IMAGE ENCRYPTION AND DECRYPTION USING CONFUSION AND DIFFUSION TECHNIQUES

1. ABSTRACT

Image encryption converts an original image into an encrypted form, preventing unauthorized access, ensuring its confidentiality during transmission or storage. Blurring and diffusion techniques, utilizing complex mathematical algorithms, are employed to mix and hash image data, imparting high entropy and making changes in the ciphertext significantly impact the encrypted image. During encryption, obfuscation complicates input-output relations, employing substitution or permutation to generate seemingly random output. In diffusion, methods like pixel movement, rotation, or XOR with a key spread and mix data to affect the entire image with a single pixel change. Decryption reverses encryption steps, requiring a decryption key for authorized access. This process maintains a high level of security, resisting brute force or statistical attacks, as a single pixel alteration impacts the entire image. Hence, utilizing obfuscation and diffusion techniques is an effective method for securing image confidentiality during transmission or storage.

2. INTRODUCTION

In the dynamic sphere of data security, the perpetual balancing act between safeguarding sensitive information and its susceptibility to threats endures. The contemporary era, heavily reliant on the internet, witnesses an exponential surge in data volume, intensifying the urgency to ensure secure data transmission and storage. As data travels across the internet, the looming specter of potential attacks poses a considerable risk to the confidentiality and integrity of information. Hence, modern cryptographic algorithms play a pivotal role in fortifying data security, ensuring both confidentiality and integrity.

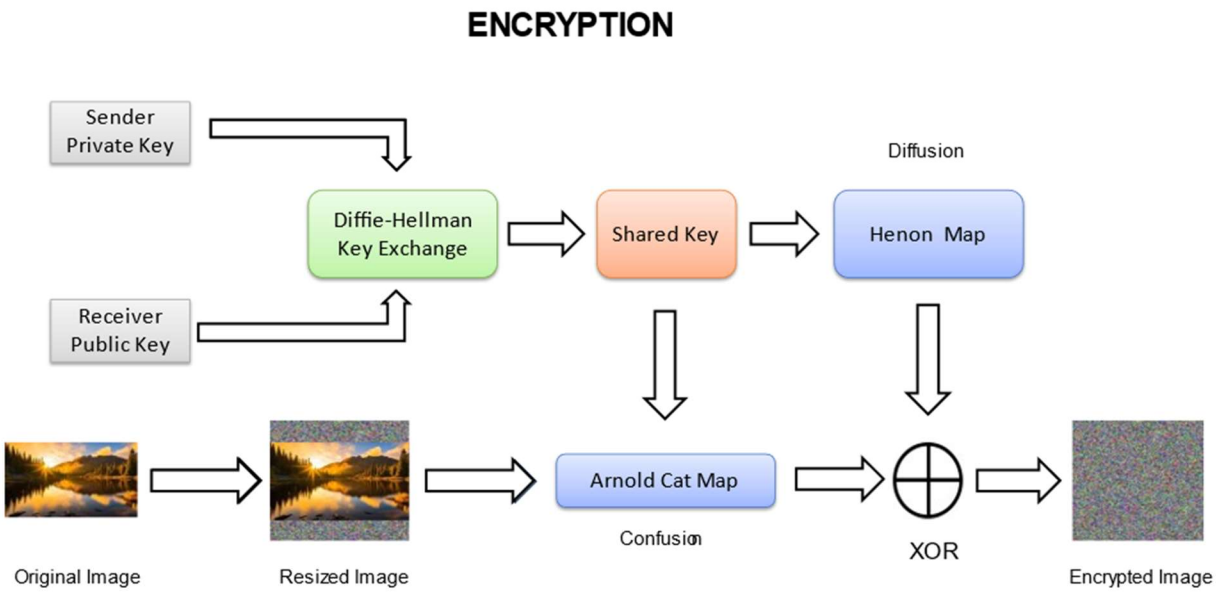
Two primary encryption methods come into play: Public-key cryptography and Private-key cryptography. Public-key cryptography employs a shared and a secret key, enabling secure data transfer and verification. Conversely, private-key cryptography relies on a singular secret key for encryption and decryption, safeguarding sensitive data.

A significant portion of internet-transmitted data comprises images, often containing sensitive personal information that could compromise privacy. Consequently, there's a growing urgency to ensure image security during both transmission and storage.

Chaotic maps, mathematical formulas producing sequences of highly unpredictable numbers, enhance image encryption by generating vastly different outcomes with minimal alterations in initial conditions, fortifying the security and privacy of the process.

The proposed encryption method for RGB images in this paper employs public-key cryptography, eliminating the need for a pre-shared private key between parties. By manipulating pixel values through a blend of Confusion and Diffusion—incorporating Arnold's Cat map and Henon's map—the aim is to fortify the image's integrity and confidentiality, making it resilient against diverse attacks.

3 PRELIMINARY



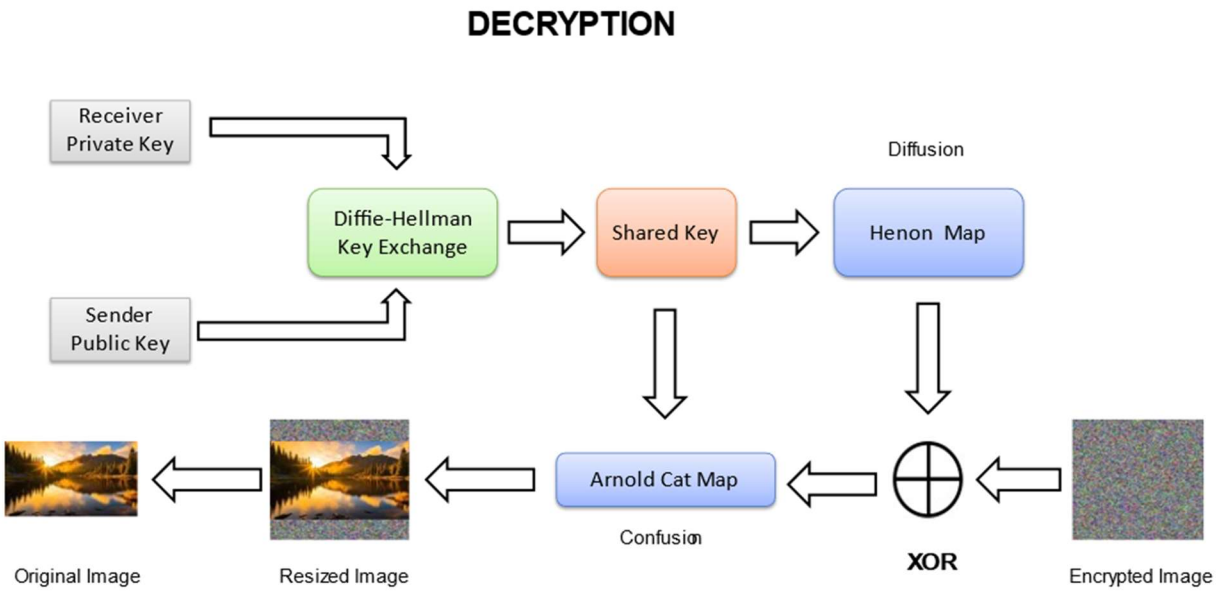


Fig.1: Flow diagram for proposed image encryption methodology

4 PROPOSED METHOD

4.1 Key pair generation

Step1 Initialize the values of 'P' and 'G' for the Diffie-Hellman Key exchange algorithm[2].

Step2 The users generate their Private keys 'Pk_1'and 'Pk_2' which is a random number of 100 bits in size.

Step 3 Their respective Public keys 'PUk_1' and 'PUk_2' are calculated by using the following formula.

$$PUk_1 = GPk_1 \pmod{P}$$

$$PUk_2 = GPk_2 \pmod{P}$$

Step 4 The obtained Public keys 'PUk_1' and 'PUk_2' are shared between the users. With these public keys a shared secret key 'K' is produced by using the formula.

$$K = PUk_2^{Pk1} \pmod{P}$$

$$K = PUk_1^{Pk2} \pmod{P}$$

4.2 Encryption process

A. Resizing the image into a square

Step1 The height and width of the original image is calculated.

Step2 The original image is converted from BRG format to BRGA format with alpha value = 255.

Step3 A new square image whose size is the maximum of width and height of the original image is generated with random pixel values and alpha value = 254.

Step4 The original image is placed in the center of the generated square image and thereby we get a new image on which the encryption process is further continued.

B. Confusion

Step1 The number of iterations 'I' is initialized as the sum of the last 6 digits of the shared key 'K'.

Step2 The values 'p' and 'q' are initialized as the first half of the digits of shared key 'K' and second half of the digits of shared key 'K' respectively.

Step3 The images dimensions are initialized to variable 'N'

Step4 For the each iteration from 1 to 'I' the new pixel positions 'x_{map}' and 'y_{map}' using the current pixel position 'x_A' and 'y_A' are calculated using the Arnold's Cat Map formula[3].

$$x_{map} = (x_A + y_A * p) \% N$$

$$y_{map} = (x_A * q + y_A * (p * q + 1)) \% N$$

C. Diffusion

Step1 The initial parameters for the Henon map 'X_H' and 'Y_H' are initialized using the first half and last half of the shared key 'K' and its ensured that they are below the value 0.97 .

Step2 Iteratively the values for the Henon map 'X_H' and 'Y_H' is calculated till the size of the image by using the formula[4]

$$X_H = Y_H + 1 - 1.4 * X_H^2$$

$$Y_H = 0.3 * X_H$$

Step3 A threshold value 'T' = 0.3992 and if $X_H <= 0.3992$ bit is allocated as '0' else it is allocated as '1' and 8 bits are grouped into a byte.

Step4 The image is flattened to a 1D array.

Step5 Bitwise operation of the Henon Map with the 1D array of the image yields a 1D output array.

Step6 The 1D output array is reconstructed to an image and gives the Final Encrypted Image

4.3 Decryption process

A. Reverting Diffusion

Step1 The initial parameters for the Henon map 'X_H' and 'Y_H' are initialized using the first half and last half of the shared key 'K' and its ensured that they are below the value 0.97 .

Step2 Iteratively the values for the Henon map[5] 'X_H' and 'Y_H' is calculated till the size of the encrypted image by using the formula

$$X_H = Y_H + 1 - 1.4 * X_H^2$$

$$Y_H = 0.3 * X_H$$

Step3 A threshold value 'T' = 0.3992 and if $X_H <= 0.3992$ bit is allocated as '0' else it is allocated as '1' and 8 bits are grouped into a byte.

Step4 The encrypted image is flattened to a 1D array.

Step5 Bitwise operation of the Henon Map with the 1D array of the image yields a 1D output array.

Step6 The 1D output array is reconstructed to an image and gives the Image after reverting diffusion.

B. Reverting Confusion

Step1 The number of iterations 'I' is initialized as the sum of the last 6 digits of the shared key 'K'.

Step2 The values 'p' and 'q' are initialized as the first half of the digits of shared key 'K' and second half of the digits of shared key 'K' respectively.

Step3 The images dimensions are initialized to variable 'N'.

Step4 For the each iteration from 1 to 'I' the new positions 'x_A' and 'y_A' using the current pixel position 'x_{map}' and 'y_{map}' are calculated using the Arnold's Cat Map formula.

$$x_{map} = (x_A + y_A * p) \% N$$

$$y_{map} = (x_A * q + y_A * (p * q + 1)) \% N$$

C. Cropping to Original size

Step1 The height and width of the original image is obtained by calculating the number of rows and columns in the encrypted image with alpha channel value equal to 255.

Step2 A New empty image matrix of RGB format is created.

Step3 The pixels with alpha value equal to 255 is filtered and copied to the empty image matrix.

Step4 The new matrix obtained is the cropped image with border removed.

5 RESULTS AND ANALYSIS

RGB images of the size 256*256 such as Lena, Tree, Girl and Couple are used for conducting tests and obtain the efficiency and security of the proposed image encryption methodology. Statistical analysis like Histogram, Entropy, Correlation[6] and Encryption quality were performed.

5.1 Histogram Analysis

5.2 Entropy Analysis

It measures how chaotic and different the pixel values are in a image. If the entropy of the image is less then there is a higher probability for an attacker to obtain the image due to the existence of certain patterns in the image. Hence a methodology which obtains High entropy is desired and gives high security and is less prone to attacks.

Entropy is calculated by the formula:

$$Entropy = \sum_{i=1}^{N} P(pix_i) * \log(P(pix_i))$$

P denotes the Probability of occurrence of a particular pixel and N denotes the number of pixels in the image

Table1: Entropy values of Original image and Cipher Image in various planes.

	Plane Name	Original Image Entropy	Cipher Image Entropy
Couple	**Red**	6.2499	7.9365
	Green	5.9642	7.8684
	Blue	5.9309	7.8566
Tree	**Red**	7.2104	7.9367
	Green	7.4136	7.9389
	Blue	6.9207	7.9302
Girl	**Red**	7.4379	7.9642
	Green	7.4566	7.9721
	Blue	6.9610	7.9106
Lena	**Red**	7.2796	7.9757
	Green	7.6315	7.9783
	Blue	6.9891	7.9717

5.3 Correlation Anlaysis:

Knowing the value of a pixel in an image makes it easier to estimate the value of the pixel around it, because they are connected. An encryption method that reduces this connection makes the image harder to attack.

Correlation is calculated by the formulas[7]

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} X_i$$

$$Var(x) = \frac{1}{N}\sum_{i=1}^{N} (X_i - E(X))^2$$

$$cov(P_1, P_2) = \frac{1}{N}\sum_{i=1}^{N}(P_{1_i} - E(P_1))(P_{2_i} - E(P_2))$$

$$Correlation_{P1P2} = \frac{cov(P_1P_2)}{\sqrt{Var(P_1)}\sqrt{Var(P_2)}}\sum_{i=1}^{N}(X_i - E(X))^2$$

Table 2: Correlation of different planes of the original image and cipher image[8]

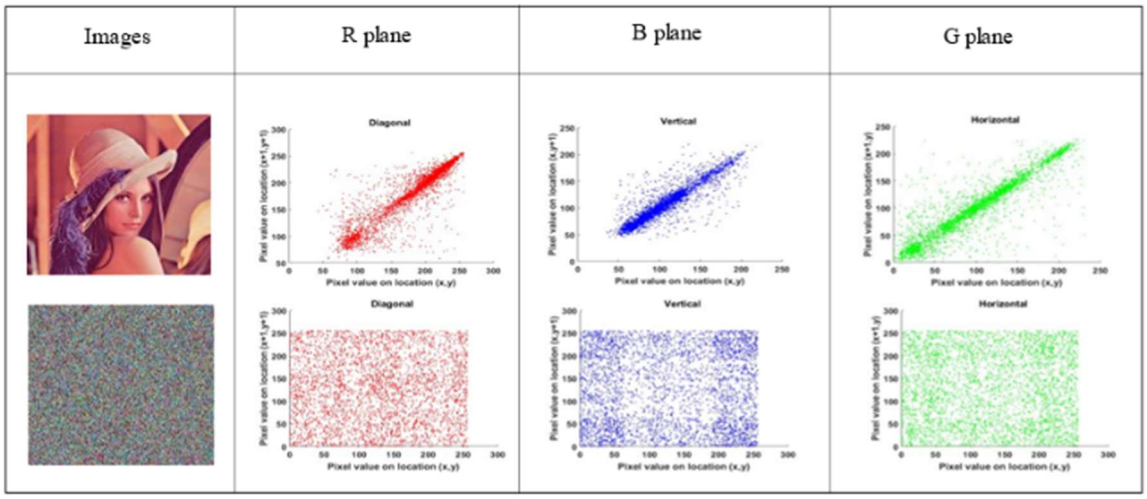


Table 3: Various correlation values of original and cipher images[9]

	Correlation Direction	Original Image			Cipher Image		
		Red	Green	Blue	Red	Green	Blue
Couple	**Horizontal**	0.9493	0.9308	0.9178	0.0343	0.0602	0.0588
	Vertical	0.9562	0.9534	0.9442	-0.0074	-0.0039	-0.0042
	Diagonal	0.9176	0.9002	0.8890	0.0004	0.0036	0.0038
Tree	**Horizontal**	0.9590	0.9687	0.9612	-0.0063	-0.0056	0.0003
	Vertical	0.9361	0.9457	0.9406	-0.0007	0.0048	0.0019
	Diagonal	0.9159	0.9318	0.9265	-0.0069	0.0004	-0.0024

Girl	**Horizontal**	0.9697	0.9509	0.9377	0.0010	0.0020	0.0107
	Vertical	0.9856	0.9781	0.9669	0.0013	0.0039	0.0007
	Diagonal	0.9572	0.9319	0.9106	0.0047	0.0016	0.0039
Lena	**Horizontal**	0.9338	0.9049	0.8597	0.0213	0.0015	-0.0079
	Vertical	0.9641	0.9467	0.9061	0.0029	-0.0053	-0.0015
	Diagonal	0.9074	0.8802	0.8383	-0.0001	0.0050	0.0038

5.4 Encryption Quality Analysis

This analysis uses the deviation between the pixel values of the original image and the cipher image to measure the quality of the encryption.

1.Maximum Deviation:

It's the difference between how the original image and cipher image look. First, the distribution of the pixel values for both images is computed. Here 'd' denotes how much the distributions vary from each other. When the value of Maximum deviation is large, the cipher image is more secure from attacks.

$$Maximum\ Deviation = \frac{d_0 + d_{255}}{2} + \sum_{i=1}^{254} d_i$$

2.Deviation from uniform histogram:

It's the deviation between the obtained histogram of the cipher image and uniformly distributed histogram[10]. When the value of it is small it is less prone to attacks.

$$Deviation\ from\ Uniform\ Histogram = \frac{\sum_{i=0}^{255} |256 - h_i|}{256 * 256}$$

3.Irregular Deviation:

It is how similar the distribution of the histogram deviation is to a uniform distribution. Here, ‘d’ is how much the image and its cipher differ. ‘h’ is the distribution of ‘d’. When the Irregular deviation value is low, it is more secure from attacks.

$$A_H = \frac{1}{256}\sum_{i=0}^{255} h_i$$

$$Irregular\ Deviation = \sum_{i=0}^{255} |h_i - A_H|$$

Table 4: Results of encryption quality analysis

	Maximum Deviation			**Deviation from Uniform Histogram**			**Irregular Deviation**		
	R	**G**	**B**	**R**	**G**	**B**	**R**	**G**	**B**
Couple	88221	97767	100778	0.2315	0.3536	0.3812	55589	59282	60035
Girl	26409	24540	35408	0.4166	0.4178	0.4504	20582	25567	21694
Tree	55621	55986	71395	0.2360	0.2401	0.2573	31757	32808	29593
Lena	57153	40137	77781	0.1519	0.1411	0.3120	20101	39570	37570

CONCLUSION

Finally, it must be stated that decryption and encryption of images are essential to ensure the safety and reliability of sensitive image information. Under encryption, mathematical algorithms and keys are used to convert images into unreadable formats so that they can be protected against unauthorized access. On the other hand, decryption reverts encryption so that an original image can be obtained using a correct key to decrypt it.

Image encryption techniques use different algorithms, such as symmetric key encryption and asymmetric key encryption. Symmetric key cryptography is fast and efficient because it uses the same key for both encryption and decryption. Asymmetric key cryptography, on the other hand, uses a pair of mathematically related keys, a public key for encryption and a private key for decryption. This provides better security, but higher computational requirements.

The solidness of an encryption calculation and the mystery of cryptographic keys are critical in guaranteeing that picture encryption is secure. Without any information on the right decoding key, vigorous encryption calculations guarantee that recovering the first image is computationally unimaginable. To guarantee that the unscrambling picture doesn't have unapproved access, encryption keys should be held and traded in a solid way among approved parties.

Overall, image encryption and decryption provide essential mechanisms for safeguarding visual data in various applications, including secure communication, data storage, and digital rights management. The combination of robust encryption algorithms, secure key management, and appropriate security practices ensures the confidentiality and integrity of images in today's digital world.

REFERENCES :

[1] Z. Man *et al.*, "A novel image encryption algorithm based on least squares generative adversarial network random number generator," *Multimed Tools Appl*, vol. 80, no. 18, pp. 27445–27469, Jul. 2021, doi: 10.1007/s11042-021-10979-w.

[2] Z. Bashir, M. G. A. Malik, M. Hussain, and N. Iqbal, "Multiple RGB images encryption algorithm based on elliptic curve, improved Diffie Hellman protocol," *Multimed Tools Appl*, vol. 81, no. 3, pp. 3867–3897, Jan. 2022, doi: 10.1007/s11042-021-11687-1.

[3] V. Kumar, V. Pathak, N. Badal, P. S. Pandey, R. Mishra, and S. K. Gupta, "Complex entropy based encryption and decryption technique for securing medical images," *Multimed Tools Appl*, vol. 81, no. 26, pp. 37441–37459, Nov. 2022, doi: 10.1007/s11042-022-13546-z.

[4] Y. Zhou, C. Li, W. Li, H. Li, W. Feng, and K. Qian, "Image encryption algorithm with circle index table scrambling and partition diffusion," *Nonlinear Dyn*, vol. 103, no. 2, pp. 2043–2061, Jan. 2021, doi: 10.1007/s11071-021-06206-8.

[5] F. Ren, Y. Liu, X. Zhang, and Q. Li, "Reversible information hiding scheme based on interpolation and histogram shift for medical images," *Multimed Tools Appl*, Jul. 2023, doi: 10.1007/s11042-022-14300-1.

[6] O. Farook Mohammad, M. Shafry Mohd Rahim, and F. Y. H Ahmed, “A Survey and Analysis of the Image Encryption Methods H2020 AniAge Project-"High Dimensional Heterogeneous Data based Animation Techniques for Southeast Asian Intangible Cultural Heritage Digital Content" View project,” 2017. [Online]. Available: http://www.ripublication.com

[7] R. Sivaraman, A. Vijaykumar, P. Savarinathan, and A. Jayapalan, “Chaos blended cellular automata on fractals: the effective way of reconfigurable hardware assisted medical image privacy,” *Multimed Tools Appl*, vol. 81, no. 23, pp. 33087–33106, Sep. 2022, doi: 10.1007/s11042-022-13165-8.

[8] B. Zhang, B. Rahmatullah, S. L. Wang, and Z. Liu, “A plain-image correlative semi-selective medical image encryption algorithm using enhanced 2D-logistic map,” *Multimed Tools Appl*, vol. 82, no. 10, pp. 15735–15762, Apr. 2023, doi: 10.1007/s11042-022-13744-9.

[9] A. Rengarajan and N. Chidambaram, “A Dual 3 2 Nibble Specific Cipher Model for RGB Images using Lorenz Attractor.”

[10] A. Ihsan and N. Doğan, “Improved affine encryption algorithm for color images using LFSR and XOR encryption,” *Multimed Tools Appl*, vol. 82, no. 5, pp. 7621–7637, Feb. 2023, doi: 10.1007/s11042-022-13727-w.