A Novel Approach to Enhance Image Security using Hyperchaos with Elliptic Curve Cryptography

Ganavi M, Jawaharlal Nehru New College of Engineering, India

Prabhudeva S, Jawaharlal Nehru New College of Engineering, India

ABSTRACT

Information securities dominate the world. All the time we connect to the internet for social media, banking, and online shopping through various applications our priceless data may be hacked by attackers. There is a necessity for a better encryption method to enhance information security. The distinctive features of elliptic curve cryptography (ECC) include key atomity, speedy ciphering, and preserving bandwidth captivating its use in multimedia encipher. An encryption method is proposed by incorporating ECC, Secure Hash Algorithm – 256 (SHA-256), Arnold transform, and hyperchaos. Randomly generated salt values are concatenated with each pixel of an image. SHA-256 hash is imposed which produces a hash value of 32-bit, later used to generate the key in ECC. Stronger ciphering is done by applying Arnold's transformation and hyperchaos thereby achieved more randomness in image. Simulation outcomes and analysis show that the proposed approach provides more confidentiality for color images.

KEYWORDS

Arnold Transform, Decryption, Elliptic Curve Cryptography, Encryption, Hyperchaos, MSE, PSNR, Salting, SHA-256

INTRODUCTION

Secured digital image communication is possible by one of the means like Image encryption. Securing the characteristics of image data is predominant in medical, military, and commercial domains. Safeguarding interactive media intelligence against prohibited access became a critical issue in our routine. Minutiae of images are also surveyed & deployed by third-party which leads to immeasurable damage for the image proprietor. To overcome such difficulties, digital image techniques are mandatory to be applied to images to encrypt before transmitting them. Information security is required to reduce the risk level that is tolerable to the business. Security challenges for sensitive data exchange through online networks are confidentiality, authenticity, integrity, non-repudiation, and availability. The conventional encryption standards are not able to fulfill the demands of image scrambling. The chaotic nature of hypersensitive to inceptive state and system framework, no recurrence, and generating unpredictable codes. The resultant chaotic sequence is accurate (Huang et al., 2018) and it is of great significance in encryption algorithms.

Hash functions are tremendously beneficial in cryptography. These functions emerge as a prominent role in the applications of data security. They can accomplish preservation, perfection,

DOI: 10.4018/IJRSDA.288520

and regularity of data, authenticate a message and guarantee the authenticity by a digital signature (Seyedzade, 2010; Wang, 2018). It should obtain different security properties like collision resistance, pre-image resistance, and pseudo-randomness. The most commonly used hash algorithm, SHA-256 results in a unique 32-byte hash value for input data. This value may be used as a key to encrypt data. The generated key value is different for different input data. Image scrambling can be carried out by applying Arnold transformation (Min et al., 2013). The significant feature of this transformation is that it uses periodicity. The number of iterations to recover back original input is calculated based on the image size. Arnold transformation is enforced pixels as well as continued to image chunks which enhances the robustness and security level (Sathish, 2019). A cryptographic hash is required in securing the passwords or secret images when stored in memory to protect them from birthday and dictionary attacks. So always it is recommended to apply hash on a combination of salt and passwords or important pictures (Gauravaram, 2012).

ECC (Koblitz, 1987; Miller, 1985) uses smaller key sizes compared to Rivest, Shamir & Adleman (RSA) thereby providing more security (Bakr et al., 2018). It is the next level to asymmetric cryptosystem on the numerical design of structures over finite fields (Gutub et al., 2007; Laiphrakpam and Khumanthem, 2018; Koblitz, 1987; Ziad et al., 2018).

ECC has been a recent analysis topic within the space of information security (Kamalakannan and Tamilselvan, 2015; Vigila and Muneswaran, 2009; Roy et al., 2014). The proposed method uses ECC for generating the key required to encrypt/decrypt process for secure transfer of input image. A salt is randomly generated to the size of an image. An obtained random number is concatenated with each pixel value of an input image. Applying SHA-256 hashing on this image will generate a 32-byte hash value. The key to the elliptic curve point is communal in the middle of the sender and receiver. The number of times Arnold's transformation (Min et al., 2013) was carried is based on the value of the elliptic curve point. This point value is also applied to achieve the initial framework of the hyperchaos system. The resultant hyperchaos output is XOR with Arnold's transformed image (Sathish, 2019; Kaur and Talwar, 2017) to generate the cipher image. This proposed method prevents the necessity of communication of the look-up table information.

LITERATURE SURVEY

Avoiding intermediate knowledge from unofficial usage has become a critical and sensitive issue in this internet era. A novel approach has been suggested by adopting chaos and SHA-1(Slimane et al., 2017). Confusion and diffusion processes are applied to images to encrypt. An approach with a two-diffusion process and SHA-1 to obtain a private key based on nested chaotic encryption has been proposed (Slimane et al., 2016). Various security analyses, tests, and attacks are also explained. A hash function is suggested based on a chaotic system (Wadhwa et al., 2016). Input data is divided into pieces and passed through chaos separately. This approach uses the block ciphering technique which uses a plain image. High-dimensional chaotic systems have been proposed (Qi et al., 2016). Lorenz mapping is used to generate the Hyperhenon to improve the keyspace.

A comprehensive survey on chaotic image secret writing schemes is presented (Singh et al., 2018). Chaotic secret writing is extraordinary compared to other approaches to accomplish security. An image secret writing scheme using chaotic maps has been applied. A hybrid approach has been proposed by applying the permutation of input data using hyperchaos (Hassene and Eddine, 2016). Scrambling text and pictures is recommended in this approach resulting in the reduction of computational cost and better permutation. The SHA-2 algorithm is presented (Ibrahim et al., 2015). Scrambling and diffusion stages are presented (Salagundi et al., 2016). Circular operation in row and column scrambling is considered. A chaotic map is applied to each row and column to scramble. Parity is used in modifying picture elements for the diffusion stage.

A structure has been proposed (Abdoun et al., 2016) that blends the chaotic generator into neurons. Pixel shuffling and chaotic map methods are presented (Chaitanaya et al., 2015). This algorithm uses the Henon map to generate the values for the key. A system that uses multiple chaotic-based circular mapping has been proposed (Sathishkumar et al., 2011). The scanning pattern like raster and Zigzag is applied to the array and then separate into many sub-blocks. An algorithm that uses SHA-512 has been presented (Seyedzade et al., 2010). This uses a one-dimensional hash function like SHA-2 and generates a mask of two-dimensional. Traditionally, Arnold is applied to the square area of a picture. Focused on a multi-region for scrambling (Min et al., 2013). In this method, the number of square areas is generated from non-square pictures and then scrambles each part. A method has been imposed to enhance difficulty at decryption by applying Arnold scrambling (Sathish et al., 2019). An approach using chaos & SHA-256 has been presented (Zhu et al., 2018). To make each scramble different, the SHA-256 hash is used.

A scrambling approach has been proposed, combining many hash functions with cyclic shift operations (Wang, 2018). Chaotic sequence and logistic maps are used to scramble images. A security measure of a hash of salt combining passwords has been presented (Gauravaram, 2012). Hashes are vulnerable to birthday attacks. To generate many passwords for a single combination of salt and password and analyze the security measure. Salt with passwords is dealt with Davies-Meyer which is a compression function. Also furnished with an ofñine birthday forgery attack. This method overcomes prepended salts from birthday attacks. Explained corrective measures against attack.

With all these existing approaches, the motivation of this paper is to enhance the confidentiality of color images. Scrambling algorithms are used in randomizing the images which are helpful to achieve confidentiality. A novel approach is proposed where it combines symmetric and asymmetric methods in image scrambling. The main objectives are to achieve more randomness for the input image and to also scramble the generated keys. Scrambling is carried out in two processes. One is for the input image, and another is for the keys to be used. ECC algorithms are suitable for key mediation, digital signatures, and pseudo-random generators. Authentication approaches based on ECC provides better security for data transfer in mobile phones, smart cards, financial transaction, and sensitive information. Therefore, a complex asymmetric algorithm like ECC along with SHA-256 is used in the generation of the key and its scrambling for the proposed method. Hashing, SHA-256 is so strong that, even though the attacker gets a hash value, it is not possible to reverse back the original contents of the input data. The technique salting is also applied on the input image initially to make it difficult for the attackers to get back the actual keys. As symmetric algorithms are faster in computation, like Arnold's transformation and hyperchaos are used in image encryption. This helps in achieving more randomness in cipher images thereby enhancing the confidentiality for color images.

ENCRYPTION ALGORITHMS

Hashing

Hashing is the encryption method that builds a rare, absolute length signature for any input. These are used to generate the hash values. These hash values are used to correlate sets of data. A small change in data results in different hash values. It is a one-way function. This is the major difference among other encryption methods. Many algorithms are in use today. More popularly used are Message Digest 5 (MD5) and Secure Hash Algorithm (SHA).

Private Key Cryptography

Private Key cryptography is a secured and earliest encipher method. A single key value is kept as a secret because any third party knowing this key value can decode it. The same is used at the sender to scramble & descramble at the receiver. Based on data size, it is categorized as stream cipher and block cipher. Character by character is encrypted in stream cipher whereas block cipher encrypts fixed length of data. Most commonly used are the Data Encryption Standard (DES) and Advanced Encryption Standard (AES).

Algorithm

Input: Color image (m x n); Output: cipher (key$_1$), cipher (key$_2$)
1. Load input color image and convert it to grayscale.
2. The random number matrix is generated.
3. Concatenate random number matrix with grayscale image results to salted image.
4. Generate a 32-byte hash by applying SHA-256 on the salted image.
5. Use ECC to generate the base pointer, private, and public keys.
6. Apply point multiplication between hash value and the base pointer.
7. Generate cipher(key$_1$) by encrypting it with point addition.
8. Generate cipher(key$_2$) by encrypting the text file containing initial values required for hyperchaos.
9. Secretly share cipher(key$_1$) and cipher(key$_2$) with the receiver.

Public Key Cryptography

The public key method gives added security to private key encipher methods. As in the symmetric method, there is no need of maintaining a single key between many users. To overcome such difficulty, the two different keys are used to scramble or descramble. In this cryptography method, all users are given a public key value which is helpful to encrypt input data generating cipher data. A private key value is provided with the intended receiver to decrypt the cipher data. Algorithms like RSA, ECC, and Diffie-Hellman make use of asymmetric enciphering.

PROPOSED METHOD

This approach is designed to achieve the confidentiality of images. The system is designed with three stages like key generation, encryption, and decryption. Two key values such as *'kG'* and initial values for hyperchaos are used in this proposed approach. The ECC is adopted in the process of generation of key *'kG'* to get the elliptic curve point. *'kG'* is used to generate *cipher(key$_1$)*. The initial values for hyperchaos are stored in text file *Tm* is encrypted by ECC generating *cipher(key$_2$)*. Arnold's transformation & hyperchaos encryption are used for scrambling an input image during the process of encryption. The number of times Arnold's transformation (Min et al., 2013) was carried out is based on key *'kG'*. The resultant of hyperchaos is point multiplied with this key *'kG'* to generate the cipher image. Two keys *'kG'* and initial values *(Tm)* for hyperchaos are encrypted using ECC given by equations (2) & (3). The *cipher(key1)* and *cipher(key2)* are shared between the sender and receiver for decryption. At the receiver, keys required for decryption, *kG,* and *Tm* are decrypted using equations (4) & (5). To decrypt the original image, the reverse procedure is applied to the received image.

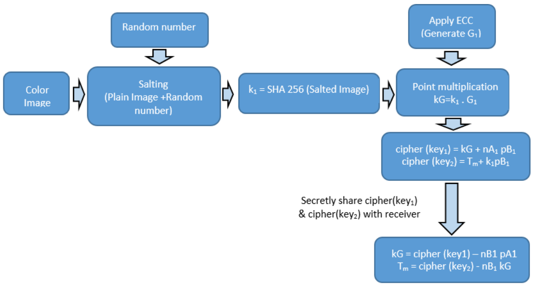
Key Generation

The procedure for the key generation in the proposed method is illustrated in Figure 1. Input color image from the database. Applying SHA-256 hashing on the salted image generates a 32-byte hash value *k1*. Use the elliptic curve equation (Gutub et al., 2007; Koblitz, 1987) over a finite field F_p to generate the base pointer '*G1*', private keys *(nA1 & nB1)*, and public keys *(pA1 & pB1)* both at sender and receiver, respectively.

Apply point multiplication (Laiphrakpam and Khumanthem, 2018) to generate

$$kG = k1 \bullet G1 \tag{1}$$

Figure 1. Key generation


Random number
Apply ECC (Generate G_1)
Color Image
Salting (Plain Image +Random number)
k_1 = SHA 256 (Salted Image)
Point multiplication $kG=k_1 . G_1$
cipher (key_1) = kG + nA_1 pB_1
cipher (key_2) = T_m+ k_1pB_1
Secretly share cipher(key_1)
& cipher(key_2) with receiver
kG = cipher (key1) – nB1 pA1
T_m = cipher (key_2) - nB_1 kG

Generate *cipher(key1)* by encrypting it with point addition and share it with the receiver.

$$cipher\left(key1\right) = kG + nA1\ pB1 \tag{2}$$

where, nA_1 = private key of the sender, pB_1 = public key of the receiver. Generate cipher(key_2) by encrypting the text file containing initial values required for hyperchaos using

$$cipher\left(key2\right) = Tm + k1\ pB1 \tag{3}$$

where, *Tm* = text file, *k1* = generated hash value, *pB1* = public key of receiver. At the receiver, perform point subtraction and extract the secretly shared key *kG* from *cipher*(key_1) used as a key for Arnold transformation, computed as

$$kG = cipher\left(key1\right) - nB1\,pA1 \tag{4}$$

where, *nB1*= private key of the receiver, *pA1* = public key of the sender. Use decrypted *kG* to extract *Tm*. So now apply point subtraction to extract the text file containing initial values for hyperchaos decryption.

$$Tm = cipher\left(key2\right) - nB1\ kG \tag{5}$$

Encryption and Decryption

The procedure for the encryption of color images in the proposed method is illustrated in Figure 2.

The number of rounds of iterations for Arnold transform is computed based on n1, a prime number as

$$\mathrm{r1} = \mathrm{mod}\left(\left(\mathrm{kG}\left(1\right) + \mathrm{kG}\left(2\right)\right), \mathrm{n1}\right) \tag{6}$$

Arnold's transformation (Min et al., 2013) to scramble the image is given as

Algorithm

Input: Color image (m x n); Output: cipher image
1. Load the color image from the dataset and divide it into R-, G- and B-channel.
2. Calculate the number of iterations ‘r_1’ from key kG (4).
3. For the R-channel image, apply Arnold’s transformation for the ‘r_1’ number of iterations. This results in the first level of image scrambling.
4. Consider the initial key values required for hyperchaos from the text file T_m.
5. Apply confusion and diffusion process by using the hyperchaos system and Chebyshev maps. This results in the second level of image scrambling.
6. Perform point multiplication with kG.
7. Repeat the process for G- and B-channel images and combine to get the colored cipher image.

$$\begin{pmatrix} I1' \\ I2' \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} I1 \\ I2 \end{pmatrix} \left(mod\ I3\right) \tag{7}$$

where I1 & I2 can take values in the range of {0, 1, 2, 3… I_3-1}, I3 is the image size of the input, $(I1, I2)^T$ is the initial array of the input, and $(I1’, I2’)^T$ is the resultant array of an Arnold transformed image. Use hyperchaos system and Chebyshev maps (Zhu et al., 2018; Li et al., 2019). The initial values are x0 as 0.398, y_0 as 0.456, z_0 as 0.784, and w_0 as 0.982. System parameters are a_1 as 35, b_1 as 3, c_1 as 12, d_1 as 7 and e_1 as 0.1583. A hyperchaos system can be modeled as

$$\frac{dx0}{dt} = a1^*\left(y0 - x0\right) + w0 \tag{8}$$

$$\frac{dy0}{dt} = \left(d1^*x0\right) - \left(x0^*z0\right) + \left(c1^*y0\right) \tag{9}$$

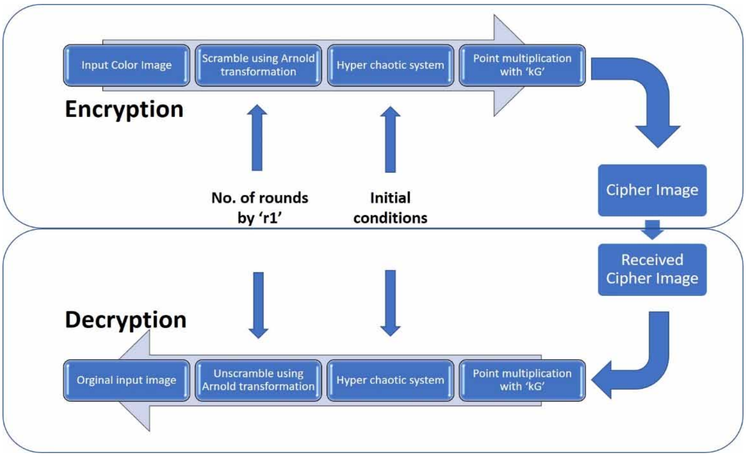
$$\frac{dz0}{dt} = \left(x0^*y0\right) - \left(b1^*z0\right) \tag{10}$$

$$\frac{dw0}{dt} = \left(y0^*z0\right) + \left(e1^*w0\right) \tag{11}$$

The Chebyshev maps are modeled as

$$u1\left(i+1\right) = \cos\left(4 * a1 \cos\left(u1\left(i\right)\right)\right) \tag{12}$$

Figure 2. Proposed system framework for encryption and decryption



$$u2\left(i+1\right)=\cos\left(4*a1\cos\left(u2\left(i\right)\right)\right) \tag{13}$$

Perform point mtiplication between output of hyperchaos *HCf(i)* and *kG.*

$$CI\left(i\right)=HCf\left(i\right)\bullet kG$$

14

The procedure for the decryption of color images in the proposed method is illustrated in Figure 2. Perform point multiplication between received cipher *CI(i)* and *kG.*

$$HCf\left(i\right)=\mathrm{CI}\left(\mathrm{i}\right)\bullet\mathrm{kG} \tag{15}$$

Produce the required chaotic sequence $A_1, A_2, A_3, A_4,$ *and* A_5 (Li et al., 2019; Zhu et al., 2018) by making use of initial key values

Algorithm

Input: cipher image; Output: Color image (m x n)
1. The received cipher is divided into R-, G- and B-channel.
2. For the R-channel image, perform point multiplication with kG.
3. Extract the initial key values required for hyperchaos from the text file Tm.
4. Apply hyperchaos system and Chebyshev maps for the first level of image descrambling.
5. Calculate the number of iterations 'r$_1$' from key kG (4).
6. Apply inverse Arnold's transformation for the 'kG' number of rounds to extract the R-channel image. This results in the second level of image descrambling.
7. Repeat the process for G- and B-channel images and combine to get back the colored image.

$$p0\left(1\right) = mod\left(HCf\left(1\right)\mathrm{Enc}\left(1\right) - \mathrm{HCF}\left(\mathrm{ind1}\left(1\right)\right),\ \mathrm{M1}\right) \tag{16}$$

Here, $M_1 = 256$. Extract *p0(1)* by decrypting it as

$$p0\left(i\right) = mod(CI\left(i\right)\mathrm{mod}\left(\mathrm{Enc}\left(\mathrm{i}\right) + \mathrm{p0}\left(\mathrm{ind2}\left(\mathrm{i}\right)\right),\ \mathrm{M1}\right) - \mathrm{CI}\left(\mathrm{ind2}\left(\mathrm{i}\right)\right), \mathrm{M1} \tag{17}$$

for *i* from (l-1) to 2

$$p0\left(1\right) = mod\left(CI\left(1\right)\mathrm{mod}\left(\mathrm{A5}\left(1\right) + \mathrm{A6}\left(1\right),\ \mathrm{M1}\right) - \mathrm{A4}\left(1\right),\ \mathrm{M1}\right)$$

$$h\left(i\right) = i + mod\left(floor\left(A1\left(i\right) * g * 10^{14}\right), w - i\right)$$

$$p0\left(i\right) = p0\left(h\left(i\right)\right)$$

$$HC\left(i\right) = p0 \tag{18}$$

Apply inverse Arnold's transformation (Min et al., 2013) for 'kG' number of rounds

$$\begin{pmatrix} I1 \\ I2 \end{pmatrix} = \begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} I1' \\ I2' \end{pmatrix} \left(mod\ I3\right) \tag{19}$$

where *I1' & I2'* can take values in the range of {0, 1, 2, 3… I_3-1}, *I3* is the image size of the *HC(i), (I1, I2)*^{T} is the initial matrix of the *HC(i), and (I1', I2')*^{T} is the matrix of the original image.

SIMULATION RESULTS

The set of input color images used for the proposed work are given in Figure 3. Input images are available from USC-SIPI Image datasets, KODAK Image dataset, and Image Manipulation Dataset. A total of 210 images were used from USC-SIPI datasets, 24 images from Kodak, and 96 images from FAU datasets. Here snapshots are shown for the Mandril color image and its cipher in Figure 4. Histograms for an input image, R-, G-, B- channels of the cipher, and for a reconstructed image are represented in Figure 5 and Figure 6. It is observed that the histograms of the cipher show uniform distribution of information, which makes any viewer judge the tonal distribution. This shows more randomness is achieved from the proposed approach. Horizontal-, vertical- & diagonal- pixel correlation for an input image, R-, G-, B- channels of ciphered image and a reconstructed image is given respectively in Figure 7. The correlation coefficient among pixels for input and reconstructed images are linear. The proposed approach does not maintain any linear correlation among the picture elements for the cipher image.

The results from the experiments show that the proposed scheme does not retain any linear correlation between the observed pixels in any direction, as the correlation values of the encrypted versions are very close to 0.00.

Figure 3. Set of color images used as input

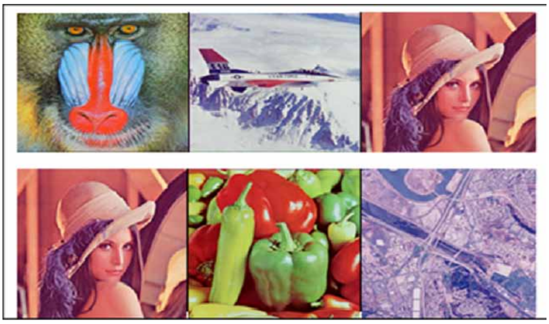


Figure 4. Original input color image and Cipher image

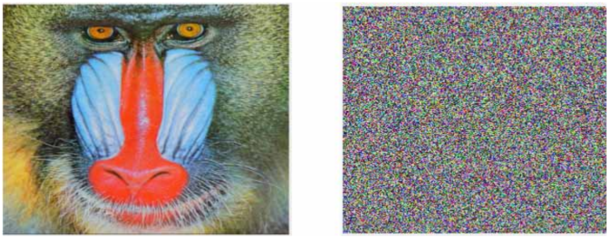
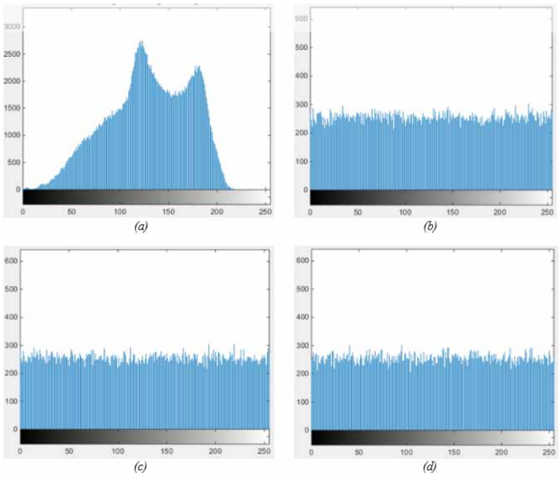


Figure 5. Histogram of (a) Input image & (b-d) Cipher images (R, G, B)



Security Analysis

Peak Signal to Noise Ratio (PSNR), Mean Square Error (MSE), and Structural Similarity Index Measure (SSIM)

PSNR is used for the computation of quality assessment among images. It is computed as the proportion amid in greatest conceivable signal estimation and the strength of falsifying noise that modifies its quality. A higher PSNR value represents higher quality reconstruction and a lower PSNR

Figure 6. Reconstructed image (output image) & Histogram

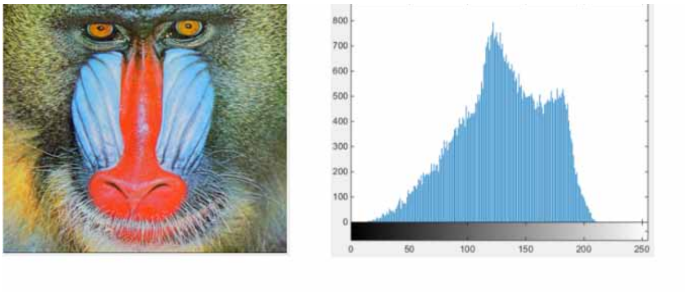
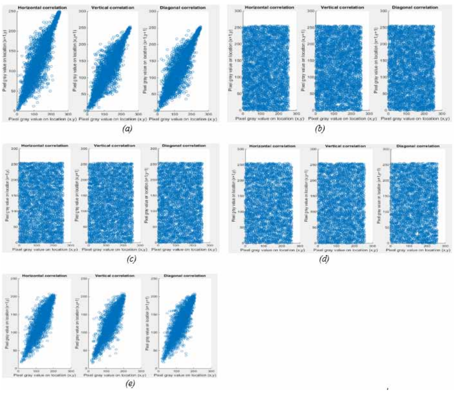


Figure 7. Horizontal-, vertical- & diagonal- pixel correlation of (a) an input image, (b)-(d) cipher image (R, G, B) & (e). Reconstructed original image



value for cipher represents more randomness so that it is difficult for an intruder to reconstruct. PSNR is expressed as

$$PSNR = 10\log_{10}\left(\frac{255^2}{MSE}\right) \tag{20}$$

MSE is used to evaluate the error among estimated and actual values. It is computed as an average of the square of the difference between two values. The littler MSE esteem speaks to better picture quality and higher MSE gives better image security. MSE is expressed as

where *P1 X P2,* number of columns & rows in the picture, *J1(i, j)* is precedent, and *J1'(i, j)* is an approximated version. Values obtained for PSNR and MSE, both for cipher and decrypted images in the proposed approach are given in Table 1.

An SSIM measure assesses the perceptual dissimilarity among two indistinguishable images. It is based on the measurement of standard deviation *(σ)* and means *(μ)* to extract the statistical image features. SSIM is calculated by using

where *I(x, y)* is a structural similarity measure among test (x) and reference (y) images. The quantity μx is the mean of pixels in x, σx^2 is the variance of pixels in x, μy is the mean of pixels in y, and σy^2 is the variance of pixels in y. c_1 & c_2 are constants: $c_1=(k_1L)^2$ and $c_2=(k_2L)^2$ where *k* is a small constant and L =255, a maximum value of pixels. The set of SSIM values obtained are shown in Table 1. SSIM of the cipher image should be nearer to zero and should be 1 for the decrypted image.

Table 1. Performance metrics of PSNR, MSE, and SSIM

Data sets		**Cipher**			**Decrypted**		
		PSNR (dB)	**MSE**	**SSIM**	**PSNR (dB)**	**MSE**	**SSIM**
USC-SIPI	Aerials	9.5851	7.2602e+03	0.0103	62.4419	0.1220	0.9998
	Miscellaneous	8.7244	9.1388e+03	0.0095	61.8929	0.1590	0.9997
	Sequences	8.7206	8.7976e+03	0.0095	60.6514	0.1569	0.9997
	Textures	8.7206	8.7976e+03	0.0095	60.6514	0.1569	0.9997
Kodak		8.6510	8.9482e+03	0.0104	51.3848	0.5896	0.9995
FAU		6.5732	15.651e+03	0.0047	Infinity	0.1298	0.9998

From Table 2, it is observed that PSNR for the cipher is much lesser when compared to the other two existing methods and it is 59.40448(dB) for the decrypted image. MSE is higher for cipher and much lesser for the decrypted image. For the proposed method, SSIM is lesser compared to another method Laiphrakpam and Khumanthem (2018) but for the decrypted image, it is nearer to 1.

Table 2. Comparing values of PSNR, MSE & SSIM

Existing Methods	**Cipher**			**Decrypted**		
	PSNR (dB)	**MSE**	**SSIM**	**PSNR (dB)**	**MSE**	**SSIM**
Dawahdeh et.al., (2018)	8.5777	NA	NA	NA	NA	NA
Laiphrakpam and Khumanthem (2018)	8.5655	NA	0.03253	Infinity	NA	1
Average of Proposed method	**8.4958**	**9.7655e+03**	**0.0089**	**59.40448**	**0.2190**	**0.9997**

Number of Pixel Change Rate (NPCR), and Unified Average Changing Intensity (UACI), and Information Entropy

NPCR & UACI are employed to estimate the ability of any cryptosystem quantitatively. NPCR is the difference in the pace of the number of pixels in cipher when just a single pixel of an input image is adjusted.

Where *A ()* =

UACI is the average level matrix change among the pixels of two images.

$$UACI = (\sum_{w1=1}^{P}\sum_{w2=1}^{Q}\left(\frac{\left|y1\left(w1,w2\right)-y1'\left(w1,w2\right)\right|}{255X\ P\ X\ Q}\right))\ X\ 100\% \tag{24}$$

where *PXQ* is image dimension, $y1(w1, w2)$ is ciphertext pixel concerning original plaintext, and $y1'(w1, w2)$ is ciphertext pixel concerning changed plaintext. The values obtained for NPCR and UACI in the recommended approach are given in Table 3.

Table 3. Performance metrics of NPCR, UACI, and Information Entropy

Data sets		Cipher			input
		NPCR (%)	UACI (%)	Information entropy	Information entropy
USC-SIPI	Aerials	99.9998	33.4635	7.9971	6.9448
	Miscellaneous	96.1513	33.4635	7.9972	6.1503
	Sequences	99.7908	33.4635	7.9972	7.0890
	Textures	99.7908	33.4635	7.9972	7.0890
Kodak		98.1485	33.4635	7.9972	7.9972
FAU		99.1093	33.4635	7.9972	7.9972

Information accumulation in the source of an image is always calibrated by entropy. It is emphasized as a mean of the number of grey levels in an image. It is helpful in quantifiable investigation and assessment of information. For any greyscale images, the maximum entropy value is 8-bits. The values obtained for information entropy are tabulated in Table 3. The entropy of information is expressed as

$$E = -\sum_{k=0}^{N} pk \log_2 \left(pk\right) \tag{25}$$

where *pk* is the probability of contingence of grey amount *k*. Measuring information is more disorienting if it produces larger entropy.

NPCR should be nearer to 100 and UACI ideal value is 33.46. For the proposed method, NPCR achieved is 100% and UACI is 33.4635(%), shown in Table 4. Information entropy for cipher should be nearer to 8. The information entropy of the input image is 6.2205. For the proposed method, information entropy is 7.9972 which is nearer to 8 and it is a bit higher compared to the other two existing methods.

Table 4. Comparing values of NPCR, UACI & information entropy

Existing methods	Cipher			Input
	NPCR (%)	UACI (%)	Information entropy	Information entropy
Dawahdeh et.al., (2018)	NA	30.4814	7.9970	NA
Bakr et al., (2018)	NA	31.1278	7.9957	NA
Average of the proposed method	**98.8317**	**33.4635**	**7.9972**	**6.2205**

Correlation Coefficient of a Pixel

The correlation coefficient of an image is represented among two adjacent pixels horizontally, vertically, and diagonally. Continuously, the point of any encryption technique is to cut back the relationship coefficient esteem between neighboring pixels. Such encryption gives more security. It is computed as

$$XC = \frac{\boldsymbol{Q}1\sum_{i=1}^{Q1}\boldsymbol{SiTi} - \sum_{i=1}^{Q1}\boldsymbol{Si}\sum_{i=1}^{Q1}\boldsymbol{Ti}}{\sqrt{\boldsymbol{Q}1\sum_{i=1}^{Q1}\boldsymbol{Si}^2 - \left(\sum_{i=1}^{Q1}\boldsymbol{Si}\right)^2}\sqrt{\boldsymbol{Q}1\sum_{i=1}^{Q1}\boldsymbol{Ti}^2 - \left(\sum_{i=1}^{Q1}\boldsymbol{Ti}\right)^2}} \tag{26}$$

where *Si and Ti* are two neighboring pixels and *Q1* is the number of pixel combinations chosen. The correlation coefficient for plain & cipher is given in Table 5. It is observed as the correlation coefficient is very less for the cipher image when compared to a plain image. So, the proposed method gives a better encryption effect and a higher security level.

Time And Space Complexity Analysis

The efficiency of this encryption algorithm is tested. The execution is carried out in MATLAB R2019b with Intel(R) Core (TM) i5-4210U CPU @ 1.70GHz and 4.00GB RAM on Windows 10 OS. In this method, the input color image is encrypted by *‘r1’* rounds first, then followed by one round

Table 5. Correlation coefficient values

Input color images	Channel	Plain images			Cipher images		
		Horizontal	Vertical	Diagonal	Horizontal	Vertical	Diagonal
Mandril.tiff	R	0.91410	0.92386	0.92017	-0.01153	-0.02641	0.00729
	G	0.86598	0.85997	0.87783	0.01994	-0.00199	-0.00111
	B	0.91152	0.90656	0.90662	0.01111	-0.00194	0.00196
CT scan.tiff	R	0.92501	0.92242	0.91607	-0.00305	0.01863	-0.01441
	G	0.86424	0.86287	0.85993	-0.00422	-0.01513	-0.01476
	B	0.90425	0.90515	0.91227	-0.01684	-0.00684	-0.01432
lena_color_256.tif	R	0.95717	0.95786	0.96088	-0.01958	-0.00372	-0.00758
	G	0.93358	0.94381	0.93964	0.01148	0.00941	-0.00457
	B	0.91397	0.92131	0.91158	-0.00798	-0.00104	-0.00666
lena_color_512.tif	R	0.97784	0.98149	0.97880	0.01059	0.00605	-0.01441
	G	0.96703	0.96915	0.96763	-0.01595	0.00792	0.00228
	B	0.93873	0.92988	0.92914	0.01379	0.00474	0.00263
Peppers.tiff	R	0.96467	0.96494	0.96004	-0.03770	-0.03102	0.00544
	G	0.98340	0.98190	0.98144	-0.00281	-0.02910	0.00111
	B	0.95995	0.96663	0.96893	0.00485	-0.00002	-0.02523
San Diego.tiff	R	0.84734	0.85506	0.83940	-0.01900	0.00852	0.00426
	G	0.78496	0.78084	0.77387	0.02253	-0.00053	0.00413
	B	0.75854	0.74835	0.75166	-0.00881	0.02017	-0.03450

of hyperchaos, and the total execution time is 1.858106 seconds as shown in Table 6. Although it is relatively slow compared with the other existing algorithms, this provides more security to the input color images. The space complexity of this algorithm is analyzed as the size of the encrypted output is the same as the size of the input. So the space required to execute this algorithm depends on the size of the input image. If MXN is the size of the input image, then the space complexity is O (MXN). As we can see in the graphs obtained, the time taken in Multiple ECC is larger for the same number of characters input as compared to single encryption. This shows that the time complexity increases by encrypting ciphertext multiple times in the encryption technique which in turn enhances the security of the data.

Table 6. Comparing encryption and decryption time

Existing methods	Encryption time (seconds)	Decryption time (seconds)
Dawahdeh et.al., (2018)	1.26435	NA
Laiphrakpam and Khumanthem (2018)	0.085	NA
Bakr et al., (2018)	1.2359	NA
Proposed method	**1.858106**	**0.2917308**

The time taken to encrypt and decrypt the input is presented in Table 6. When compared to other existing methods, the taken to encrypt for the proposed method is much higher. But this method gives better security measures like PSNR, MSE, SSIM, NPCR, UACI, information entropy, and correlation coefficient values. So, this method can be considered for applications to enhance the security of information.

Comparison With Other Algorithms

Some of the basic cryptographic algorithms are analyzed for security measures. A combination of these is also analyzed. These are presented in Table 7. The proposed method is compared with other basic algorithms. Proposed method results with lesser PSNR (8.4547), more MSE (9.3440e+03), lesser SSIM (0.0094) nearer to zero, higher NPCR (99.9919), ideal UACI (33.4635), slightly higher information entropy (7.9970) nearer to 8, when compared with other basic algorithms.

Table 7. Comparing the proposed method with other basic cryptographic algorithms, used San Diego.tiff as input image

Basic algorithms	PSNR (dB)	MSE	SSIM	NPCR (%)	UACI (%)	Information entropy
Logistic chaotic map	12.3845	3.7552e+03	0.0421	100	33.4635	7.1866
Arnold Transform	12.0063	4.0969e+03	0.0217	97.1375	33.4635	7.3545
Hyperchaotic map	8.7875	8.5959e+03	0.0088	99.5941	33.4635	7.9971
Arnold & logistic map	12.3845	3.7552e+03	0.0421	99.3469	33.4635	7.1866
Arnold & hyperchaotic	8.7875	8.6001e+03	0.0117	99.5880	33.4635	7.9970
Proposed Method	**8.4547**	**9.3440e+03**	**0.0094**	**99.9919**	**33.4635**	**7.9970**

CONCLUSION

In this proposed approach, the ECC, SHA-256, and hyperchaos algorithms are combined to strengthen and intensify the security of color images. SHA-256 is applied on input to produce key while using ECC. Only the elliptic curve point is shared instead of the costly operation of the mapping of a common lookup table. Hyperchaos and Arnold's transformation is used to generate the cipher image and their inverses are used at the decryption stage to extract back the original color image. Chaotic sequence from the hyperchaotic system shows that the algorithm gives better statistical measures. The security analysis for cipher image is carried out by measuring PSNR (8.4958), MSE (9.7655e+03), SSIM (0.0089), information entropy (7.9972), NPCR (98.8317), UACI (33.4635), and correlation coefficient (lesser value). The time and space complexity is also analyzed. The encryption/decryption time is much higher. Also analyzed with other basic cryptographic algorithms. The proposed approach gives good results for all the performance metrics when compared to other methods (Laiphrakpam and Khumanthem, 2018; Dawahdeh et.al., 2018; Bakr et al., 2018). This approach is progressively secure and beneficial for applications of scrambling images. In the future, the work can be extended on attack analysis.

REFERENCES

Abdoun, N., El Assad, S., Taha, M. A., Assaf, R., Deforges, O., & Khalil, M. (2016, June). Secure hash algorithm based on efficient chaotic neural network. In *2016 International Conference on Communications (COMM)* (pp. 405-410). IEEE. doi:10.1109/ICComm.2016.7528304

Bakr, M. A. E. H., Mokhtar, M. A., & Takieldeen, A. E. S. (2018). Elliptic curve cryptography modified Hill Cipher dependent on circulant matrix. *International Journal of Industrial Electronics and Electrical Engineering*, *6*(1), 24–29.

. Chaitanaya, G., Keerthi, B., Saleem, A., Trinadh Rao, A., & Kumar, K.T. P. S. (2015). An Image Encryption and Decryption using Chaos Algorithm. *IOSR Journal of Electronics and Communication Engineering*, *10*(2), 103-108.

Dawahdeh, Z. E., Yaakob, S. N., & Razif bin Othman, R. (2018). A new image encryption technique combining Elliptic Curve Cryptosystem with Hill Cipher. *Journal of King Saud University-Computer and Information Sciences*, *30*(3), 349–355. doi:10.1016/j.jksuci.2017.06.004

Gauravaram, P. (2012, November). Security Analysis of salt|| password Hashes. In *2012 International Conference on Advanced Computer Science Applications and Technologies (ACSAT)* (pp. 25-30). IEEE. doi:10.1109/ACSAT.2012.49

Gutub, A. A. A., Ibrahim, M. K., & Al-Somani, T. F. (2007, February). Parallelizing GF (P) elliptic curve cryptography computations for security and speed. In *2007 9th International Symposium on Signal Processing and Its Applications* (pp. 1-4). IEEE.

Hassene, S., & Eddine, M. N. (2016, March). A new hybrid encryption technique permuting text and image based on hyperchaotic system. In *2016 2nd International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)* (pp. 63-68). IEEE. doi:10.1109/ATSIP.2016.7523060

Huang, L., Cai, S., Xiao, M., & Xiong, X. (2018). A simple chaotic map-based image encryption system using both plaintext related permutation and diffusion. *Entropy (Basel, Switzerland)*, *20*(7), 535. doi:10.3390/e20070535 PMID:33265624

Ibrahim, R. K., Kadhim, R. A. J., & Alkhalid, S. S. H. (2015). Incorporating SHA-2 256 with OFB to realize a novel encryption method. *2015 World Symposium on Computer Networks and Information Security (WSCNIS)*, 1-6. doi:10.1109/WSCNIS.2015.7368295

Image Manipulation Dataset Website. (n.d.). https://www5.cs.fau.de/research/data/image-manipulation/orig

Kamalakannan, V., & Tamilselvan, S. (2015). Security enhancement of text message based on matrix approach using elliptical curve cryptosystem. *Procedia Materials Science*, *10*, 489–496. doi:10.1016/j.mspro.2015.06.086

Kaur, S., & Talwar, R. (2017). Arnold transform based Security Enhancement using Digital Image Watermarking with Complex Wavelet Transform. *International Journal of Electronics Engineering Research*, *9*, 677–693.

Koblitz, N. (1987). Elliptic curve cryptosystems. *Mathematics of Computation*, *48*(177), 203–209. doi:10.1090/S0025-5718-1987-0866109-5

KODAK Image Dataset Website. (n.d.). http://r0k.us/graphics/kodak

Laiphrakpam, D. S., & Khumanthem, M. S. (2018). A robust image encryption scheme based on chaotic system and elliptic curve over finite field. *Multimedia Tools and Applications*, *77*(7), 8629–8652. doi:10.1007/s11042-017-4755-1

Li, C., Zhao, F., Liu, C., Lei, L., & Zhang, J. (2019). A hyperchaotic color image encryption algorithm and security analysis. *Security and Communication Networks*, *2019*, 2019. doi:10.1155/2019/8132547

Miller, V. S. (1985, August). Use of elliptic curves in cryptography. In *Conference on the theory and application of cryptographic techniques* (pp. 417-426). Springer.

Min, L., Ting, L., & Yu-jie, H. (2013, November). Arnold transform based image scrambling method. In *3rd International Conference on Multimedia Technology (ICMT-13)* (pp. 1302-1309). Atlantis Press.

Qi, T., Jun-min, J., & Jun-li, J. (2016, August). An image encryption algorithm based on high-dimensional chaotic systems. In *2016 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)* (pp. 1-4). IEEE.

Roy, M., Deb, N., & Kumar, A. J. (2014). Point generation and base point selection in ECC: An overview. *International Journal of Advanced Research in Computer and Communication Engineering*, *3*(5), 6711–6713.

Salagundi, U., & Chheda, N., & Kiran. (2016, May). Image Encryption using Scrambling and Diffusion Operation Using Chaotic Map. *International Journal of Computer Science and Mobile Computing*, *5*(5), 343–348.

Sathish, Prasad, Tejaswi, Swapna, & Vijayarajan. (2019). Image scrambling through two-level Arnold transform. *Alliance International Conference on Artificial Intelligence and Machine Learning (AICAAM)*, 329-337.

Sathishkumar, G. A., Dr. Bhoopathy Bagan, K., & Dr. Sriraam, N. (2011, March). Image encryption based on diffusion and multiple chaotic maps. *International Journal of Network Security & Its Applications*, *3*(2), 181–194. doi:10.5121/ijnsa.2011.3214

Seyedzade, S. M., Mirzakuchaki, S., & Atani, R. E. (2010, October). A novel image encryption algorithm based on hash function. In *2010 6th Iranian Conference on Machine Vision and Image Processing* (pp. 1-6). IEEE. doi:10.1109/IranianMVIP.2010.5941167

Singh, C., Pandey, B. K., Mandoria, D. R. H. L., & Kumar, A. (2018). A Review Paper on Chaotic Map Image Encryption Techniques. *International Research Journal of Engineering and Technology*.

Slimane, N. B., Bouallegue, K., & Machhout, M. (2016). Nested chaotic image encryption scheme using two-diffusion process and the Secure Hash Algorithm SHA-1. *2016 4th International Conference on Control Engineering & Information Technology (CEIT)*, 1-5.

Ganavi M. is working as an Assistant Professor in the Department of Computer Science & Engineering (CSE) at Jawaharlal Nehru New College of Engineering (JNNCE), Shivamogga, Karnataka, India. She is pursuing Ph.D. in Computer Science and Engineering from the Department of CSE, JNNCE, Shivamogga, affiliated to Visvesvaraya Technological University (VTU), Belagavi-590018, Karnataka, India. Her research interests include Cryptography and information security.

Prabhudeva S. is working as a Professor and Director in the department of Master of Computer Applications (MCA) at Jawaharlal Nehru New College of Engineering (JNNCE), Shivamogga, Karnataka, India. He has received his Ph.D. degree in Reliability Engineering from IIT Bombay, India in 2010. Presently three research scholars are pursuing Ph.D. under his guidance. His research interests include Reliable and Security Modelling. He has published 18 papers in international journals and conferences. He has 17 years of research experience.