughly 0.0038 at 6,000 bits to 0.0610 at 96,000 bits for both datasets, while SSIM remains consistently high throughout the sweep, declining only marginally from 0.99999 to 0.99942 even at the largest payload tested. Critically, the Real and AI-generated datasets track each other closely at every payload level, with differences in MSE, PSNR, and SSIM remaining negligible regardless of payload size. This confirms that the near-equivalence between real and synthetic cover images observed at the 2296-bit baseline is not an artifact of a single payload size but holds consistently as embedding capacity is scaled up, reinforcing the robustness and source-agnostic behavior of the proposed framework across a wide range of practical payload sizes.

To obtain a broader assessment of the proposed approach, besides the image quality metrics, the steganalysis, various levels of JPEG compression, lossless RLE-based processing, and statistical significance tests were performed as well. The comparison of the proposed embedding process performance on real images and AI-generated images showed its successful performance on both categories without any visual quality or message extraction deterioration.

The quantum key generation module was implemented and evaluated using the Qiskit Aer simulator. Consequently, the described characteristics of quantum randomness cannot be considered as experimental validation of the physical quantum hardware. However, the results prove the ability of quantum circuit key generation to be used within the steganography framework.

Although advanced embedding techniques may offer enhanced security, imperceptibility, robustness, and extraction reliability, this study focuses on conventional LSB-based steganography, as its primary objective is to evaluate a different aspect of information hiding rather than to compare advanced embedding methods.

Funding

This research received no external funding.

Author Contribution

A.D. and M.M. contributed to the conceptualization of the study and manuscript preparation. M.T., P.S. and A.N. carried out the implementation and validation of the proposed framework.

Data Availability

Publicly available in a repository: T. Zhang, "AI-Generated Images vs Real Images," Kaggle, 2023. [Online]. Available: <https://www.kaggle.com/datasets/tristanzhang32/ai-generated-images-vs-real-images>

References

1. Petitcolas, F. A. P., Anderson, R. J. & Kuhn, M. G. Information hiding—A survey, *Proc. IEEE*, vol. 87, no. 7, pp. 1062–1078, Jul. 1999. <https://doi.org/10.1109/5.771065>
2. Cox, I. J., Miller, M. L., Bloom, J. A., Fridrich, J. & Kalker, T. *Digital Watermarking and Steganography*, 2nd edn (Morgan Kaufmann, 2008).
3. Provos, N. & Honeyman, P. Hide and seek: An introduction to steganography, *IEEE Security Privacy*, vol. 1, no. 3, pp. 32–44, May/Jun. 2003. <https://doi.org/10.1109/MSP.2003.1203220>
4. Anderson, R. J. & Petitcolas, F. A. P. On the limits of steganography, *IEEE J. Select. Areas Commun.*, vol. 16, no. 4, pp. 474–481, May 1998. <https://doi.org/10.1109/49.668971>
5. Fridrich, J., Goljan, M. & Soukal, D. Perturbed quantization steganography with wet paper codes, in *Proc. ACM Workshop Multimedia Security*, pp. 4–15, 2004. <https://doi.org/10.1145/1022431.1022436>
6. Eastlake, D., Schiller, J. & Crocker, S. Randomness Requirements for Security, RFC 4086, Jun. 2005. <https://www.rfc-editor.org/rfc/rfc4086>
7. Qian, Y., Dong, J., Wang, W. & Tan, T. Deep learning for steganalysis via convolutional neural networks, in *Proc. SPIE*, vol. 9409, Art. no. 94090J, 2015. <https://doi.org/10.1117/12.2083479>
8. Ye, J., Ni, J. & Yi, Y. Deep learning hierarchical representations for image steganalysis, *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 11, pp. 2545–2557, Nov. 2017. <https://doi.org/10.1109/TIFS.2017.2710946>
9. Shor, P. W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, *SIAM J. Comput.*, vol. 26, no. 5, pp. 1484–1509, Oct. 1997. <https://doi.org/10.1137/S0097539795293172>
10. Nielsen, M. A. & Chuang, I. L. *Quantum Computation and Quantum Information*, 10th Anniversary ed. Cambridge Univ. Press, 2010. <https://doi.org/10.1017/CBO9780511976667>
11. Wootters, W. K. & Zurek, W. H. A single quantum cannot be cloned, *Nature*, vol. 299, pp. 802–803, Oct. 1982. <https://doi.org/10.1038/299802a0>
12. Ma, X., Yuan, X., Cao, Z., Qi, B. & Zhang, Z. Quantum random number generation, *npj Quantum Inf.*, vol. 2, Art. no. 16021, May 2016. <https://doi.org/10.1038/npjqi.2016.21>
13. Barenco, A. et al. Elementary gates for quantum computation, *Phys. Rev. A*, vol. 52, no. 5, pp. 3457–3467, Nov. 1995. <https://doi.org/10.1103/PhysRevA.52.3457>
14. Goodfellow, I. et al. Generative adversarial nets, in *Proc. Adv. Neural Inf. Process. Syst.*, pp. 2672–2680, 2014. <https://arxiv.org/abs/1406.2661>
15. Ho, J., Jain, A. & Abbeel, P. Denoising diffusion probabilistic models, in *Proc. Adv. Neural Inf. Process. Syst.*, pp. 6840–6851, 2020. <https://arxiv.org/abs/2006.11239>
16. Fridrich, J., Goljan, M. & Du, R. Detecting LSB steganography in color and gray-scale images, *IEEE Multimedia*, vol. 8, no. 4, pp. 22–28, Oct.–Dec. 2001. <https://doi.org/10.1109/93.959097>
17. Ker, A. D. Improved detection of LSB steganography in grayscale images, in *Proc. Int. Workshop Inf. Hiding*, pp. 97–115, 2004. https://doi.org/10.1007/978-3-540-30114-1_8
18. Sallee, P. Model-based steganography, in *Proc. Int. Workshop Digital Watermarking*, pp. 154–167, 2003. https://doi.org/10.1007/978-3-540-24624-4_12
19. Raja, K. B., Chowdary, C. R., Venugopal, K. R. & Patnaik, L. M. A secure image steganography using LSB, DCT and compression techniques on raw images, in *Proc. 3rd Int. Conf. Intell. Sensing Inf. Process.*, pp. 170–176, 2005. <https://doi.org/10.1109/ICISIP.2005.1619442>
20. Boroumand, M., Chen, M. & Fridrich, J. Deep residual network for steganalysis of digital images, *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 5, pp. 1181–1193, May 2019. <https://doi.org/10.1109/TIFS.2018.2871749>
21. Bennett, C. H. & Brassard, G. Quantum cryptography: Public key distribution and coin tossing, in *Proc. IEEE Int. Conf. Comput., Syst. Signal Process.*, pp. 175–179, 1984. <https://arxiv.org/abs/2003.06557>
22. Lo, H. K., Curty, M. & Tamaki, K. Secure quantum key distribution, *Nature Photon.*, vol. 8, pp. 595–604, Jul. 2014. <https://doi.org/10.1038/nphoton.2014.149>
23. Herrero-Collantes, M. & Garcia-Escartin, J. C. Quantum random number generators, *Rev. Mod. Phys.*, vol. 89, no. 1, Art. no. 015004, Feb. 2017. <https://doi.org/10.1103/RevModPhys.89.015004>
24. Karras, T., Laine, S. & Aila, T. A style-based generator architecture for generative adversarial networks, in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 4401–4410, 2019. <https://doi.org/10.1109/CVPR.2019.00453>

25. Rombach, R., Blattmann, A., Lorenz, D., Esser, P. & Ommer, B. High-resolution image synthesis with latent diffusion models, in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., pp. 10684–10695, 2022.
<https://arxiv.org/abs/2112.10752>
26. Ramesh, A. et al. Hierarchical text-conditional image generation with CLIP latents, arXiv:2204.06125, Apr. 2022. <https://arxiv.org/abs/2204.06125>
27. Abraham, H. et al. Qiskit: An open-source framework for quantum computing, Zenodo, 2019.
<https://doi.org/10.5281/zenodo.2573505>
28. Zhang, T. AI-Generated Images vs Real Images, Kaggle, 2023.
<https://www.kaggle.com/datasets/tristanzhang32/ai-generated-images-vs-real-images>
29. Asha D., Manisha M., Mamta T., Renuka C., Pramila S.: Hybrid Quantum-Inspired Phase–Amplitude Frequency Domain Model for Robust Audio-in-Image Steganography. *J. Intelligent Decision Making and Information Science*. Vol.3. No.12S (2026), <https://jidmis.org/index.php/jidmis/article/view/3346>