Algorithms for Encrypting Images using SEE Transformation

[1]Mohamed Yahay, [2]Najlae Falah Hameed Al Saffar

[1]Department of Mathematics,
[1]Faculty of Computer Science & Mathematics, University of Kufa. Iraq

***Abstract*—Nowadays, images play a big part in many applications, thus it's important to safeguard sensitive data from unwanted access or even manipulation. One of the finest ways to accomplish this goal is through encryption. Researchers have developed numerous methods for image encryption in recent years. To boost security, they employ different concepts to be tools for image encryption. In this work SEE transformation will be a tool to encrypt images for the 1st proposed algorithm. RSA public key cryptosystem will be involved together with SEE transformation for the 2nd proposed algorithms in two cases to make the transform of image is more efficient. Mean square error, peak signal to noise ratio and unified average changing intensity were used to test the proposed algorithms, demonstrating their speed and high level of security.**

***Index Terms*—Image Encryption, SEE Transformation, RSA Public Key Cryptography, MSE, PSNR and UACI.**

I. Introduction (Heading 1)

Cryptography is a tool to store and transmit information in a validated structure, in order for a client to read and process it in secret ways. It includes encryption and decryption of messages. The most common way of changing over plain information into cipher is known as encryption, while the method involved with recuperating the plain information from encoded one is known as decryption [1]. Cryptography has a wide range of applications, protecting the images is one of them. In fact, transmission of images across open networks is a necessary activity. Consequently, it is crucial to share confidential information in a secure manner. Different tools were used to increase the level of security such like [2] with two proposed algorithms and [3].

In this work See transformation [4] will be involved to be a tool to increasing the level of security for transform in image. For the 1st proposed algorithm. In the rest algorithms, RSA [5] public key cryptosystem will be used to make the transform of image is more efficient.

SEE transform [4] was created in 2021, which takes its name from Sadiq, Emad and Eman, is an integral transform that solved linear ordinary and partial differential equations with constant coefficients. In the same year, the same authors [6] derived the formula for the complex SEE and applied it in solving Laplace transformation problem. Also they trying to solve linear system of ordinary differential equations in the time domain [7]. In 2021 and 2022 the same authors [8] and [9] used SEE and complex SEE transformation as a tool for public key cryptosystem and applied them in text encryption and image encryption respectively. This why this work will focus on using SEE transform to be a tool to encrypt and decrypt images.

In order to offer confidentiality for transforming images, this work will introduce image encryption algorithms using the properties of SEE transform together with public key cryptosystem. Mean square error (MSE) Peak Signal to Noise Ratio (PSNR) and Unified Average Changing Intensity (UACI) security measures will be used to gauge how effectively these algorithms perform.

II. SEE TRANSFORM

Sadiq, Emad and Eman (SEE) [4] are derived from the Laplace transformation. SEE integral was investigated to work differential conditions in the time area. It is change the characterized for capacity of outstanding request, where the capacities in the set A characterized by the following:

$A = \{F(T)$: there exist $M, l_1, l_2 > 0. |f(t)| < M\, e^{|t|\, l_i}$, if $t \in (-1)^i \times [0, \infty)\}$ … (1)

For a given capacity in the arrangement of A, the constant M have to be bounded, while l_1 and l_2 need not to be might be limited. SEE transformation has been defined by the following:

$S\big(f(t)\big) = T(v) = \frac{1}{v^n}\int_0^\infty f(t)e^{-vt}dt = T(s)\ t \geq 0, n \in z^{\dot{+}}, l_1 \leq v \leq l_2$ … (2)

The variable v in this definition is used to aspect the variable t. Invers of SEE transformation is defined to be $S^{-1}\big(T(v)\big) = f(t)$.

III. RSA PUBLIC KEY CRYPTOSYSTEM

In 1976 Diffie and Hellman [10] have been introduced any new type of cryptography which opened the door for the more significant invention, it was the RSA public key cryptosystem by Rivest, Shamir, and Adleman [5] in 1978. An explanation of this system will be brief within three levels: Key Generation, Encryption and Decryption level for any two users A and B.

First Level: Key Creation (User B)

- Select secret primes p and q.
- Select public key e with $\gcd\big(e, (p-1)(q-1)\big) = 1$.
- Publish $N = p \cdot q$ and e.

Second Level: Encryption (User A)

- Select plaintext M.
- Calculate $C \equiv M^e \bmod N$.
- Send cipher text C to B.

Third level: Decryption (User B)

- Calculate d where $ed \equiv 1 \bmod (p-1)(q-1)$.

- Calculate the plain text $M \equiv C^d \bmod N$.

IV. Proposed Algorithms

As mentioned before a wider range of areas, including mathematics, physics, and electrical engineering, use the SEE transform. In this work, two SEE transform-based encryption and decryption images are proposed. The further usage of these algorithms is carried forward in MATLAB R2020a (8.3.0.532) 64–bit software on a work station Intel® Core™ i7- 10510U with CPU 1.80 GHz and RAM 8 GB with Microsoft Windows 11.

First Algorithm

Suppose that two users A and B went use this algorithm to a message, they have pass those levels:

First Level: Encryption Steps: (A)

Step 1: choose the image that need to encrypt.
Step 2: let the G_i represent pixels image, where i is the length of image.
Step 3: use the trigonometric function Taylor series of $cos\ x$,

$$\cos x = 1 + \frac{x^2}{2} - \frac{x^4}{4!} + \cdots = \sum_0^\infty \frac{(-1)^n * x^n}{2n!}. \quad (1)$$

Step 4: multiplication G_i by Taylor trigonometric function as:

$$G_i \cos x = G_0 1 + \frac{G_1 x^2}{2!} - \frac{G_2 x^4}{4!} + \cdots = \sum_0^\infty \frac{(-1)^n * G_i x^n}{2n!}. \quad (2)$$

Step 5: taking SEE transform for (2) as:

$$S\{G_i \cos x\} = S\left\{G_0 1 + \frac{G_1 x^2}{2!} - \frac{G_2 x^4}{4!} + \cdots\right\} = S\left\{\sum_0^\infty \frac{(-1)^n * G_i x^n}{2n!}\right\} = \frac{G_0}{s^2} + \frac{G_1 x^2 * 2!}{s^4} - \frac{G_2 x^4 * 4!}{s^6} + \cdots = \sum_0^\infty \frac{(-1)^n * G_i x^n * 2n!}{s^{2n}}.$$

(3)

Step 6: divide all numbers in (3) by 256 and store the reminder as matrix.
Step 7: the output matrix will be the encryption image.

Second Level: Key generation: (User A)

Consider

$$x_i = 256\, k_i + r_i \quad (4)$$

where x_i are the number of the result of (3), r_i are the reminder when divided x_i by 256 , k_i are the number of key that need to generate and i is the length of image. So, there are $k_0, k_1, \ldots, k_i$ of keys according to solving the following system:

$x_0 = 256\, k_0 + r_0$
$x_1 = 256\, k_1 + r_1$
.
.
.
$x_i = 256\, k_i + r_i$

Third Level: Decryption Steps: (B)

Step 1: converting the encrypted image to the numbers.
Step 2: let the G_i' represent these numbers image and i the length of image.
Step 3: solve the following system:

$y_0 = 256\, k_0 + G_0'$
$y_1 = 256\, k_1 + G_1'$
.
.
.
$y_i = 256\, k_i + G_i'$

To get the values of $y_0, y_1, \ldots, y_i$

Step 4: calculating the invers of SEE of y_i multiply by trigonometric function Taylor series of $cos\ x$:

$$S^{-1}\left\{\frac{y_0}{s^2} + \frac{y_1 x^2 * 2!}{s^4} - \frac{y_2 x^4 * 4!}{s^6} + \cdots\right\} = S^{-1}\left\{\sum_0^\infty \frac{(-1)^n * y_i x^n * 2n!}{s^{2n}}\right\} = \left\{y_0 + \frac{y_1 x^2 * 2!}{s^4} - \frac{y_2 x^4 * 4!}{s^6} + \cdots\right\} = \sum_0^\infty \frac{(-1)^n * y_i x^n}{2n!}.$$

Step 5: store the numerator integer value as matrix.
Step 6: the output matrix will be the plain image

Second and Third Algorithm

Making a combination of the RSA algorithm and the SEE transform is an efficient approach to encrypt the image which is a scheme that has been introduced in this section with two cases, where the RSA will be involved to encrypt the ciphered image for the 1st case while the 2nd case, the RSA will be implemented for the plain image.

V. Implementations

In this section, the results that have been gotten appeared the significance of the proposed algorithms. Lena image of size 256 × 256 is considered a plain image.

First Algorithm: A wants to send a Lena image to B

First Level: Encryption Steps: (A)

Choose "Lena" as plain image:

She separates the pixel values of "Lena 256 × 256" into blocks of size four as:

	1	2	3	4	5	6	7	8	…	256
1	250	255	255	251	249	255	255	255	…	
2	250	255	255	251	250	255	255	254	…	
3	251	255	255	251	251	255	255	249	…	
4	252	255	255	251	253	255	253	242	…	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮		⋮
256									…	65536

Converting a block to a single row from $i = 1$ to 65536 for example we choose the vertex of i from 1 to 8:

	1	2	3	4	5	6	7	8	…	65536
1	250	250	251	252	253	254	255	255	…	

Taylor for $\cos x = 1 + \frac{x^2}{2} - \frac{x^4}{4!} + \cdots = \sum_0^\infty \frac{(-1)^n * x^n}{2n!} = G_0 * 1 + \frac{G_1 x^2}{2!} - \frac{G_2 x^4}{4!} \ldots = \sum_0^\infty \frac{(-1)^n * G_i x^n}{2n!}$,

$\{G_0 = 250, G_1 = 250, G_2 = 251, G_3 = 252, G_4 = 253, G_5 = 254, G_6 = 255, G_7 = 255 \ldots,$

$\{250 + \frac{250x^2}{2!} - \frac{251x^4}{4!} + \frac{252x^6}{6!} - \frac{253x^8}{8!} + \frac{254x^{10}}{10!} - \frac{255x^{12}}{12!} + \frac{255x^{14}}{14!} - \cdots.$

Taking SEE transform from the series

$S\{G_i \cos x\} = S\{250 + \frac{250x^2}{2!} - \frac{252x^4}{4!} + \frac{253x^6}{6!} - \frac{254x^8}{8!} + \frac{255x^{10}}{10!} - \frac{255x^{12}}{12!} + \cdots\}$

$= \{250 * \frac{1}{v} + \frac{250*2!*2!}{v^3} - \frac{251*4!*4!}{v^5} + \frac{253*6!*6!}{v^7} - \frac{254*8!*8!}{v^9} + \frac{255*10!*10!}{v^{11}} - \frac{255*12!*12!}{v^{13}} + \frac{255*14!*14!}{v\text{^}15} + \ldots,$

$= \{250 * \frac{1}{v^2} + \frac{1000}{v^4} - \frac{144576}{v^6} + \frac{130636800}{v^8} - \frac{411302707200}{v^{10}} + \cdots\},$

Taking mod 256 for the numbers in the numerators

	1	2	3	4	5	6	7	8	…	65536
1	250	232	192	0	0	0	0	0	…	

The output is the Encryption image. Reset the block to 256*256

	1	2	3	4	5	6	7	8		256
1	250	0	0	0	0	0	220	0	…	
2	232	0	0	0	0	0	192	0	…	
3	192	0	0	0	0	0	0	0	…	
4	0	0	0	0	0	0	0	0	…	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱	
256										

Second Level: Key generation: (A)

A will do key generation steps for $i = 1$ to 65536

$250 = 256\, k_0 + 250, k_0 = 0,$

$1000 = 256\, k_1 + 232, k_1 = 3,$

$144576 = 256\, k_2 + 192, k_2 = 564,$

$130636800 = 256\, k_3 + 0, k_3 = 510300,$

$411302707200 = 256\, k_4 + 0, k_4 = 1606651200,$

$\vdots \quad \vdots \quad \vdots \quad \vdots,$

So the keys are $\{0, 3, 564, 510300, 1606651200, \ldots\}$.

Third Level: Decryption Steps: (B)

Converting the encrypted image to matrix.

Let the G_i' represent these numbers image and i the length of image.

Solve the following system:

$y_i = 256\, k_i + G_i',$

$y_0 = 256 * 0 + 250 = 250,$

$y_1 = 256 * 3 + 232 = 1000,$

$y_2 = 256 * 564 + 192 = 144576,$

$y_3 = 256 * 510300 + 0 = 130636800,$

$y_4 = 256 * 1606651200 + 0 = 411302707200,$

$\vdots \quad \vdots \quad \vdots \quad \vdots$

Calculating the invers of SEE of y_i multiply by trigonometric function Taylor series of $cos\ x$:

$$S^{-1}\left\{250 * \frac{1}{v^2} + \frac{1000}{v^4} - \frac{144576}{v^6} + \frac{130636800}{v^8} - \frac{411302707200}{v^{10}} + \cdots\right\},$$

$$= \left\{250 + \frac{250x^2}{2!} - \frac{251x^4}{4!} + \frac{252x^6}{6!} - \frac{253x^8}{8!} + \frac{254x^{10}}{10!} - \frac{255x^{12}}{12!} + \frac{255x^{14}}{14!} - \cdots\right\},$$

$$= \{G_0 = 250, G_1 = 250, G_2 = 251, G_3 = 252, G_4 = 253, G_5 = 254, G_6 = 255, G_7 = 255 \ldots\},$$

The block of output:

	1	2	3	4	5	6	7	8	…	65536
1	250	250	251	252	253	254	255	255	…	

Reset the block to 256*256:

	1	2	3	4	5	6	7	8	…	256
1	250	255	255	251	249	255	255	25	…	
2	250	255	255	251	250	255	255	254	…	
3	251	255	255	251	251	255	255	249	…	
4	252	255	255	251	253	255	253	242	…	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱	
256										

Fig 1. will summarize the output of execution of this implementation.

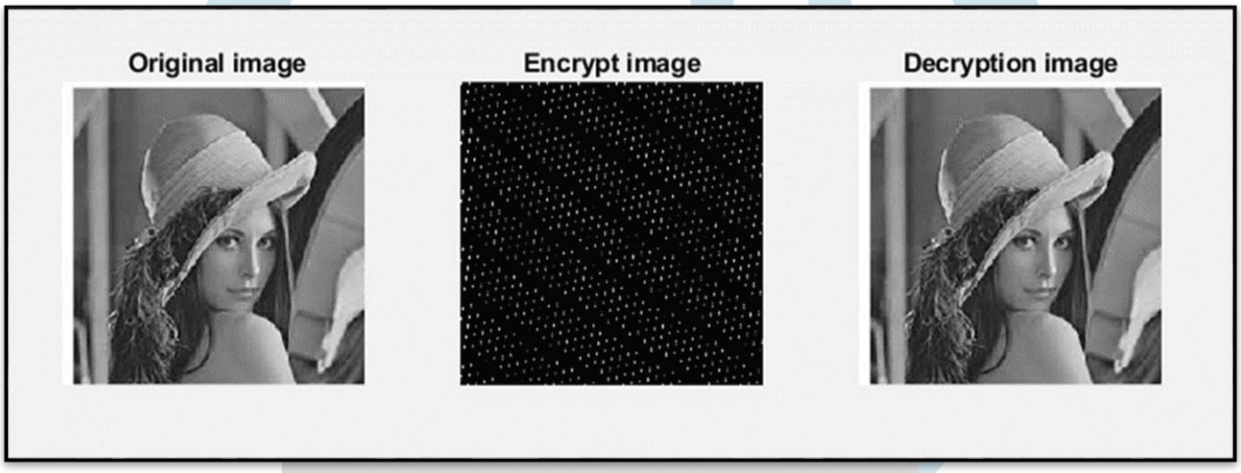

Figure 1: Implementation the Lena (256×256) image for the 1st proposed algorithm

Second Algorithm: A wants to send a Lena image to B
First Case: Encryption by SEE and RSA

A separates the pixel values of "Lena 256× 256" into blocks of size four as:

	1	2	3	4	5	6	7	8	…	256
1	250	255	255	251	249	255	255	255	…	
2	250	255	255	251	250	255	255	254	…	
3	251	255	255	251	251	255	255	249	…	
4	252	255	255	251	253	255	253	242	…	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱	
256										

Encode images as in the 1st proposed algorithm using SEE:

	1	2	3	4	5	6	7	8	…	256
1	250	0	0	0	0	0	220	0	…	
2	232	0	0	0	0	0	192	0	…	
3	192	0	0	0	0	0	0	0	…	
4	0	0	0	0	0	0	0	0	…	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱	
256										

Then she will encrypt the output with RSA, she will choose prime numbers whose product must be greater than 256, for example 17 and 19 whose product is 323:

1	2	3	4	5	6	7	8	…	256

1	175	0	0	0	0	0	0	0	…
2	175	0	0	0	0	0	0	0	…
3	233	0	0	0	0	0	0	0	…
4	40	0	0	0	0	0	0	0	…
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱
256									

A will generate keys as:

$250 = 256\, k_0 + 250, k_0 = 0,$
$1000 = 256\, k_1 + 232, k_1 = 3,$
$144576 = 256\, k_2 + 192, k_2 = 564,$
$130636800 = 256\, k_3 + 0, k_3 = 510300,$
$411302707200 = 256\, k_4 + 0, k_4 = 1606651200,$
$\vdots \quad \vdots \quad \vdots \quad \vdots,$

Now, B will receive the output, then decrypt it according to RSA algorithm:

	1	2	3	4	5	6	7	8	…	256
1	250	0	0	0	0	0	220	0	…	
2	232	0	0	0	0	0	192	0	…	
3	192	0	0	0	0	0	0	0	…	
4	0	0	0	0	0	0	0	0	…	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱	
256										

As in the 1st algorithm:

	1	2	3	4	5	6	7	8	…	256
1	250	255	255	251	249	255	255	255	…	
2	250	255	255	251	250	255	255	254	…	
3	251	255	255	251	251	255	255	249	…	
4	252	255	255	251	253	255	253	242	…	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱	
256										

Fig 2. will summarize the output of execution of this implementation.

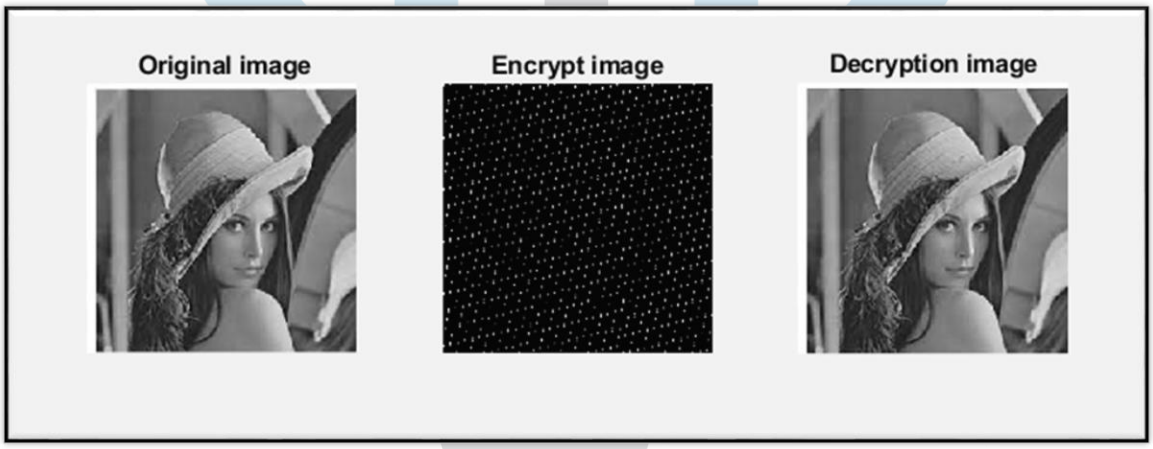

Figure 2: Implementation the Lina (256×256) Image for the 2nd Proposed Algorithm/ 1st case

Second Algorithm: A wants to send a Lena image to B
Second Case: Encryption by RSA and SEE
Firstly, A will encrypt image using RSA and then encryption using SEE:

	1	2	3	4	5	6	7	8	…	256
1	250	255	255	251	249	255	255	255	…	
2	250	255	255	251	250	255	255	254	…	
3	251	255	255	251	251	255	255	249	…	
4	252	255	255	251	253	255	253	242	…	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱	
256										

She chooses two prime 17 and 19 for RSA, the result will be as follows:

	1	2	3	4	5	6	7	8	…	256
1	300	221	221	55	260	221	221	221	…	
2	300	221	221	55	300	221	221	220	…	
3	55	221	221	55	55	221	221	260	…	
4	313	221	221	55	43	221	43	200	…	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱	
256										

Then encrypt the output of RSA using SEE. According to the 1st proposed algorithm:

	1	2	3	4	5	6	7	8	…	256
1	300	283	97	39	36	31	204	296	…	
2	231	198	66	273	230	244	204	133	…	
3	26	170	0	6	197	97	34	174	…	
4	150	62	111	236	28	147	138	230	…	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱	
256										

Key generation: (A)
As a generation key steps: as the 1st proposed algorithm, but here the difference is that the mod is the value of n=323.
Decryption: (B)
As in the 1st proposed algorithm from SEE:

	1	2	3	4	5	6	7	8	…	256
1	300	221	221	55	260	221	221	221	…	
2	300	221	221	55	300	221	221	220	…	
3	55	221	221	55	55	221	221	260	…	
4	313	221	221	55	43	221	43	200	…	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱	
256										

Appling RSA algorithm for decryption as:

	1	2	3	4	5	6	7	8	…	256
1	250	255	255	251	249	255	255	255	…	
2	250	255	255	251	250	255	255	254	…	
3	251	255	255	251	251	255	255	249	…	
4	252	255	255	251	253	255	253	242	…	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱	
256										

Fig 3. will summarize the output of execution of this implementation.

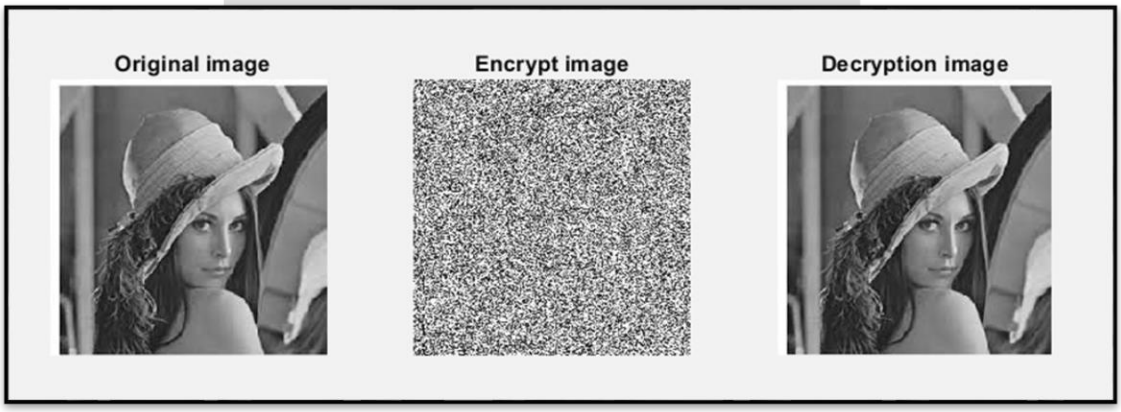

Figure 3. Implementation the Lena (256×256) Image for the 2nd Proposed Algorithm/ 2nd Case

SECURITY EXAMINATION

There are three measures used to assess encrypted images and compare them with plain images to calculate the performance of the proposed algorithms.

Mean Square Error and Peak Signal-to-Noise Ratio: Mean square error (MSE) and peak signal-to-noise ratio (PSNR) are efficient tools to measure the efficiency of image encryption algorithms. So, employing MSE and PSNR is a good way to evaluate the quality of plain images with respect to the encrypted images. The equation of the PSNR is as: $\mathrm{PSNR} = 10\log_{10}\frac{255\cdot 255}{\mathrm{MSE}}$ where $\mathrm{MSE} = \frac{1}{m*n}\sum_{i=1}^{m}\sum_{j=1}^{n}(X(i,j) - Y(i,j))^2$, $X(i,j)$ and $Y(i,j)$ are the pixel value of plain image and encrypted image respectively both have size $m \times n$. Obviously, that calculating the MSE will affect the value of PSNR, if PSNR is decreasing means that the MSE is increasing, in other words the high value of MSE and low value of PSNR indicates that the two images are totally different, and this leads to the efficiency of the proposed algorithms.
In this work, the upsides of the MSE were 242.0946, 242.0946 and 82.7409 for 1st proposed algorithm, 2nd proposed algorithm/ 1st Case and 2nd proposed algorithm/ 2nd Case respectively, while PSNR were 24.2910, 24.2910 and 28.9536 for 1st proposed algorithm, 2nd proposed algorithm/ 1st Case and 2nd proposed algorithm/ 2nd Case respectively for the Lena as a plain image, it is so difficult for an aggressor to recuperate the plain image.

Unified Average Changing Intensity (UACI): It is one of differential investigations used to assess the strength of image encryption, where it is assessed the differentiation between the encrypted image and plain image [3], [9]. The highest value of the UACI implies that the proposed algorithms are safe against differential assaults. The condition of the UACI is as: : $\mathrm{UACI} = \frac{1}{256*256}\sum_{i=1}^{256}\sum_{j=1}^{256}\frac{X(i,j)-Y(i,j)}{255}\cdot 100\%$, $X(i,j)$ and $Y(i,j)$ are the pixel value of plain image and encrypted image respectively.

In this work, the worth of the UACI were 42.2457, 42.2457 and 9.8888 for 1st proposed algorithm, 2nd proposed algorithm/ 1st Case and 2nd proposed algorithm/ 2nd Case respectively of the encrypted image with respect to Lena as a plain image, it is so hard for an attacker to recover the plain image.

Table 1 The performance of the proposed algorithms using various security measures for different image with size 256×256

Plain image Name	Measures	1st Algorithm	2nd Algorithm/ 1st case	2nd Algorithm/ 2nd case
Lena	MSE	242.0946	242.0946	82.7409
	PSNR	24.2910	24.2910	28.9536
	UACI	42.2457	42.2457	9.8888
Peppers	MSE	244.3767	244.3767	110.0144
	PSNR	24.2502	24.2502	27.7163
	UACI	55.8598	55.8598	14.8896
Cameraman	MSE	226.9854	226.9854	87.0020
	PSNR	24.5708	24.5708	28.7355
	UACI	44.0755	44.0755	10.9918
Baboon	MSE	243.4494	243.4494	94.0968
	PSNR	24.2667	24.2667	28.3951
	UACI	48.2092	48.2092	11.1703

Table 2 Comparison of the proposed algorithm with other methods over security measures for different image with size 256×256

		Lena			**peppers**			**Cameraman**			**Baboon**		
		MSE	**PSNR**	**UACI**	**MSE**	**PSNR**	**UACI**	**MSE**	**PSNR**	**UACI**	**MSE**	**PSNR**	**UACI**
This Study	**1st Proposed Algorithm**	242.0946	24.2910	42.2457	244.3767	24.2502	55.8598	226.9854	24.5708	44.0755	243.4494	24.2667	48.2092
	2nd Proposed Algorithm/ 1st Case	242.0946	24.2910	42.2457	244.3767	24.2502	55.8598	226.9854	24.5708	44.0755	243.4494	24.2667	48.2092
	2nd Proposed Algorithm/ 2nd Case	82.7409	28.9536	9.888	110.0144	27.7163	14.8896	87.0020	28.7355	10.9918	94.0968	28.3951	11.17030.
Zahr	**1st Case**	…	9.2996	28.1854	…	…	…	…	9.0097	26.9897	…	9.8015	27.0906

	2nd Case	…	9.2549	28.529	…	…	…	…	8.0926	32.092	…	9.7765	27.188
Najlae [3]		12002.3	7.36791	33.47067	12021.9	7.343033	33.398	13268.97	6.903213	33.79933	7815.133	9.221403	32.47023
Rajvir [11]		11015.636	7.7107	33.5844	11720.228	7.4414	346475	…	…	…	6949.448	9.7113	27.3642
Zahraa [12]		7640.4	9.2996	28.1854	…	…	…	8167.8	9.0097	26.9897	6806.6	9.8015	27.0906

Table 3 Encryption and Decryption time for the proposed scheme for different gray image with size 256×256.

Image Name	Encryption/ and Decryption time in seconds
Lena	7.697
Peppers	20.249s
Cameraman	10.939s
Mandrill	20.391s

According to Table 1, the values of PSNR were very low while the values of MSE were very high, which means that the proposed scheme is an efficient algorithm. The values of UACI for the selected images were almost close to the expected value of 33.46, and this demonstrates the strength of the proposed algorithms against the attackers. These results actually showed that the encrypted images were barely perceptible, which serves as the foundation for a strong encryption technique.

In Table 2, a comparison of the proposed method and a few other recent algorithms regarding security precautions for various gray images with size 256×256 is presented. Thus, it can be said that the proposed algorithms is more effective than the alternatives.

Finally, table 3 summarized the time consumed in encryption and decryption processes over the implementing images considered in this work. The time it takes for each step of the proposed algorithms to complete is used to gauge how effective they are; table 3 demonstrates how quickly the proposed scheme can be implemented.

REFERENCES

[1] Tayal, Sandeep, et al. "A Review paper on Network Security and Cryptography," Advances in Computational Sciences and Technology. 2017; 10(5):763–770.

[2] Z. K. Obaid, N. F. H. Al Saffar, "Image encryption based on elliptic curve cryptosystem," International Journal of Electrical and Computer Engineering. 2021; 7(2): 1293-1302.

[3] N. F. H. Al Saffar, I. R. Al-Saiq, R. R. M. Abo Alsabeh, "Asymmetric image encryption scheme based on Massey Omura scheme," International Journal of Electrical and Computer Engineering. 2022; 12(1): 1040-1047.

[4] Eman A,et al. The new integral transform "SEE transform" and its applications. Periodicals of Engineering and Natural Sciences. 2021; 9(2): 1016-1029.

[5] Rivest, R. L., Shamir, A. and Adleman, L., "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," Communications of the ACM 21 (2): 120–126, 1978.

[6] Eman A.Mansour et al. Application of New Transform "Complex SEE Transform" to Partial Differential Equations, Journal of Physics: Conference Series, 2021 J. Phys.: Conf. Ser. 1999 012155.

[7] Eman A.Mansour et al. "On the SEE transform and systems of ordinary differential equations," Periodicals of Engineering and Natural Sciences. 2021; 9(3): 277-281.

[8] Eman A.Mansour et al. "Applying Complex SEE Transformation in Cryptography," MJPS. 2021; 8(2): 99-104.

[9] Emad A. Kuffi et al. "Color Image Encryption Based on New Integral Transform SEE," Journal of Physics: Conference Series, 2022 J. Phys.: Conf. Ser. 2322 012016.

[10] Diffie, W. and Hellman, M. E., "New Directions in Cryptography," IEEE Transactions on Information Theory IT-22 (6): 644–654, 1976.

[11] Rajvir, C., Satapathy, S., Rajkumar, S. and Ramanathan, L., "Image Encryption Using Modified Elliptic Curve Cryptography and Hill Cipher," Smart Intelligent Computing and Applications. 2020: 675–683.

[12] Z. K. Obaid, N. F. H. Al Saffar, "Image Encryption Based on Menezes Vanstone Elliptic Curve Cryptosystem," Solid State Technology. 2020; 63(3): 5256 – 5265.