



Encoding and Decoding of Noise Images after Enhancement

Ayat Ali Abd-Allah¹, Atheel Nowfal Alkhayyat², Lara Mwafaq³, and Omer Abdulhaleem Naser⁴

¹ Department of Unmanned of Aerial Vehicle Engineering, College of Engineering, Al-Nahrain University, Baghdad, Iraq

^{2,3} Department of Computer Engineering, College of Engineering, Al-Nahrain University, Baghdad, Iraq

⁴ University of Information Technology and Communications, Baghdad, Iraq

• These authors contributed equally to this work

DOI: <https://www.doi.org/10.29194/NJES.29030442>

Received: Feb. 26, 2025

Revised: May 9, 2025

Accepted: Oct. 6, 2025

Published: Sep. 20, 2026

Abstract

This paper tests the effect of noise on digital images. Since Gaussian noise can affect almost all computer and communication systems and therefore can come from various natural sources, it was interesting to measure the effect of such noise on features of the image. To generate the noisy images, a few images were used with Gaussian noise, and a data set was applied to this purpose. The wide variety of noise parameters was measured using the same input image with different mean and variance values each time. Gaussian noise was then removed with enhancement filters. In all fields of online communication, multimedia systems, medical imaging, and military communications, digital images have important roles to play. Internet and cellular networks are storing and delivering colored images. A lot of data needs to be protected from unwanted access. In cryptography, messages are encoded to render them unintelligible. Security of data is always essential. There are different cryptographic algorithms for encrypting and decrypting images. Our purpose of this study is to encrypt and decrypt the images with the help of the Advanced Encryption Standard (AES) algorithm for authentication. This paper guarantees that after processing, the quality of the image is maintained and no corruption was occurred

Keywords: *Encryption, Decryption, AES, Noise, Image, Gaussian*

Corresponding author: Provide the corresponding author information and publisher here. E-mail address: ayata8179@gmail.com

1. Introduction

There are numerous commonly accessible encryption methods that are employed in information security. Both symmetric (private) and asymmetric (public) key encryption fall under this category. With symmetric key encryption, also known as secret key encryption, data is encrypted and decrypted using a single key. Before the key is transmitted between entities, it should be dispersed. Keys have a crucial function. If the algorithm uses a weak key, then anyone can decrypt the data. The size of the key used determines the symmetric key encryption's strength. It is more difficult to break encryption using a longer key for the same method than one employing smaller key. Many cryptographic techniques, such as RC2, DES, 3DES, RC6, Blowfish, and AES, have both strong and weak keys. RC2 utilizes a single 64-bit key. DES utilizes a single 64-bit key. Triple DES (3DES) employs three 64-bit keys, whereas AES uses a variety of keys totaling 128, 192, or 256 bits. Blowfish employs various keys (32-448); the default is 128 bits, whereas RC6 uses a variety of keys (128, 192, 256)[1]. Brief definitions of the most common encryption techniques are provided as follows: NIST (National Institute of Standards and Technology) first approved the Data Encryption Standard (DES). DES is a 64-bit key with a 64-bit block size. Numerous attacks and approaches have shown DES's flaws, rendering it an unsafe block cipher [2,3]. 3DES is an improved

version of DES, with a 64-bit block size and 192-bit key size. This standard's encryption mechanism is similar to that of the original DES, but it is applied three times to enhance the encryption level and average safe time. 3DES is reported to be slower than other block cipher algorithms [2]. RC2 is a block cipher with a 64-bit block cipher and flexible key sizes ranging from 8 to 128 bits. RC2 is vulnerable to related-key attacks with 234 selected plaintexts [2]. Blowfish is a block cipher with 64-bit blocks that can be used to replace the DES algorithm. It accepts a variable-length key ranging from 32 to 448 bits; the default is 128 bits. Blowfish is not patented, requires no permission, and is freely available for all usage. Blowfish has versions with 14 rounds or fewer. Blowfish is the successor to Twofish [4]. AES is a block cipher. It features a changeable key length of 128, 192, or 256 bits; the default is 256. It encrypts 128-bit data blocks in 10, 12, or 14 rounds, depending on the key size. AES encryption is quick and adaptable; it may be used on a variety of platforms, particularly tiny devices [5]. AES has been extensively tested for various security applications [2,6]. RC6 is a block encryption derived from RC5. It was created to meet the requirements of the Advanced Encryption Standard competition. RC6 proper has a block size of 128 bits and supports key sizes of 128, 192, and 256 bits. Some According to [7], RC6 is an Advanced Encryption Standard. Recent developments in AES in order to

address new security needs like side-channel threats, low-power device lightweight solutions, and integration with cutting-edge trends like edge computing and quantum-secure techniques. For instance, Zhang et al. [8] created an improved AES technique that is low complexity, significantly reduces computational overhead, and maintains security for IoT systems. Similarly, Li et al. [9] presented a lightweight encryption technique for cloud-edge IoT systems based on AES that is low latency and energy efficient. Additionally, it has been demonstrated that AES with chaotic systems and noise models is a more efficient way to make the ciphertext less recognizable and more cryptographically secure. To raise the entropy of the encrypted image, for instance, Jassim [10] created a variation of AES based on chaos. Alzain et al. [11] investigated quantum-inspired methods to strengthen AES against new quantum attacks. These developments show that AES is still relevant and useful in contemporary cryptography research. Investigating recent developments in AES-based encryption and decryption systems, assessing their advantages and disadvantages, and suggesting enhancements to meet contemporary security and performance requirements are the objectives of this study. Image enhancement is a process where image quality or usability is improved. Image encryption means that the image is rendered unreadable to unauthorized users. By image decryption, an image is brought back to its original state and becomes accessible to those authorized to see it. In applications such as secure image transmission over public networks, medical imaging, and remote surveillance, issues of data confidentiality and integrity require these techniques. using improvement, encryption, and decryption, we balance security and quality with an end result. We have a few ways to check image quality, such as approaches like Peak Signal to Noise Ratio (PSNR) and mean square error (MSE). Image quality was evaluated in this research using PSNR and MSE.

2. Noise Generator

Noise is an undesirable signal used to degrade the image's original quality. Some noises, such as Gaussian noise, salt and pepper noise, quantization, Rayleigh noise, Gama noise, brown noise, speckle noise, and much fundamental noise, degrade the image. The main sources of noise in digital images are as follows: 1) At the moment of image acquisition or transmission. 2) acquiring the light level, 3) environmental conditions during image acquisition, 4) electronic image transmission, and 5) dust particles on the scanner screen [12].

2.1 Gaussian Noise

Thermal noise is modeled through the Gaussian noise model. The univariate Gaussian noise n has density eq. (1) [12].

$$p_n(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, -\infty < x < +\infty \quad (1)$$

In other words, μ is the mean value, and σ^2 is the variance. x is the gray level of a digital image; we usually say $x \in [0, 255]$ [12].

3. Related work

This section examines the results gathered from different resources in order to provide a more comprehensive view of the performance of the compared algorithms. investigated [13] the energy consumption of many standard symmetric key encryptions on mobile devices. After only 600 encryptions of a 5 MB file using

Triple-DES, the remaining battery power is 45%, and additional encryptions are impossible due to the battery's rapid depletion. determined [14] that AES is faster and more efficient than alternative encryption methods. When the transmission of data is considered, there is no substantial variation in the performance of different symmetric key schemes (most of the resources are needed for data transmission rather than computation). Even during data transport, it is recommended to employ AES if the encrypted data is stored at the opposite end and decoded several times. Increasing the AES key size to 64 bits results in an 8% increase in energy consumption without any data transfer. The distinction is not discernible. Reducing the number of rounds saves electricity, but it renders AES insecure and should be avoided. Seven or more rounds are generally regarded as safe and can save energy in some scenarios. [15] describes a modified AES-based technique for image encryption. Encryption is the most common method of achieving image security using technology. Image and video applications are quite diverse, spanning from internet communication to multimedia systems, medical images, telemedicine, and even military communication. Vector quantization is one of the various image protection methods available. Vectorizing an image can be done in a variety of methods, including breaking it into vectors and encoding and decoding each vector individually. Or you can break the image in the desired shape into millions of shadows, making it inaccessible to unauthorized users.

Secure image encryption using AES has been the subject discussed in [16]. This time the world is the most concerned about the security issue; the most important thing is to protect the confidentiality of transmission of the image from unauthorized access. Thus, it becomes difficult to ensure the security of an individual. Different protective measures for data from unauthorized users are available. For the image's encryption and decryption, the AES is used to change the image into a non-recognizable image form and again to restore the image to the original image through an authorized receiver.

The authors in [17] demonstrate image encryption and decryption via the use of the AES algorithm. The AES algorithm is used for encryption and the decryption of the image for effective security of the image transmission. Because AES offers greater security, AES has replaced the Data Encryption Standard (DES). A 128-bit key is used for encryption in AES key expansion through bitwise exclusive or image-set pixel operations.

For decrypting and encrypting data, we [18] researched which encryption algorithms may provide the most performance. This was done using four standard algorithms: Blowfish, AES, DES, and 3DES. To analyze these methods, the contents and sizes of the encrypting input files were altered, and two distinct platforms were utilized to test them: P-II 266 MHz and P-4 2.4 GHz. The results illustrate that Blowfish can offer the best performance of the other algorithms, and AES outperforms 3DES and DES. 3DES 1/3 throughputs are also provided for DES.

The performance of symmetric encryption techniques is examined. This study [19] investigated many different standard algorithms: AES, DES, Blowfish, and RC6. We applied different variables to these algorithms, with the kind of data used, different data block sizes, key sizes, power consumed by the battery, and encryption and

decryption speed data. The technology presented in this article [20] uses simple operations to alter an image's binary matrix. The required steps include extending the 8-column matrix to 64 columns, restructuring it into 64 columns, dividing it into four blocks, and shuffling the columns with secret index keys. These keys are generated from four sets of common chaotic logistic parameters. Each set uses a chaotic logistic map model to generate a chaotic key, which is subsequently converted to an index key. This index key shuffles columns during encryption and reverses during decryption. The cryptographic technique guarantees a vast key space that can withstand hacking. This work [21] describes a strong and adaptable visually safe image encryption algorithm. First, a new compression and reconstruction network is developed using meta-learning compressed sensing (Meta-learning CS) with the goal of compressing the plain image. The compressed image is then encrypted using a 2D-SLC hyperchaotic map, which produces a noise-like secret image. Finally, a visually safe cipher image is obtained utilizing a newly created embedding and separation network based on traditional methodology.

4. Materials and Methods

The input image and several image-enhancing techniques with the AES algorithm are used to perform the encryption and decryption processes of this experiment. The quality of the image is determined by MSE and PSNR, as well as the time taken (in seconds).

4.1 Test Material

The input image set as shown in figure 1.



Figure 1. Input image [22]

4.2 Cryptography system

Within many cryptographic approaches, it is primarily used to protect data integrity and data security. There are basically two cryptographic systems depending on the use of keys. They are symmetric key encryption (use the same key) and asymmetric key cryptography (use different keys) [23].

4.2.1 Components of Cryptosystem

The components of a cryptosystem are [23]:

- Plaintext: Something that the algorithm can read in an intelligible format that acts as its input, presumably a message or a piece of data.
- Ciphertext: This encrypted message is created as an unreadable output. It requires plaintext and the key.
- Encryption Algorithm: An encryption algorithm modifies plaintext (readable communication) to produce ciphertext.

- Decryption algorithm: It takes the secret key and ciphertext as inputs and outputs plaintext. In general, it reverses the encryption process.

- Public and Private Keys: A cryptosystem is either asymmetric (private key and public key) or symmetric (two keys) in nature, and both the public key (known by all) and the private key (known only by the user).

4.2.2.AES Algorithm

The Advanced Encryption Standard (AES) development initiative was initiated in 1997 by the National Institute of Standards and Technology (NIST) [24], which also invited the public to contribute potential block cipher algorithms. Many cryptographic services, particularly those that guarantee data confidentiality, are based on block ciphers. Three Rijndael instantiations are announced by this standard: AES-128, AES-192, and AES-256, where the suffix specifies the key's bit length. The block size (the length of the data inputs and outputs) is 128 bits in both cases. Rijndael allows additional block sizes and key lengths that are not adopted by this standard. A comparison between DES and AES is shown in Table 1 [25].

Table 1. The distinction between DES and AES

Factors	DES	AES
Length of a key	56 bits	128, 192, or 256 bits
Size of a block	64 bits	128, 192, or 256 bits
Cipher type	Symmetric block cipher	Symmetric block cipher
Year Developed	Introduced in 1977	Introduced in 2000
Security	No longer secure	Currently considered secure
Number of Keys	2^{56}	$2^{128}, 2^{192}, 2^{256}$

4.3 Data Flow Diagrams

The image to be sent is encrypted using the AES technique. The steps for performing AES are as follows [24]:

Step1: SUBBYTES () applies a substitution table (S-box) to each byte.

Step2: SHIFTRROWS () shifts rows of the state array by different offsets.

Step3: MIXCOLUMNS () mixes the data within each column of the state array.

Step 4: ADDROUNDKEY () combines a round key with the state. AES supports three key lengths: 128 bits, 192 bits, and 256 bits. This algorithm begins with the key and image are encrypted with a random number before being scrambled through four mathematical operations. The key used to encrypt the communication must also be used to decrypt it, as shown in figure (2).

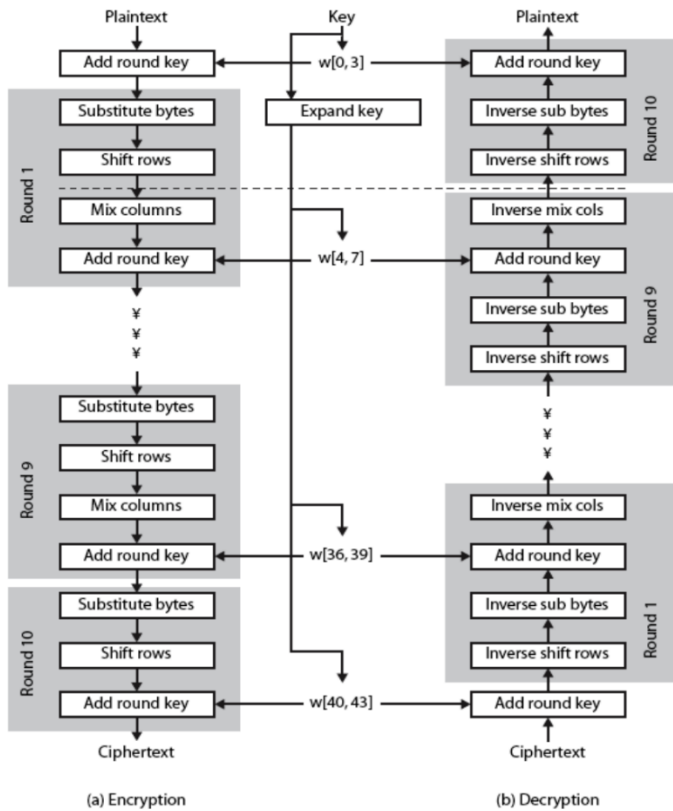


Figure 2. AES algorithm [24]

4.4 Encryption process

Encryption is a popular technique that plays a major role to protect data from intruders. AES algorithm uses a particular structure to encrypt data to provide the best security. To do that it relies on a number of rounds and inside each round comprise of four sub-processes. Each round consists of the following four steps to encrypt 128-bit block.

A. Substitute Bytes Transformation

The first stage of each round starts with Sub Bytes transformation. This stage is depending on nonlinear S-box to substitute a byte in the state to another byte. According to diffusion and confusion Shannon's principles for cryptographic algorithm design it has important roles to obtain much more security [12]. For example, in AES if we have hexa 53 in the state, it has to replace to hexa ED. ED created from the intersection of 5 and 3. For remaining bytes of the state have to perform these operations. This is shown in Table 2 and figure 3.

Table 2. Sbox(): substitution values for the byte (in hexadecimal format) [25]

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	08	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

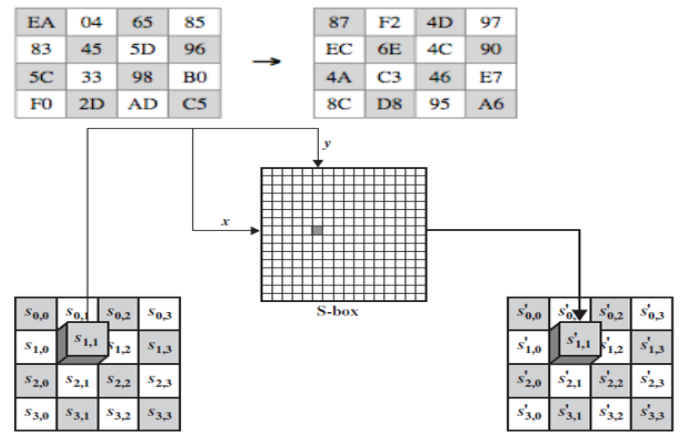


Figure 3. Illustration of sub byte [25]

B. Shift Rows

After sub byte, the next step in the state processing chain is shifting rows. This step is basically about moving each row's bytes cyclically to the left instead of to row zero. This technique does not permute the bytes in row 0. Only one byte is circularly shifted one bit to the left in the first row.

Two bytes are moved to the left of the second row. However, in the final row, the final row is moved three bytes to the left [26]. The size of the new state is not changed and continues to be the same original size of 16 bytes, but the position of the bytes in the state is shifted, as shown in figure 4.

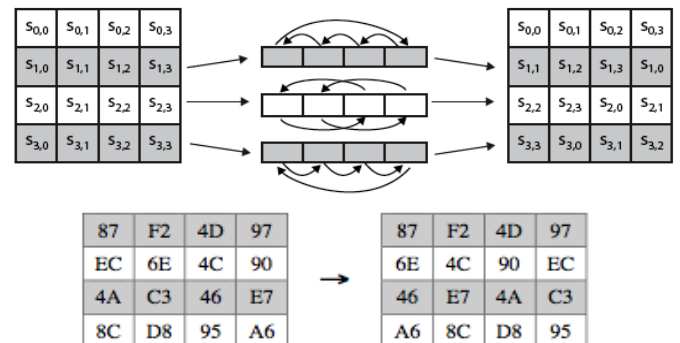


Figure 4. Shift row [25]

C. Mix Columns

There is another important phase in the state available—Mix Column. The multiplication is carried out outside of the state. Taking an example, in matrix transformation, each byte of one row is multiplied by the value (byte) of the state column. What this means is that every column of state must be multiplied by every row in the matrix transformation. These multiplication outputs are then XOR'ed together to give a new 4-byte sequence to send to the next stage. In this phase, the size of the state does not change; even the state is kept at its original 4x4 in figure 5[15].

$b_1 = (b_1 * 2) \text{ XOR } (b_2 * 3) \text{ XOR } (b_3 * 1) \text{ XOR } (b_4 * 1)$. All columns of the state are exhausted and so on [23].

D. Add Round Key

The Add Round Key is the most important stage for the AES algorithm. A 4x4 byte matrix [23] stores the key and the input data called the state. The spread of the 128-bit key and input data over the byte matrices is illustrated in figure 7. Unlike Encrypt Key, Add Round Key can afford to be much more secure when it generates

encryption for the data. It is about the way of the relationship between the key and the cipher text. After the earlier stage, comes the cipher text. Similar to figure 8, user input resulted in the result depending on the key that was given to the users [23]. There is also the subkey paired in stage with the state. In Rijndael, there is a key schedule that is used on each round of the cipher, and part of the main keys are used to extract the subkey. It has the same state and subkey size. Each byte of the subkey is XORed bit for bit with the corresponding byte of the state [23]. As shown in figure 6. In figure 7 shows the process of encryption.

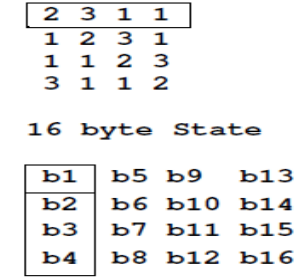


Figure 5. Mix columns [25]

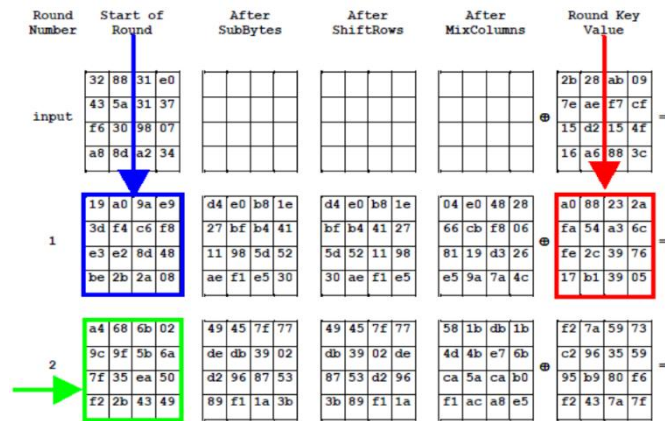


Figure 6. Round key [23]

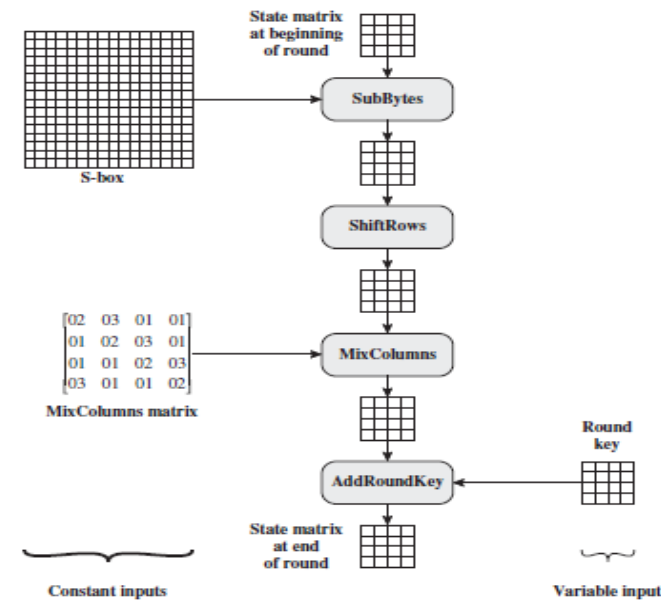


Figure 7. Encryption process [23]

4.5 AES Key expansion

AES key expansion is used to encrypt and decode data by the AES algorithm. This is one of the most vital steps of the AES structure.

There is a different key in each round. The AES Key Expansion mechanism is the main topic of this section. The round keys are generated word per word using a word defined as an array of 4 bytes, using the key expansion routine. It will return $4 \times (Nr+1)$ words. As shown in figure 8 [23], the

number of rounds is represented by Nr . The steps are as follows: Using the cipher key (or starting key), we use the first four words. figure 12 shows that the size of the key is 16 bytes (k_0 to k_{15}), which is an array. w_0 is expressed through the first four bytes (k_0 to k_3), w_1 is expressed in the first four bytes (k_4 to k_7), and so forth. With this equation, we may easily compute and find keys in each round.

$$K[n]: w[i] = k[n-1]: w[i] \text{ XOR } k[n]: w[i]$$

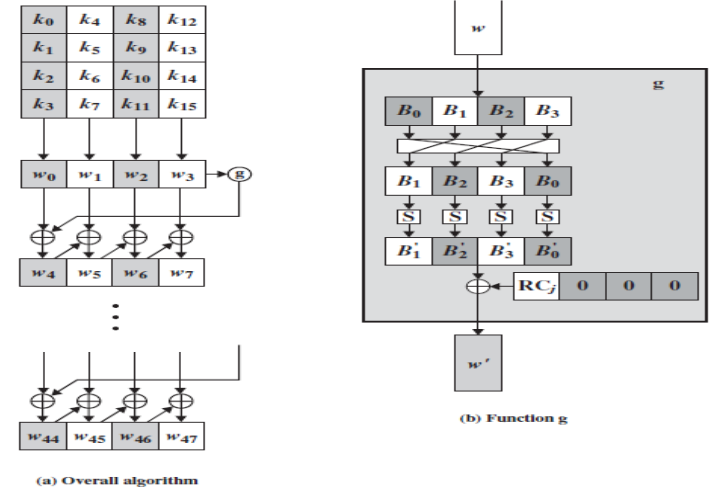


Figure 8. Key expansion [23]

4.6 Decryption process

The process by which the data is given a form that renders it illegible to any person other than the intended recipient is encryption. This is based on the key we get from the data sender. The AES decryption process is the same as that of the encryption process in a reverse manner in which both transmitter and receiver perform the AES decryption process with the same key to encrypt and decrypt data. The final round of the decryption stage has three stages: InvShiftRows, InvSubBytes, and AddRoundKey, shown in figure 9.

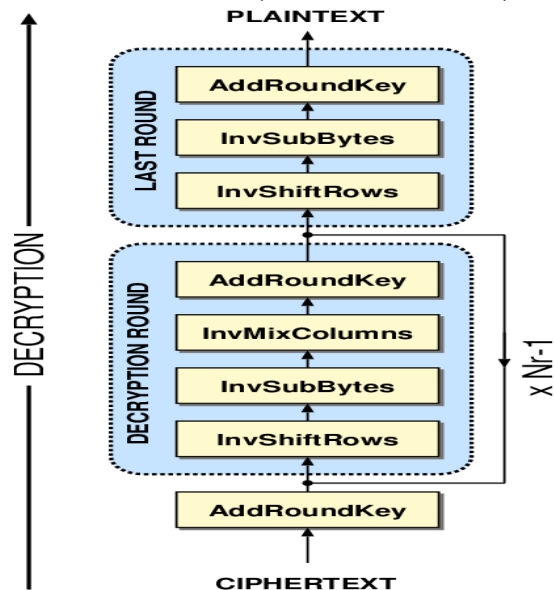


Figure 9. Decryption process [23]

5. Proposed algorithm

In the proposed algorithm we used many libraries for this work. These library cv2 , NumPy, PIL for import image , crypto. Cipher to import AES, crypto random to import get random bytes and import time to calculate time. Algorithm steps as follows:

Step 1: loading RGB image.

```
original_image = cv2.imread(path)
```

Step 2: adding the gaussian noise to original image.

```
noisy_image = add_noise(original_image)
```

Step 3: calculating MSE and PSNR between original and noise image.

```
mse_noisy = mse(original_image, noisy_image)
```

```
psnr_noisy = psnr(mse_noisy)
```

Step 4: for enhancing quality image we used specific library in python language.

```
def enhance_resolution(image):
```

```
    pil_image = Image.fromarray(cv2.cvtColor(image, cv2.COLOR_BGR2RGB))
```

```
    enhanced_image = pil_image.resize((image.shape[1] * 2, image.shape[0] * 2), Image.LANCZOS)
```

```
    return np.array(enhanced_image)
```

Step 5: after enhancing the image we used it to encrypt.

```
def aes_encrypt(image_data, key):
```

```
    cipher = AES.new(key, AES.MODE_CBC)
```

```
    encrypted = cipher.encrypt(pad(image_data, AES.block_size))
```

```
    return base64.b64encode(cipher.iv + encrypted)
```

Step 6: decrypting the output data from encryption process.

```
key = get_random_bytes(16)
```

```
def aes_decrypt(encrypted_data, key):
```

```
    encrypted_data = base64.b64decode(encrypted_data)
```

```
    iv = encrypted_data[:16]
```

```
    cipher = AES.new(key, AES.MODE_CBC, iv)
```

```
    decrypted = unpad(cipher.decrypt(encrypted_data[16:]), AES.block_size)
```

```
    return decrypted
```

Step 7: culauling time from start encryption to decryption process.

Step 8: displaying encryption image as data.

```
print("Encrypted image data in base64:")
```

Step 9: displaying decryption image as image.

```
decrypted_image = np.frombuffer(decrypted_data, dtype=np.uint8).reshape(enhanced_image.shape)
```

Step 10: culcating MSE and PSNR for decription image.

```
mse_decrypted = mse(enhanced_image, decrypted_image)
```

```
psnr_decrypted = psnr(mse_decrypted)
```

Step 11: saving the noise image, enhance image and decrypting image.

The flowchart of proposed algorithm shown as in Figure 10.

The flowchart of proposed algorithm shown as in Figure 10.

6. Parameter analysis

Different analyses parameters were used to measure the result subjectively by show eyes, MES, PSNR, and time in seconds.

The average square error difference between the original image and enhanced image is called the Mean

Square Error (MSE). Various ways in which the difference in the values implied by an estimate and the actual quality. calculated in the equation below, where Mis the width of the picture and N is the

height. Image A (i, j) is the original, and image B(i, j) is the enhanced. Eq. (2)[27]: Here are iand j for row-column pixels of both original and optimum images.

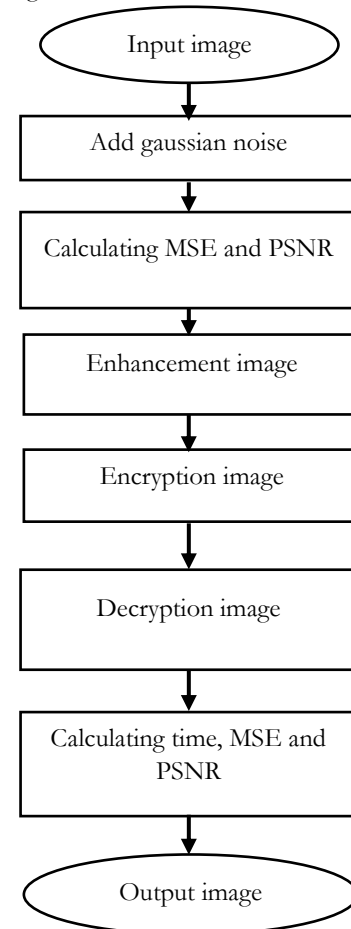


Figure-10. flowchart of proposed algorithm

$$MSE = \frac{1}{M * N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [B(i, j) - A(i, j)]^2 \quad (2)$$

The Peak Signal Noise Ratio (PSNR) has an effect on image representation, which is a ratio between maximum available power and corrupting noise. PSNR is a term used in engineering that provides a comparison of corrupting noise power to the code representation of the signal that allows the best representation. Normally, PSNR is stated in decibels. PSNR is usually stated in decibels of the logarithmic

scale because so many signals have such a very large dynamic range. Visual quality and (PSNR) are then used to evaluate the proposed algorithms. As shown in eq. (3)[27], normally, the PSNR is used as a measure of the image reconstruction quality.

$$PSNR = 20 \log_{10} \left[\frac{(2^n - 1)}{\sqrt{MSE}} \right] [dB] \quad (3)$$

7. Results

In this paper, the execution was implemented using an Intel Core i7 (PyCharm 2020.3.4) using Python version 3.8.6. We load the input RGG image with size 73.9 KB. We used different types of noises; the image quality subjectively shows poor quality. We found after that the best adding noise is Gaussian noise, as shown in figure (11).

Following that, we compute MSE and PSNR between the noise and input images. As illustrated in tables 3, 4, and 5. The change of mean and sigma experimentally is many, and we found the best values are 25, 10, and 5. Following that, we apply the enhanced filter to all cases to adjust the mean and sigma, demonstrating better quality images both subjectively, as shown in figure (12), and objectively, as stated in tables 3, 4, and 5. Following that, the enhanced image was encrypted by the AES technique using a secret key. The image is converted into data known as ciphertext. This process takes only 0.013 seconds. This achievement is considered relatively fast as compared to work in [1] in terms of time consumption. To decode an image, we transform the ciphertext to the image using the same secret key and save the result as shown in the figure (13). For imaging quality objectively, we calculate MSE and PSNR for the decrypted image, and we found the best result after finishing the decryption process. The image quality stays the same and never changes during the encryption and decryption processes, as shown in tables 3, 4, and 5.



Figure 11. imaging Gaussian noise



Figure 12. enhancement image



Figure 13. decryption image

Table 3. When mean = 0 and sigma = 25

Image	MSE	PSNR
Noise image	2.1616	94.782
Enhanced image	0.0	360
Decrypted image	0.0	360

Table 4. When mean = 0 and sigma = 10

Image	MSE	PSNR
Noise image	0.00019	85.128
Enhanced image	0.0	360
Decrypted image	0.0	360

Table 5. When mean 0 and sigma = 5

Image	MSE	PSNR
Noise image	4.9591	91.176
Enhanced image	0.0	360
Decrypted image	0.0	360

8. Conclusion

Noise has the potential to affect the image quality. Gaussian noise can be eliminated effectively through the use of enhancement filters. In this paper, the AES encryption and decryption can be applied to the enhanced images while maintaining their original quality. The result of this work subjectively shows better image quality and objectively uses MSE, PSNR, and time in seconds.

9. Reference

- [1] D. S. Abdul, H. M. Abdul Kader, and M. M. Hadhoud, "Performance evaluation of symmetric encryption algorithms," in Proc. 11th Int. Bus. Inf. Manag. Assoc. Conf. (IBIMA), 2009, pp. 1001-1006. <https://ibimapublishing.com/wp-content/uploads/articles/CIBIMA/2009/600033/600033>
- [2] D. L. Evans, P. J. Bond, and K. H. Brown, "FIPS 197: Advanced encryption standard," Fed. Inf. Process. Stand. Publ., Nov. 2001; updated May 2023, pp. 91-126. <https://doi.org/10.6028/NIST.FIPS.197-upd1>
- [3] D. Coppersmith, "The Data Encryption Standard (DES) and its strength against attacks," IBM J. Res. Dev., vol. 38, no. 3, pp. 243-250, May 1994. <https://doi.org/10.1147/rd.383.0243>
- [4] B. Schneier, "The Blowfish encryption algorithm," Oct. 2008. <https://dl.acm.org/doi/10.5555/220626>
- [5] K. Naik and D. S. L. Wei, "Software implementation strategies for power-conscious systems," Mobile Netw. Appl., vol. 6, pp. 291-305, 2001. <https://doi.org/10.1023/A:1011487018981>
- [6] J. Daemen and V. Rijmen, "Rijndael: The Advanced Encryption Standard," Dr. Dobbs's J., pp. 137-139, Mar. 2001. <https://link.springer.com/book/10.1007/978-3-662-04722-4>
- [7] N. El-Fishawy, "Quality of encryption measurement of bitmap images with RC6, MRC6, and Rijndael block cipher algorithms," Int. J. Netw. Secur., pp. 241-251, Nov. 2007. <https://www.airitilibrary.com/Article/Detail/18163548-200711-201308060013-201308060013-241-251>
- [8] W. Zhang, Y. Li, X. Chen, and Y. Zhang, "An improved lightweight AES encryption algorithm for IoT devices," IEEE Access, vol. 9, pp. 76372-76382, 2021. <https://doi.org/10.1109/ACCESS.2021.3082166>
- [9] H. Li, L. Chen, and Y. Zhao, "Lightweight AES-based encryption scheme for secure data transmission in cloud-edge-IoT environment," Future Gener. Comput. Syst., vol. 108, pp. 135-143, 2020. <https://doi.org/10.1016/j.future.2020.02.036>
- [10] F. A. Jassim, "Chaotic key-dependent AES encryption technique for secure image communication," Int. J. Electr. Comput. Eng., vol. 12, no. 1, pp. 568-576, 2022. <https://doi.org/10.11591/ijece.v12i1.pp568-576>

- [11] M. A. Alzain, A. Khan, and A. W. Siddiqui, "Quantum-enhanced AES cryptosystem for secure communications," *IEEE Access*, vol. 11, pp. 20945-20953, 2023. <https://doi.org/10.1109/ACCESS.2022.3154321>
- [12] A. A. Al-Jaburi and A. H. Al-Sudani, "Hybrid techniques with support vector machine for improving artifact ultrasound images," *Iraqi J. Sci.*, vol. 64, no. 2, pp. 944-957, 2023. <https://doi.org/10.24996/ijis.2023.64.2.37>
- [13] P. Ruangchaiatupon and P. Krishnamurthy, "Encryption and power consumption in wireless LANs," in *Proc. 3rd IEEE Workshop Wireless LANs*, Newton, MA, Sep. 2001. <https://www.semanticscholar.org/paper/A-Comparative-Study-of-Symmetric-Key-Encryption-Jain-Bhatnagar/cfe429d943d07d58ab89250f10d89755ff60b1b9>
- [14] S. Hirani, "Energy consumption of encryption schemes in wireless devices," M.S. thesis, Univ. Pittsburgh, Apr. 2003. <https://d-scholarship.pitt.edu/collections/4a9c1f1a-b6b1-41ad-b342-2b5e53da784c>
- [15] H. Wang, "Corporate social responsibility in China," *Palgrave Handb. Corp. Soc. Responsib.*, pp. 1017-1040, 2021. https://doi.org/10.1007/978-3-030-42465-7_71
- [16] P. R., "Secure image encryption using AES," *Int. J. Res. Eng. Technol.*, vol. 1, no. 2, pp. 115-117, 2012. <https://doi.org/10.15623/ijret.2012.0102006>
- [17] C. Engineering, "Image encryption and decryption using AES," *Commun. Eng. Technol. (IJECET)*, vol. 6, no. 1, pp. 23-29, 2015.
- [18] H. Nadeem, "A performance comparison of data encryption algorithms," *IEEE Inf. Commun. Technol.*, pp. 84-89, 2006. <https://doi.org/10.1109/ICICT.2005.1598556>
- [19] S. M. M. H. Diaa, E. Hatem, and M. A. K., "Evaluating the performance of symmetric encryption algorithms," *Int. J. Netw. Secur.*, vol. 10, no. 3, pp. 213-219, 2010. <https://www.semanticscholar.org/paper/Evaluating-The-Performance-of-Symmetric-Encryption-Elminaam-Abdual-Kader/e2bc791344a938ead55f77a2b2fd43949dbedc6f>
- [20] M. Al-Laham, F. Omar, and Z. A. Alqadi, "Image binary matrix processing to encrypt-decrypt digital images," *Int. J. Adv. Comput. Sci. Appl.*, vol. 15, no. 3, 2024. <https://doi.org/10.14569/IJACSA.2024.01503121>
- [21] W. Ji, Y. Wang, and Ren, "Visually secure image encryption: Exploring deep learning for enhanced robustness and flexibility," *Expert Syst. Appl.*, vol. 266, Mar. 2025. <https://doi.org/10.1016/j.eswa.2024.126027>
- [22] J. Borovec and J. Kybic, "jSLIC: superpixels in image," in *Proc. Comput. Vis. Winter Workshop*, Mar. 2014, pp. 14-18. <https://core.ac.uk/download/pdf/47181098.pdf>
- [23] R. M., S. K. S., S. B. S., and S. T. J., "Digital image encryption and decryption based on RSA algorithm," *Int. J. Sci. Res. Sci. Eng. Technol.*, vol. 4099, pp. 168-173, 2022. <https://doi.org/10.32628/IJSRSET229431>
- [24] J. Nechvatal, E. Barker, L. Bassham, W. Burr, M. Dworkin, J. Foti, and E. Roback, "Report on the development of the Advanced Encryption Standard (AES)," *J. Res. NIST*, May 2001. <https://doi.org/10.6028/jres.106.023>
- [25] S. Bhandari, Z. K. Mundargi, and P. G. Student, "A review on image encryption and decryption," *Int. J. Creat. Res. Thoughts*, vol. 6, no. 2, pp. 2320-2882, 2018. <https://doi.org/IJCRT1813263>
- [26] A. M. Abdullah, "Advanced Encryption Standard (AES) algorithm to encrypt and decrypt data," *Cryptogr. Netw. Secur.*, vol. 16, no. 1, p. 11, 2017. https://www.researchgate.net/publication/317615794_Advanced_Encryption_Standard_AES_Algorithm_to_Encrypt_and_Decrypt_Data
- [27] D. Asamoah, E. Ofori, S. Opoku, and J. Danso, "Measuring the performance of image contrast enhancement technique," *Int. J. Comput. Appl.*, vol. 181, no. 22, pp. 6-13, 2018. <https://doi.org/10.5120/ijca2018917899>