sciendo

BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **23**, No 4
Sofia • 2023 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2023-0036

Hybridized Cryptographic Encryption and Decryption Using Advanced Encryption Standard and Data Encryption Standard

Avaneesh Kanshi[1], *Rajkumar Soundrapandiyan*[1], *V. S. Anita Sofia*[2], *Rajasekar V. R.*[3]

[1]*School of Computer Science and Engineering, Vellore Institute of Technology, India*
[2]*Department of Computer Science, PSG College of Arts and Science, India*
[3]*University of Technology and Applied Sciences, Oman*
E-mails: avaneeshkanshi5@gmail.com rajkumarsrajkumar@gmail.com sofia.joel@gmail.com rajasekar.sur@cas.edu.om

***Abstract*:** *This research proposes an efficient hybridized approach for symmetrical encryption of image files in bitmap formats. Due to the heavy use of lightweight encryption in fields such as military and corporate workplaces, intruders try to intercept communication through illegal means and gain access to classified information. This can result in heavy losses if the leaked image data is misused. The proposed enhances the security and efficiency of one of the most used standard symmetric algorithms, Advanced Encryption Standard* (*AES*). *In the proposed method, the AES architecture has been modified using a less intensive algorithm, Data Encryption Standard* (*DES*). *DES carries a sub-process of permuting data columns rather than the AES's mixing feature. The proposed algorithm is analyzed using a set of 16 bitmap images of varying memory sizes and resolutions. The effectiveness of the algorithm is evaluated solely in terms of perceptual invisibility as per the main objective of the research.*

***Keywords*:** *Symmetric encryption, Advanced encryption standard, Data encryption standard, Symmetric key, Hybridization.*

1. Introduction

The development of network systems has increased the dependency on information exchange a lot. However, at the same time, it also brought about a plethora of new issues and concerns, chief among them being the need to protect data and resources from disclosure during storage as well as transmission, guaranteeing the authenticity of data and messages, and protecting systems from network-based attacks. To safeguard against these issues, a variety of encryption algorithms have been proposed. The two main categories of encryption standards are considered namely, symmetric encryption and asymmetric encryption. Symmetric encryption involves the use of one key for both encryption and decryption. The plaintext is read into an encryption algorithm along with a key. The key works with the algorithm to turn the plaintext into ciphertext, thus encrypting the original sensitive data. This works well

for data that is being stored and needs to be decrypted later. The use of just one key for both encryption and decryption reveals an issue, as the compromise of the key would lead to a compromise of any data the key has encrypted. This also does not work for data-in-motion, which is where asymmetric encryption comes in. The beginning of asymmetric encryption involves the creation of a pair of keys, one of which is a public key, and the other – a private key. The public key is accessible by anyone, while the private key must be kept a secret from everyone but the creator of the key. This is because encryption occurs with the public key, while decryption occurs with the private key. The recipient of the sensitive data will provide the sender with their public key, which will be used to encrypt the data. This ensures that only the recipient can decrypt the data, with their own private key.

Symmetric encryption, with its use of a single key, is better used for data-at-rest. Data stored in databases needs to be encrypted to ensure it is not compromised or stolen. This data does not require two keys, just the one provided by symmetric encryption, as it only needs to be safe until it needs to be accessed in the future. For low-powered devices and for quick encryption, symmetric encryption algorithms are preferred over asymmetric ones as they provide efficient and effective encryption. A variety of symmetric algorithms exists, like the AES, DES, Blowfish, and Triple DES. With the limitations of DES's 56-bit key and the advent of faster computers, DES could no longer be considered a secure algorithm. Simple brute force attacks could crack into the DES algorithm in less than 10 hours rendering it to be a less secure and outdated algorithm over time. Moreover, these algorithms had to conform to several strict requirements:

- It must be a block cipher;
- Longer key length;
- Larger block size;
- Fast in computation and;
- Greater flexibility.

After several rounds of submissions and eliminations, the National Institute of Standards and Technology (NIST) narrowed the applicant pool down to five finalists out of which Advanced Encryption Standard (AES) algorithm. Although brute force attacks are proven to be ineffective against AES by NIST to date even though it is purely a symmetrical algorithm. However side-channel attacks make the cipher very vulnerable if situated in the same server. AES therefore is unable to handle algebraic and cryptanalysis-based attacks.

M a n i k a n d a n and R a j a l a k s h m i [1] propose a black box approach to generate plain text from a user-given input. A hierarchical model is deployed where the entire algorithm is divided into two phases. The first phase involves subjecting the plain text to perturbation, shifting, and swapping techniques. Perturbation involves adding noise. The text is then cleaved into two chunks. Each of these chunks is individually subjected to shifting and swapping. The second phase takes the modified string and it is encrypted with a user-selected key. A b d u l l a h [2] aims to study the importance of the Advanced Encryption Standard (AES) algorithm. The predecessors of AES are included but are not limited to implementing the encryption algorithm on handheld devices such as mobile phones and Drawbacks PDAs, where

incorporating such complex architecture on tightly bound hardware is surely tough. Another pitfall deals with the tradeoff between a better performance and a higher computationally intensive algorithm. Other challenges revolve around the uncertainty of tuning parameters such as the size of the key. With the increasing size of a key, the battery and time consumption also increases but it provided much more secure results. Akkar and Giraud [3] propose a new protection principle coined as transformed masking method. The current algorithm fixes this problem by processing all the bits at the end of every round. Hence, this results in the intruder only potentially obtaining the hamming weight but not anything else. Rihan, Khalid and Osman [4] evaluate two algorithms' AES and DES. These two algorithms are evaluated in terms of the computational power required, the CPU usage during runtime, and the security aspect. The results indicate that AES is relatively much faster than DES algorithm but at the same time, AES has taken up more than double the CPU usage as compared to the DES algorithm. Moharir and Suresh [5] employ their algorithm in various steps, the first step being the AES SubBytes transformation. This enables the algorithm to provide a non-linear substitution. The subsequent intermediate stages consist of eight or more rounds. Each round performs a DES cipher. After this stage, an EX-Or operation is performed over a series of 128 bits that are generated from the previous expansion stage. The final stage involves an Add round key step, which adds a round key to the state by employing a simple bitwise XOR operation. Despite providing good security, it is found to be more computationally intensive than both AES and DES. Ali [6] has proposed a hybrid encryption that uses the advantages of Blowfish algorithm to its advantage and then secures another layer by employing an Advanced Encryption Standard (AES) layer. However, it has been found that it requires a large processing time and poses a very complex architecture. Rehman et al. [7] propose an algorithm that combines two traditional algorithms to generate a much more secure algorithm to cater to all the needs. The two algorithms are Elliptic Curve Cryptography (ECC) and Advanced Encryption Standard (AES). This way, the algorithm always ensures data integrity and authentication. Due to its complexity, it leads to strenuous integration with existing software. Bansal and Jagdev [8] state that the communication of two nodes situated far away invites security issues like transferring of information from one node to another can be easily intercepted and interfered with by a third party. To counter these issues symmetrical cryptographic algorithms are used. These algorithms are the most advanced and adopted ones for performing cryptography in both hardware and software. No successful cryptanalytic attacks against AES have been discovered to date. The feature of the flexible key length of symmetric encryption allows a degree of future-proofing against exhaustive key attacks. Singh and Supriya [9] have found that AES algorithm is most efficient in terms of speed, time, throughput, and the avalanche effect. The security provided by these algorithms can be enhanced further if more than one algorithm is applied to data. Mondal and Maitra [10] have explored that in the fields of multimedia communication using images, videos, and audio, encryption of the data using AES algorithm directly leads to low security as the key values remain fixed. This paper proposes a modified version of AES algorithm by shifting the pixel position and randomizing the key value, but

is extremely time inefficient as it requires additional functions to achieve better the desired encryption.

Table 1. Summary of different symmetric encryption-decryption methods

Reference	Methods	Research gaps	Metrics
[10]	Incorporating AES standard into each festal round of DES	Has not been tested on large-scale data transfer	-
[11]	AES, DES, and 3DES	Overhead increases as the data size is increased. Compromise between key size and security	-
[12]	AES, DES, and RSA	Symmetric encryption algorithms lack scalability. Asymmetric encryption is more complex and poses high time complexity	-
[13]	AES-DES-RSA Hybridization	Encryption time higher than the individual algorithms	Avalanche effect, Encryption time, CPU usage, Throughput
[14]	The key size of AES increased to 320 bits	Computationally intensive as the size of input increases	Encryption time, Encryption throughput
[15]	Parallel encryption, Splitter function, Merging outputs of three different encryption techniques	Memory usage is very high as compared to AES, DES, and 3DES. High time and space complexity	-
[16]	Data encryption using AES/DES/RSA, Steganography using LSB substitution technique	A simple series-type encryption Can be easily decrypted by statistical attacks	Buffer size, PSNR, Time taken for encryption and decryption
[17]	Hybridization of Symmetric and Asymmetric encryption methods, AES, ECC, RSA	If two heavy algorithms are used, the system becomes computationally intensive. The amount of memory wasted in the process	Throughput, Encryption-decryption time, Algorithm complexity
[18]	Hybrid cryptography using a cross-relation between encryption algorithms and hashing functions, Key encryption	The complexity of the algorithm increases exponentially as more components are added for encryption	-

This paper presents an idea of integrating AES into the Feistel architecture of DES, embracing advantages from either of the constituent standards. This results in a much more efficient and crack-resistant hybrid encryption algorithm. The objectives of the proposed algorithm are as follows:

1. Randomness factor. The entropy of the cipher text is improved using a permutation function rather than a MixColumn, to improve the randomness of the bits encrypted.

2. Robustness. The proposed approach provides highly robust and secure encryption processes.

3. Time efficiency. Utilizing a comparatively less complex subprocess PermuteBytes instead of MixColumn improves the time taken to randomize the bits.

The rest of the paper is organized as follows: Section 2 presents the application scenario of the proposed method, Section 3 presents the proposed method, experimental analysis, and performance analysis is presented in Section 4. Section 5 concludes the paper.

2. Application scenario of the proposed method

The proposal focuses on presenting a lightweight encryption model while enhancing the encryption efficiency of the vanilla AES Algorithm. Looking at the application point of view, the proposed method can be employed in exchange for top-secret imagery in various fields explained as follows:

2.1. Military warfare

The military sector requires advancement in technological systems, especially in the field of cyber security. Due to increasing threats and the evolution of electronic warfare, military organizations around the world require secure communication channels to prevent data breaches that could potentially cause unwanted circumstances. The proposed methodology provides a good measure to assuage this, by providing speed and security, both being advantages.

2.2. Modern healthcare

In the increasing usage of electronic healthcare devices to help facilitate new treatments, imagery plays an important role in capturing data. These days, everything is connected to the internet, and transmission of healthcare data over the data may bring threats. The use of encryption-decryption techniques removes the possibility of data hampering quite a lot. Encrypting the pixels would prevent the attacker from interpreting the data as to him/her; it would seem a distorted image. Moreover, the lightweight feature of the proposed method would allow it to be incorporated into health devices that are mobile and have low power usage.

2.3. Satellite mapping

The use of satellite mapping is tremendous in amount. It is utilized in almost every GPS system, weather data, terrain mapping for architectural services, etc. The satellites capture the images and transmit the data using radio waves to stations on Earth. However, the acquisition of data always comes with a drawback, i.e., data manipulation. The streams of data that are transferred can be easily interfered with, as there is no secure transmission of radio waves. The proposed algorithm would convert the image data into an encrypted image that would be impossible to interpret.

3. Proposed approach

The skeletal view of the proposed symmetric encryption-decryption algorithm is portrayed in Fig. 1. The significant processes involved are as follows:

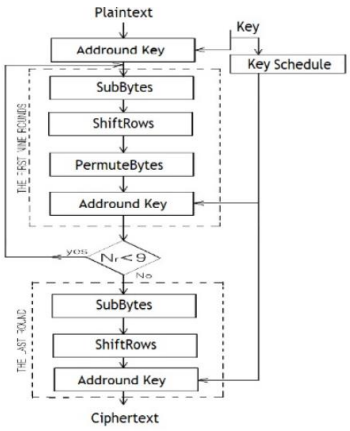


Fig. 1. Flow diagram of the modification proposal in AES encryption algorithm

3.1. SubBytes

This is the process of byte substitution where the byte encoded input are substituted using the Rijndael S-box table. This substitution is done in a way that a byte is never substituted by itself and not substituted by another byte, which is a compliment of the current byte. This outputs a 4×4 matrix:

$$\begin{bmatrix} b_0 & b_4 & b_8 & b_{12} \\ b_1 & b_5 & b_9 & b_{13} \\ b_2 & b_6 & b_{10} & b_{14} \\ b_3 & b_7 & b_{11} & b_{15} \end{bmatrix}.$$

3.2. ShiftRows

In this step, each row is shifted several times. However, the first row is left untouched. The second row is shifted once to the left, the third is shifted left twice and the last row is shifted thrice left. This results in the following 4×4 matrix:

$$\begin{bmatrix} b_0 & b_4 & b_8 & b_{12} \\ b_5 & b_9 & b_{13} & b_1 \\ b_{10} & b_{14} & b_2 & b_6 \\ b_{15} & b_3 & b_7 & b_{11} \end{bmatrix}.$$

3.3. PermuteBytes

This step has been derived from DES Algorithm. Using the output after ShiftRows process, the bytes are permuted. These permuted data are now divided into two halves and multiple rounds of permutation are carried out, similar to DES encryption method. Finally, the two halves are rejoined and a final permutation is performed on the combined block. This outputs a 128-bit stream.

3.4. AddRoundKey

The 64-bit stream of the previous stage is now XOR-ed with the current round's key. Instead of considering the matrix as 16 bytes, it outputs a data stream of 128 bits. The output of this round now acts like the input for the next round of the proposed algorithm.

4. Experimental results and performance analysis

4.1. Dataset

The proposed symmetric encryption algorithm has been evaluated using a dataset comprising 16 bitmap format images of varying resolutions.

The detailed description of these 16 images is given in Table 2. All these images are unique in terms of file size, resolution, and pixel density. Table 2 contains the information of the images such as the name of the image, resolution, size of the file in bytes, and number of pixels. In addition to this, image files have been depicted in Fig. 2.

Table 2. Dataset of 16 images

Index No	Image	File Size (kB)	Resolution	Number of pixels
1	Barbara_gray	258	512×512	262,144
2	beach	1100	750×500	37,500
3	bird	120	240×320	76,800
4	blackbuck	769	512×512	262,144
5	butterfly	10	275×183	50,325
6	dots	215	467×467	218,089
7	flower	136	600×600	360,000
8	land	770	1024×768	786,432
9	Mercury	1116	732×520	380,640
10	Mickey	403	346×396	137,016
11	monkey	193	584×447	261,048
12	peppers	1876	800×800	640,000
13	plane	42	427×427	182,329
14	sail	386	768×512	393,216
15	snail	193	256×256	65,536
16	Venus	45	300×150	45,000

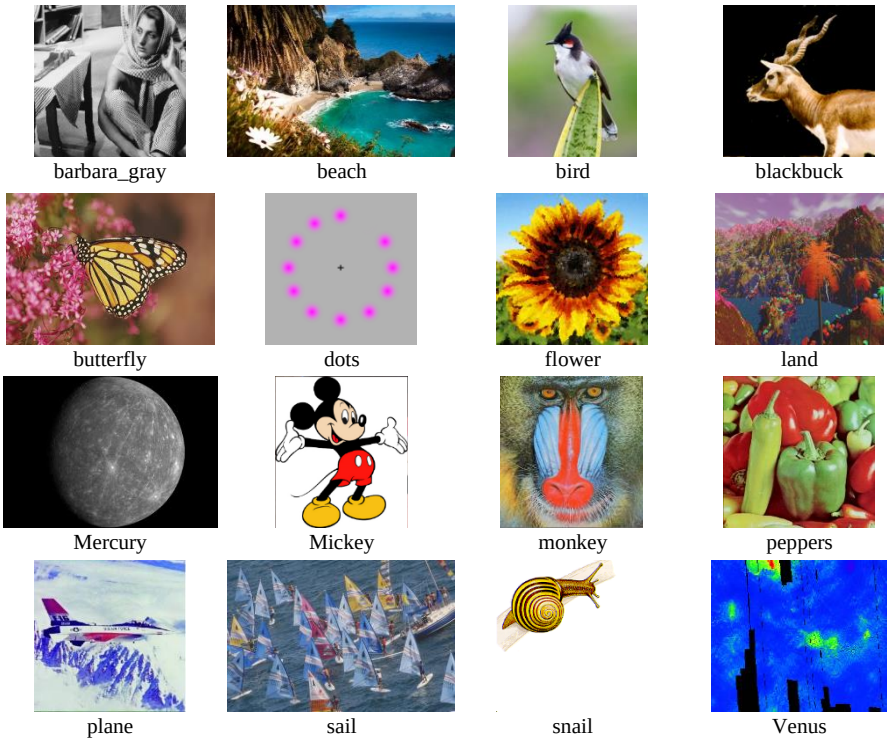


Fig 2. Illustration of the dataset

4.2. Performance evaluation

The performance of the symmetric encryption of images is mainly evaluated in the following categories: a) Perceptual Invisibility, b) Robustness, and c) Encryption efficiency. The evaluation criteria used to test perceptual invisibility are Peak Signal to Noise Ratio (PSNR), Mean Square Error (MSE), Structural Similarity Measure

Index (SSIM), Structural Content (SC), and Normalized Absolute Error (NAE). The robustness of the algorithm is evaluated using the following metrics, Avalanche Effect (AE) and Shannon's Entropy (SE). Finally, the efficiency of the algorithm is tested using the time taken to encrypt and decrypt and file size change from original to encrypted. Table 3 displays the accepted values for the metrics considered.

Table 3. Accepted values for evaluation metrics

Index No	Performance attributes	Metrics	Accepted range
1	Perceptual invisibility	PSNR	Above 30 dB
		MSE	Below 30 dB
		SC	Below 0.1
		NAE	0 to 0.1
2	Robustness	AE	0.95 to 1.0
		SE	Above 5
3	Encryption efficiency	Time taken	Lower the better
		File size change	Lower the better

This section discusses the various performance metrics used for evaluation in detail.

4.2.1. Perceptual invisibility

This metric helps to determine the structural correlation between the encrypted image and the original image. The lesser the similarity of the encrypted image to the original image, the better the encryption randomness, as it is a quantitative measure which informs how difficult is it to interpret the encrypted image. The visual interpretability is determined using the following metrics.

- Peak Signal to Noise Ratio (PSNR)

The PSNR statistically calculates the difference between the original and encrypted image in decibels (dB). It is calculated using the next equation,

$$\text{PSNR} = 10\log_{10}\frac{r^2}{\text{MSE}}, \tag{1}$$

where 'r' represents the maximum pixel value (i.e., 255) of the frame, MSE is the mean square error value. PSNR value below 30dB indicates that distortion is visible to the human eye. A higher value of PSNR specifies the higher quality reconstruction of the frame.

- Mean Square Error (MSE)

The dissimilarity between the original and encrypted image is measured by computing the mean of the squared error between them. It is calculated using the equation

$$\text{MSE} = \frac{1}{R \times C}\sum_{i=1}^{R}\sum_{j=1}^{C}\left(F_{\text{O}}(i,j) - F_{\text{S}}(i,j)\right)^2, \tag{2}$$

where: $R \times C$ is Resolution of Image; $F_{\text{O}}(i, j)$ is Original image's pixel intensity value at coordinate (i, j); $F_{\text{S}}(i, j)$ is Encrypted image's pixel intensity value at coordinate (i, j).

The higher value of MSE depicts better dissimilarity between the original and encrypted image.

- Structural Similarity Index Measure (SSIM)

The index is calculated using the luminance, contrast, and structure of the image. SSIM has been computed using the equation

$$\text{SSIM}\left(F_{\text{O}}, F_{\text{S}}\right) = \frac{\left(2\mu_{F_{\text{O}}}\mu_{F_{\text{S}}} + C_1\right)\left(2\sigma_{F_{\text{O}}F_{\text{S}}} + C_2\right)}{\left(\mu_{F_{\text{O}}}^2 + \mu_{F_{\text{S}}}^2 + C_1\right)\left(\sigma_{F_{\text{O}}}^2 + \sigma_{F_{\text{S}}}^2 + C_2\right)}, \tag{3}$$

where: $\mu_{F_{\text{O}}}, \mu_{F_{\text{S}}}$ are the mean values of the original and encrypted image; $\sigma_{F_{\text{O}}}$, $\sigma_{F_{\text{S}}}$ are the standard deviations of the original and encrypted image; $\sigma_{F_{\text{O}}F_{\text{S}}}$ is the cross-covariance between the original and encrypted image.

The variables $C_1=(K_1G)^2$ and $C_2=(K_2G)^2$, wherein G is the dynamic range of the pixel values in the frames F_{O} and F_{S}, by default K_1 and K_2 is considered as 0.01 and 0.03, respectively. The agreeable value of SSIM varies between 0 (No match) to 1 (Exact match).

- Structural Content (SC)

The structural content describes the arrangement of the pixels in an image, spatially. The higher the structural content, the lesser the quality of the encrypted image. The SC is evaluated using the next equation:

$$\text{SC} = \frac{\sum_{i=1}^{R}\sum_{j=1}^{C}\left(F_{\text{S}}(i,\,j)\right)^2}{\sum_{i=1}^{R}\sum_{j=1}^{C}\left(F_{\text{O}}(i,\,j)\right)^2}, \tag{4}$$

where $F_{\text{S}}(i, j)$ is the encrypted image and $F_{\text{O}}(i, j)$ is the original image.

- Normalized Absolute Error (NAE)

Normalized Absolute Error is used to determine estimated deviations between predicted and calculated values of original and encrypted images. A higher value of NAE shows that the encrypted image is tougher to decrypt or be interpreted by an attacker. It is calculated using

$$\text{NAE} = \frac{\sum_{i=1}^{R}\sum_{j=1}^{C}\left(\left|F_{\text{O}}(i,\,j) - F_{\text{S}}(i,\,j)\right|\right)}{\sum_{i=1}^{R}\sum_{j=1}^{C}\left(F_{\text{O}}(i,\,j)\right)}. \tag{5}$$

4.2.2. Robustness

The robustness of the encryption algorithm is determined by Avalanche effect and Shannon's entropy. Both of these metrics determine the randomness in encryption, which indicates the unpredictability of the original data from the ciphered data.

- Avalanche Effect (AE)

This is one of the most desirable properties of an encryption algorithm. It determines how significantly the encrypted file changes for a small change in key or original data. AE is evaluated using the next equation:

$$\text{AE} = \frac{\Delta\text{Cipher}}{\text{Cipher}}, \tag{6}$$

where Cipher is the number of bits in ciphertext.

- Shannon's Entropy (SE)

Shannon's entropy measures the unexpectedness of the bits of an encrypted image file. Higher entropy means a bigger and less predictable search space, increasing the randomness. This also results in fewer redundancies and increases difficulty in detecting correlations. The entropy has been calculated using the next equation:

$$\mathrm{SE} = -\sum_i^C P(i) \times \log_2 P(i), \tag{7}$$

where C ts distinct characters in the encrypted bitstream, and $P(i)$ is probability of occurrence of character i.

4.2.3. Encryption efficiency

This category defines how efficiently the algorithm encrypts a piece of data. An algorithm can be defined as efficient if it takes the least amount of time to encrypt the same set of data as compared to the existing algorithm and the change of file size from original to encrypted is to its minimum. The following metrics are computed to evaluate the efficiency of the proposed algorithm:

- Encryption-decryption time.
- Change in file size from original to encrypted.

4.3. Performance analysis

The performance of the proposed algorithm experimental results has been analyzed using previously mentioned analysis criteria, and compared against vanilla AES and DES algorithms.

4.3.1. Perceptual invisibility result analysis

Table 4 summarizes the evaluation and precisely quantifies the imperceptibility of the encrypted images. According to the table, the PSNR values are quite similar to each other, while being under 30 dB. This indicates that a significant amount of distortion is present and hence, the encrypted image cannot be physically interpreted to its original image.

Moreover, the MSE value identifies the average deviation of the encrypted image from the original image in terms of pixel intensity. The proposed methodology's MSE values are well above the 30 dB limit indicating huge variance in pixel intensity between the original and encrypted image. SSIM is another measure to check the similarity of two images, and the value lies in a spectrum from no match (0) to perfect match (1). As for the results, the obtained values of SSIM between encrypted and original images are close to 0, making the two images highly dissimilar. SC depicts the randomness of the pixel arrangement in an image, the higher the value, the higher the randomness of pixels.

From the results in Table 4, the structural content metric value of the proposed method is greater than 0.1, as same as the parent algorithms. This indicates high randomness in the pixel arrangement of the encrypted image, ensuring no pattern is followed in the encryption process, which can make it susceptible to being decrypted easily by attackers. Additionally, NAE is another metric to measure the variance

between two images. The higher the value, the higher the deviation of pixel intensity at a particular location of both images (0 indicates that both images are the same). The majority of images have an NAE of around 0.5, indicating a high deviation of the encrypted image's pixel intensity from the original image's pixel intensity.

Table 4. Perceptual invisibility quantitative evaluation results

Metric	Image	Data encryption standard	Advanced encryption standard	Proposed algorithm
PSNR	Barbara_gray	28.786	28.789	28.779
	beach	28.845	28.837	28.904
	bird	29.745	29.754	29.861
	blackbuck	27.515	27.282	27.635
	butterfly	29.896	29.912	29.801
	dots	29.604	29.657	29.237
	flower	28.932	28.922	28.930
	land	29.489	29.427	29.516
	Mercury	27.847	27.771	27.929
	Mickey	26.811	27.126	26.859
	monkey	30.046	30.063	30.100
	peppers	29.478	29.477	29.440
	plane	28.326	28.318	28.313
	sail	30.027	30.037	30.022
	snail	27.300	26.991	27.157
	Venus	28.662	28.767	29.084
MSE	Barbara_gray	7396.233	7384.391	7420.383
	beach	7196.901	7224.948	7003.616
	bird	4754.376	4736.375	4507.050
	blackbuck	13280.615	14784.257	12563.910
	butterfly	4436.267	4403.679	4634.836
	dots	5073.484	4951.872	6007.599
	flower	6914.581	6946.824	6920.132
	land	5350.067	5504.314	5283.810
	Mercury	11398.753	11803.639	10975.367
	Mickey	18366.394	15886.828	17964.719
	monkey	4138.672	4107.825	4037.291
	peppers	5376.305	5379.751	5473.047
	plane	9139.173	9172.774	9196.629
	sail	4175.494	4156.962	4186.044
	snail	14660.960	16905.268	15661.907
	Venus	7830.977	7459.692	6448.389
SSIM	Barbara_gray	0.015	0.017	0.017
	beach	0.017	0.018	0.017
	bird	0.020	0.023	0.019
	blackbuck	0.008	0.006	0.008
	butterfly	0.037	0.023	0.024
	dots	0.018	0.023	0.016
	flower	0.016	0.016	0.018
	land	0.018	0.019	0.020
	Mercury	0.007	0.010	0.012
	Mickey	0.014	0.014	0.014
	monkey	0.019	0.020	0.022
	peppers	0.020	0.020	0.020
	plane	0.014	0.007	0.015
	sail	0.017	0.019	0.018
	snail	0.031	0.020	0.021
	Venus	0.017	0.018	0.017

Table 4 (c o n t i n u e d)

Metric	Image	Data encryption standard	Advanced encryption standard	Proposed algorithm
SC	Barbara_gray	0.889	0.890	0.832
	beach	1.450	1.451	1.390
	bird	0.745	0.749	0.782
	blackbuck	1.960	2.237	1.768
	butterfly	1.478	1.467	1.521
	dots	0.660	0.599	0.615
	flower	0.963	0.959	0.972
	land	1.854	1.903	1.839
	Mercury	3.080	3.344	3.161
	Mickey	0.312	0.398	0.342
	monkey	1.020	1.022	0.952
	peppers	1.079	1.078	1.069
	plane	0.494	0.496	0.490
	sail	1.367	1.358	1.335
	snail	0.327	0.321	0.330
	Venus	3.862	3.709	3.300
NAE	Barbara_gray	0.560	0.559	0.584
	beach	0.548	0.549	0.552
	bird	0.439	0.437	0.415
	blackbuck	0.861	0.848	0.890
	butterfly	0.427	0.422	0.431
	dots	0.457	0.454	0.544
	flower	0.538	0.542	0.536
	land	0.476	0.476	0.474
	Mercury	0.765	0.721	0.711
	Mickey	1.113	0.892	1.050
	monkey	0.412	0.410	0.421
	peppers	0.467	0.467	0.475
	plane	0.644	0.643	0.648
	sail	0.412	0.413	0.419
	snail	0.852	0.932	0.879
	Venus	0.574	0.570	0.569

Table 5 represents the analysis of perceptual distortion of the proposed method in the form of histogram plots. These plots are the graphical bin representation of the pixel intensity (0-255) of a grayscale image. To evaluate the encryption power of the proposed algorithm, the histogram plot of the encrypted image is compared to the plot of the original image. In the table, the first column is the name of the image being compared, the second column represents the histogram plot of the original image and the third column represents the histogram plot of the encrypted image. The *y*-axis of the plot depicts the pixel count having a particular intensity, and the *x*-axis depicts the pixel intensity values in increasing order from 0-255. A heavy deviation exists in the histogram of the encrypted image from the original image. From Table 5, all the encrypted images have a normal curve meaning that the pixels are normally distributed throughout the spectrum. However, the plots for “blackbuck”, “dots”, “mercury”, “mickey”, “snail”, and “venus” have pixel count spikes in random intervals. This is a result because the original image has a large pixel count in few intensities.

Table 5. Robustness evaluation results

Image	Shannon's Entropy (SE)			Avalanche Effect (AE) (in %)		
	DES	AES	Proposed	DES	AES	Proposed
Barbara_gray	15.71	15.72	15.74	99.60	99.61	98.11
beach	16.45	16.46	17.39	26.71	99.61	98.60
bird	14.99	14.99	15.98	99.61	99.61	98.41
blackbuck	10.99	16.33	16.32	22.95	99.60	95.46
butterfly	11.48	11.48	11.51	99.62	99.65	98.20
dots	8.45	15.57	15.38	99.91	99.92	99.99
flower	15.13	15.12	15.14	99.59	99.61	98.41
land	16.32	16.34	16.31	99.61	99.61	98.32
Mercury	14.54	16.45	15.13	15.87	99.58	59.16
Mickey	13.29	13.32	13.33	97.23	99.65	99.21
monkey	15.46	15.46	16.47	99.62	99.59	98.42
peppers	16.58	16.58	16.92	89.86	99.65	68.27
plane	13.66	13.68	13.75	99.55	99.63	98.42
sail	16.00	15.99	16.21	99.61	99.60	98.01
snail	14.28	15.19	15.21	16.27	99.12	97.63
Venus	13.73	13.76	13.79	99.49	99.62	98.79

4.3.2. Robustness result analysis

Table 5 summarizes the local entropy and the avalanche effectiveness of each test case. By observing the data, the proposed algorithm holds the highest average entropy with an average of around 15.29 followed by AES and DES algorithm with entropy of 15.15 and 14.19, respectively. However, the proposed algorithm poses an Avalanche effect of 93.96% at an average lower than AES by approximately 5.1%.

The robustness of the algorithm determines how strong and how secure it is from various attacks. The key size also helps determine the same but for the purpose of this research, the key size is kept the same for all three algorithms. Moreover, both Shannon's entropy and Avalanche effect determine the degree of chaos of the encryption process.

The Shannon's entropy has been calculated with a radius of three, as a constant for all the test cases. In this analysis, the test cases were the encrypted files of the original images. As a higher entropy is an indicator of higher disorientation of the bits, the proposed method shows a lower frequency of a particular bit in the encryption byte stream as compared to the vanilla algorithms. Similar results have been observed for differed radii of local entropy, and it could be observed that from increasing radius, the entropy reduces, meaning that the occurrences of a particular pixel increase a local area. On the other hand, the Avalanche effect is another quantitative measure of the robustness of the encryption algorithm. It helps determine how vast the encryption set can be for a small input set. In other words, a small change in the input stream can stimulate a huge change in the encrypted stream or not. For a good encryption algorithm, the AE must remain above 50% for all cases. The test has been done by changing one random pixel RGB intensities and the encrypted image of both the original and modified image has been compared. With reference to Table 6, it can be inferred that both the proposed algorithm and AES have AE values above 50%, averaging around 98% to 99%. This means that one change in a bit changes

98-99% of the encrypted bits. However, DES has an AE of lower than 50% for a few test cases, making it a weaker algorithm than AES and the proposed method.

4.3.3. Encryption efficiency result analysis

Fig. 3 depicts the total time taken to encrypt and decrypt the image using AES, DES, and the proposed encryption algorithm. The time has been calculated by starting a system timer as soon as the first round of the encryption is initiated and stopping the same timer, after the last round of the decryption block. In general, the proposed algorithm takes the lowest amount of time (in seconds) to carry out the encryption and decryption process, followed by AES algorithm. As DES is computationally heavy, it takes the most time as compared to the other two algorithms. However, for very small image files, all three algorithms take almost the same time to encrypt and decrypt the image, but as the file size increases, the time difference between the three algorithms becomes significant. It can be easily noticed from Fig. 3 and Table 6, that as the file size increases, the time taken by DES algorithm increases exponentially. On the other hand, the total time taken by proposed and AES algorithms increases almost linearly.

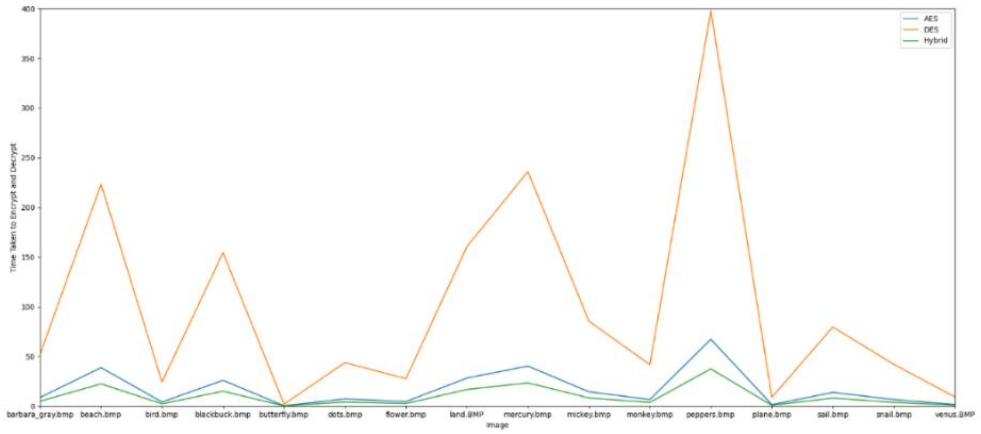


Fig. 3. Time taken to encrypt and decrypt

With respect to Table 6, an average of 32.44% increase in file size from original to encrypted can be observed. The increase in file size is mostly due to the padding of the original image file having an aspect ratio of 1:1. This padding is done to provide a perfect number of blocks for the encryption process. This padding process also helps prevent predictability attacks to find known image data, which can result in breaking the encryption.

From the table below, it can also be inferred that there is no loss during the encryption process, however, more data is appended (an average of around 35% increase in image size is observed) meaning that deciphering the encrypted image will provide unnecessary data to the attacker, potentially hiding the original plaintext. These values are quite similar to the ones observed during AES and DES encryption.

Table 6. Encryption efficiency evaluation results of proposed algorithm

Index No	Image	File size, kB		Total time (Encrypt and decrypt), s	Change in file size
		Original	Proposed method		
1	Barbara_gray	258	346	5.79	34.11%
2	beach	1,100	1469	21.83	33.55%
3	bird	120	161	2.36	34.17%
4	blackbuck	769	1027	15.02	33.55%
5	butterfly	10	14	0.21	40.00%
6	dots	215	288	4.19	33.95%
7	flower	136	182	2.66	33.82%
8	land	770	1027	14.92	33.38%
9	Mercury	1,116	1491	21.61	33.60%
10	Mickey	403	537	7.84	33.25%
11	monkey	193	259	3.72	34.20%
12	peppers	1,876	2502	36.45	33.37%
13	plane	42	58	0.83	38.10%
14	sail	386	516	7.47	33.68%
15	snail	193	257	3.78	33.16%
16	Venus	45	61	0.87	35.56%

5. Conclusion and future work

This research work proposes a lightweight and robust symmetrical encryption algorithm using an advanced encryption standard as a base and modifying its architecture using sub-processes of lesser complex encryption standard, DES. MixColumn process of the AES has proven to be the most extensive operation in the entire algorithm, in cases of plaintext data of 1 kB or above. In order to improve the randomizing operation of AES, the randomizing process of the less costly algorithm DES, PermuteBytes has been used in place of MixColumn. This increases the randomness of the encryption byte stream and reduces overhead computation costs by a huge margin. The proposed method has been then evaluated on a dataset comprising 16 bitmap images on the basis of perceptual invisibility, robustness, and efficiency and compared against AES and DES methods. The hybridized algorithm provides a substantial performance in terms of perceptual invisibility, and robustness and outperforms the existing algorithms in encryption efficiency. Hybridization of the AES with DES has proved to be a lightweight encryption algorithm that can be used in various applications that require low power capacity and need highly efficient encryption system to protect the data against attacks. In the future, the proposed image encryption approach can be researched and modified to handle video data and even sound data.

References

1. M a n i k a n d a n, P. H., K. H. R a j a l a k s h m i. A Hybrid Technique for Enhancing Data Security. – International Journal of Pure and Applied Mathematics, Vol. **119**, 2018, No 12, pp. 13309-13315.
2. A b d u l l a h, A. M. Advanced Encryption Standard Algorithm to Encrypt and Decrypt Data. 2017, pp. 22-27 (Online).
3. A k k a r, M. L., C. G i r a u d. An Implementation of DES and AES, Secure against Some Attacks. – Cryptographic Hardware and Embedded Systems 2001, Vol. 2162, 2001, pp. 309-318.

4. Rihan, S. D., A. Khalid, S. E. F. Osman. A Performance Comparison of Encryption Algorithms AES and DES. – International Journal of Engineering Research & Technology (IJERT), Vol. **4**, 2015, Issue 12, pp. 21-27. ISSN: 2278-0181.
5. Moharir, M., A. V. Suresh. Data Security with Hybrid AES-DES. – Elixir Comp. Sci. & Engg., Vol. **66**, 2014, pp. 20924-20926.
6. Ali, E. Taki El-Deen. Design and Implementation of Hybrid Encryption Algorithm. – International Journal of Scientific & Engineering Research, Vol. **4**, 2013, Issue 12, pp.12-16.
7. Rehman, S., N. Talal Bajwa, M. A. Shah, A. O. Aseeri, A. Anjum. A Hybrid AES-ECC Model for the Security of Data over Cloud Storage. – Electronics 2021, 2021, pp. 2673.
8. Bansal, S., Gagandeep Jagdev. Analyzing Working of DES and AES Algorithms in Cloud Security. – International Journal of Research Studies in Computer Science and Engineering (IJRSCSE), Vol. **4**, 2017, Issue 3, pp. 1-9.
9. Singh, G., S. Kinger. A Study of Encryption Algorithms (RSA, DES, 3DES and AES) for Information Security. – International Journal of Computer Applications (0975-8887), Vol. **67**, 2013, No 19, pp. 101-1011.
10. Mondal, S., S. Maitra. Data Security-Modified AES Algorithm and Its Applications. – ACM SIGARCH Computer Architecture News, Vol. **42**, 2014, No 2, pp. 1-8.
11. Dutta, A., P. Bharti, S. Agrawal, K. S. Surekha. Hybrid AES-DES Block Cipher: Implementation Using Xilinx ISE 9.1i. – In: International Conference on Advances in Electronics and Electrical Engineering, 2012, pp. 1-5.
12. Alanazi, H. O., B. B. Zaidan, A. A. Zaidan, H. A. Jalab, M. Shabbir, Y. Al-Nabhani. New Comparative Study Between DES, 3DES and AES within Nine Factors. – Journal of Computing, Vol. **2**, 2010, Issue 3, pp. 1-7.
13. Mahajan, P., A. Sachdeva. A Study of Encryption Algorithms AES, DES and RSA for Security. – Global Journal of Computer Science and Technology Network, Web & Security, Vol. **13**, 2013, Issue 15, p. 205.
14. Ghosh, S. N., D. T. Biradar, G. C. Shinde. Performance Analysis of AES, DES, RSA and AES-DES-RSA Hybrid Algorithm for Data Security. – Journal of Innovative and Emerging Research in Engineering, Vol. **2**, 2015, Issue 5, p. 170.
15. Kumar, P., S. B. Rana. Development of Modified AES Algorithm for Data Security. – Optik, Vol. **127**, 2016, Issue 4, pp. 2341-2345s.
16. Singh, G., S. Kinger. Integrating AES, DES, and 3-DES Encryption Algorithms for Enhanced Data Security. – International Journal of Scientific & Engineering Research, Vol. **4**, 2013, Issue 7, p. 2059.
17. Padmavathi, B., S. Ranjitha Kumari. A Survey on Performance Analysis of DES, AES and RSA Algorithm along with LSB Substitution Technique. – International Journal of Science and Research (IJSR), 2013, pp.170-174.
18. Araballi, A., Z. Subedar. Hybrid Cryptography: Performance Analysis of Various Cryptographic Combinations for Secure Communication. – International Journal of Mathematical Sciences and Computing, Vol. **6**, 2020, pp. 35-41.
19. Francis, N., T. Monoth. An Analysis of Hybrid Cryptographic Approaches for Information Security. – International Journal of Applied Engineering Research, Vol. **13**, 2018, pp. 279-319.
20. Kumari, S., S. Sangwan. Security in Cloud Computing Using AES & DES. – International Journal on Recent and Innovation Trends in Computing and Communication, Vol. **5**, 2017, No 4, pp. 194-200.

Received: 06.03.2023; Accepted: 25.08.2023