Journal of Engineering and Applied Sciences 13 (21): 9194-9203, 2018
ISSN: 1816-949X

Image Encryption and Decryption via. (GSVD-Modular Numbers)

[1]Mohammed Abdul Hameed Jassim AL-Kufi, [2]Dheiaa Shakir Redhaa and [3]Mujtaba Zuhair Ali
[1]Department of Islamic Education, University of Kufa, 31001 Al-Najaf, Iraq
[2]Ministry of Education, Baghdad, Iraq
[3]Department of Computer Technical Engineering, College of Technical Engineering,
The Islamic University, Najaf, Iraq

Abstract: It is no hidden to everyone that the rapid scientific development which included the field of information security has led to the transformation of countries into electronic governments continuing to contact the latest scientific developments because if they did not do so or lag behind the scientific development, lost its immunity as a state of immunity and privacy because it simply became a state ruled by others. The most sensitive scientific bfield in this field is information security. In order for the departments and institutions that represent the state entity to maintain the confidentiality of their information, they must resort to the use of information encryption technology. Information encryption is a very large field and has evolved dramatically until encryption methods have become multi-methods after they have passed by many years, the most important and the most rapid and developed of them were in the last 10 years. In a previous study, the analysis technique GSVD was used in the image encryption and in more than one style. In this study, we will encrypt the image using analysis of GSVD with the technique of modular together in the image encryption to increase the complexity of the previous method to protect it from hackers. The research will be done through two step encryption. The first phase is the encryption using modular technique and the second stage is the encryption of the image encrypted in the first phase and that's by using GSVD technology what makes the method powerful and disobedient to the intruders and we will see the steps of encryption in both stages clearly and in detail later.

Key words: Encryption, SVD, GSVD, modular, image encryption, stages

INTRODUCTION

The assessment of nations and countries in this age which is subject to a huge scientific explosion is by the degree of their specificity (Wu and Rul'kov, 1993). There is no country with legal value without privacy (Sathishkumar *et al*., 2011). This privacy is at the forefront of the security issue of information including the image, texts and so on (Parker and Chua, 1987). The only means to be used to preserve this privacy is the process of encryption of the digital information which is one of the pillars of the strong state and this encryption process should not stop at the limit it must be adopted by a scientific team who always tries to develop it to prevent the intrusion of hackers taking into account the calculations of the time of encryption and decryption in line with the great development in this area, especially as we deal with a large amount of information and data (Macq and Quisquater, 2005; Yi *et al*., 2001; Yang *et al*., 2004).

The digital information that we need to be hidden and not to be acquainted but by the authorized, need to be undergone into a complex mathematical process and a certain key for the purpose of encryption and preventing non-authorized persons to know it and this process (encryption) is the best method used by humans to protect the digital information whether being maps messages, images, films or audio recordings (Al-Kufi *et al*., 2017; Wei *et al*., 2016). By this, the non-authorized won't be able to know them as being they who know the encryption algorithm and encryption key (Shah and Saxena, 2011). So, encryption can be defined as the process of hiding the digital info. in a reversible mathematical way (Hansen, 1989; Divya *et al*., 2012).

Also, encryption can be defined as being coding the data and transfer it into other numbers different than it used to be, so, it cannot be decrypted but by the authorized and these coded numbers are considered the result of the mathematical process conducted on the data. And the process of reversing the mathematical process

Corresponding Author: Mohammed Abdul Hameed Jassim AL-Kufi, Department of Islamic Education, University of Kufa, 31001 Al-Najaf, Iraq

to restart the original date is called decryption and that which can include some errors (Hansen, 1989; Divya *et al*., 2012).

It's not hidden to specialists that this field of info. security has passed by many accelerated developing phases as many techniques have arisen related to encryption of texts, maps, images and the digital process of images and other within this field. This development has been concerned with important factors such as accuracy, error and time of encryption and decryption (Hansen, 1989; Divya *et al*., 2012).

In recent times, matrix analyses have been used (in all its types) such as SVD and GSVD, alone or with other algebraic techniques such as modular or by using the two analyses together by promising researchers as Al-Kufi *et al*. (2017), Abdul-Hameed and Al-Kufi (2014) and Al-Rammahi and Al-Kufi (2016). These matrix analyses are flexible enough to be a successful factor in conducting processing, pressure or encryption (Hansen, 1989).

Science has accelerated rapidly so inventions have been developing day after day such as the mobile phone, the PDA and many other functions had innovated for these inventions and the activity of the digital interchange of info. increased through multi-media systems (Sathishkumar *et al*., 2011).

The distribution of multi-media systems service obliged us to adopt new techniques and methods to protect the digital info. and that's for the privacy of users through insuring and hiding them from hackers (Hansen, 1989; AL-Rammahi and Al-Kufi, 2016; Wu and Rul'kov, 1993).

For this purpose, we and after using the analysis method of SVD with modular (AL-Rammahi and Al-Kufi, 2016) will use the analysis of GSVD and modular with a development to the method which represents a compound encryption of an image as it'll be explained later. And we've called the encrypted images as Mk-carpet 9 and 10, respectively.

MATERIALS AND METHODS

GSVD and modular: These two subjects are from the algebraic subjects in mathematics and they've a high capacity of flexibility in processing numbers through their features and criteria and they're used in image field a lot such as image processing, encryption, pressure in addition to their use with texts and maps. And there're a lot of them in the resources of mathematics and algebraic references in all its types. For curtailment and to add a privacy on our paper's subject and to take care of the research ideas, we'll just refer to what is related to our study only.

If you need more details about these two subjects, you can have a look at the references available in this studies conclusion which is rich in information and detailed explanation.

GSVD: Algebra matrices have evolved steadily in all aspects, especially, matrix analysis. Among these analyzes is the analysis of GSVD which includes several types and considered the main element in many applications in the arithmetic of matrices in particular and the general applied matrices in general and in this study we show below a useful summary of this analysis within limits of the research need, so that, the details of the algorithm and the role of this analysis be clear (Al-Kufi *et al*., 2017; Wei *et al*., 2016; Hansen, 1989; Wu and Rul'kov, 1993; Kolman, 1984).

GSVD generalized singular value decomposition: Let A and B are tow matrix [U, V, X, C, S] = GSVD (A, B) returns unitary matrices U and V, a (usually) square matrix X and non negative diagonal matrices C and S, so that:

$$A = U * C * X^T$$

$$B = V * S * X^T$$

$$C'*C+S'*S = 1$$

A and B must have the same number of columns but may have different numbers of rows. If A is m-by-p and B is n-by-p, then U is m-by-m, V is n-by-n and X is p-by-q where, q = min (m+n, p).

For more information, you can go back to the source which contains other details that cannot be explained here in this manuscript which we want them to be concise and focused.

Modular: (MOD) Modulus after Division (AL-Rammahi and Al-Kufi, 2016; Hameed and Al-Kufi, 2018), mod(x, y) is x-n.*y where n = floor (x./y) if y~ = 0. If y is not an integer and the quotient x./y is within roundoff error of an integer, then n is that integer. The inputs x and y must be real arrays of the same size or real scalars. The statement "x and y are congruent mod m" means mod(x, m) = mod(y, m). By convention:

mod(x, o) is x

mod(x, x) is 0

mod(x, y) for x~ = y and y~ = 0 has the same sign as y. Note: REM (x, y) for x~ = y and y~ = 0 has the same sign as x. mod(x, y) and REM(x, y) are equal if x and y have the same sign but differ by y if x and y have different signs.

Methodology of proposed algorithm of encryption and decryption

Encryption: On the assumption that A is an image matrix: first, we define another image and make it a key to encryption and let B and another real number c. Encode the initial encoding in the modular way by which we generate two new arrays from the original image matrix A:

$$A1 = (A+B) \bmod (256)$$
$$A2 = (A+5*B) \bmod (256)$$
$$F = \begin{bmatrix} A1 \\ A2 \end{bmatrix}$$

It represents the image encoded by modular and is a white image:

$$A1 = -10 * c*A1$$
$$A2 = -c*A2$$

We use the second B key in the GSVD account for both A1 and A2 as it comes:

$$[u1, v1, x1, c1, s1] = gsvd(A1, B)$$
$$[u2, v2, x2, c2, s2] = gsvd(A2, B)$$

We recalculate the code matrices as follows:

$$AA1 = u1*c2*x1^T$$
$$AA2 = u2*c1*x2^T$$

We then build the encrypted image matrix as follows:

$$F = \begin{bmatrix} AA1 \\ AA2 \end{bmatrix}$$

It is a black image.

Decryption: We have the image encoded F and the encryption key image B. We divide the matrix F into two halves:

$$F = \begin{bmatrix} AA1 \\ AA2 \end{bmatrix}$$
$$[uu1, vv1, xx1, cc1, ss1] = gsvd(AA1, B)$$
$$[uu2, vv2, xx2, cc2, ss2] = gsvd(AA2, B)$$
$$A1_{new} = uu1*cc2*xx1^T$$
$$A2_{new} = uu2*cc1*xx2^T$$
$$A1_{new} = \frac{A1_{new}}{-10*c}$$
$$A1_{new} = (A1_{new} - B) \bmod (256)$$

The last matrix $A1_{new}$ represents the image matrix after decoding.

Clarification example: We have implemented the steps of this algorithm on an illustrative example using MATLAB (Knuth, 1997). Let:

$$a = \begin{bmatrix} 199 & 245 & 92 \\ 200 & 134 & 89 \end{bmatrix}$$
$$b = \begin{bmatrix} 222 & 111 & 59 \\ 231 & 108 & 2 \end{bmatrix} \text{ and } c = 109$$

Encryption:

$$a_1 = (a+b) \bmod (256)$$
$$= \left(\begin{bmatrix} 199 & 245 & 92 \\ 200 & 134 & 89 \end{bmatrix} + \begin{bmatrix} 222 & 111 & 59 \\ 231 & 108 & 2 \end{bmatrix} \right)_{mod(256)}$$
$$= \left(\begin{bmatrix} 421 & 356 & 151 \\ 431 & 242 & 91 \end{bmatrix} \right)_{mod(256)}$$
$$= \begin{bmatrix} 165 & 100 & 151 \\ 175 & 242 & 91 \end{bmatrix}$$

$$a_2 = (a+5*b) \bmod (256)$$
$$= \left(\begin{bmatrix} 199 & 245 & 92 \\ 200 & 134 & 89 \end{bmatrix} + 5* \begin{bmatrix} 222 & 111 & 59 \\ 231 & 108 & 2 \end{bmatrix} \right)_{mod(256)}$$
$$= \left(\begin{bmatrix} 1309 & 800 & 387 \\ 1355 & 674 & 99 \end{bmatrix} \right)_{mod(256)}$$
$$= \begin{bmatrix} 29 & 32 & 131 \\ 75 & 162 & 99 \end{bmatrix}$$

$$a_{11} = -10*c*a_1 = -10*109* \begin{bmatrix} 165 & 100 & 151 \\ 175 & 242 & 91 \end{bmatrix}$$
$$= \begin{bmatrix} -179850 & -109000 & -164590 \\ -190750 & -263780 & -99190 \end{bmatrix}$$

$$a_{22} = -c*a_2 = -109*\begin{bmatrix} 29 & 32 & 131 \\ 75 & 162 & 99 \end{bmatrix}$$

$$= \begin{bmatrix} -3161 & -3488 & -14279 \\ -8175 & -17658 & -10791 \end{bmatrix}$$

$$[u_1, v_1, x_1, c_1, s_1] = \text{gsvd}(a_{11}, b)$$

$$u_1 = \begin{bmatrix} 0.9996 & 0.0267 \\ -0.0267 & 0.9996 \end{bmatrix}$$

$$v_1 = \begin{bmatrix} -0.5690 & -0.8224 \\ -0.8224 & 0.5690 \end{bmatrix}$$

$$x_1 = (1.0e+5)*\begin{bmatrix} -0.0032 & -1.7470 & -1.9548 \\ -0.0015 & -1.0193 & -2.6659 \\ -0.0004 & -1.6189 & -1.0354 \end{bmatrix}$$

$$c_1 = \begin{bmatrix} 0 & 1.0000 & 0 \\ 0 & 0 & 1.0000 \end{bmatrix}$$

$$s_1 = \begin{bmatrix} 1.0000 & 0 & 0 \\ 0 & 0.0003 & 0 \end{bmatrix}$$

$$[u_2, v_2, x_2, c_2, s_2] = \text{gsvd}(a_{22}, b)$$

$$u_2 = \begin{bmatrix} 0.9999 & 0.0169 \\ -0.0169 & 0.9999 \end{bmatrix}$$

$$v_2 = \begin{bmatrix} -0.6722 & -0.7404 \\ -0.7404 & 0.6722 \end{bmatrix}$$

$$x_2 = (1.0e+4)*\begin{bmatrix} -0.0320 & -0.3023 & -0.8227 \\ -0.0155 & -0.3190 & -1.7714 \\ -0.0041 & -1.4095 & -1.1030 \end{bmatrix}$$

$$c_2 = \begin{bmatrix} 0 & 1.0000 & 0 \\ 0 & 0 & 1.0000 \end{bmatrix}$$

$$s_2 = \begin{bmatrix} 1.0000 & 0 & 0 \\ 0 & 0.0030 & 0 \end{bmatrix}$$

$$aa_1 = u_1*c_2*x_1^T$$

$$= \begin{bmatrix} 0.9996 & 0.0267 \\ -0.0267 & 0.9996 \end{bmatrix}*\begin{bmatrix} 0 & 1.0000 & 0 \\ 0 & 0 & 1.0000 \end{bmatrix}$$

$$*\left((1.0e+5)*\begin{bmatrix} -0.0032 & -1.7470 & -1.9548 \\ -0.0015 & -1.0193 & -2.6659 \\ -0.0004 & -1.6189 & -1.0354 \end{bmatrix}\right)^T$$

$$= (1.0e+5)*\begin{bmatrix} -1.7958 & -1.0900 & -1.6459 \\ -1.9075 & -2.6378 & -0.9919 \end{bmatrix}$$

$$aa_2 = u_2*c_1*x_2^T$$

$$= \begin{bmatrix} 0.9999 & 0.0169 \\ -0.0169 & 0.9999 \end{bmatrix}*\begin{bmatrix} 0 & 1.0000 & 0 \\ 0 & 0 & 1.0000 \end{bmatrix}$$

$$*\left((1.0e+4)*\begin{bmatrix} -0.0320 & -0.3023 & -0.8227 \\ -0.0155 & -0.3190 & -1.7714 \\ -0.0041 & -1.4095 & -1.1030 \end{bmatrix}\right)^T$$

$$= (1.0e+4)*\begin{bmatrix} -0.3161 & -0.3488 & -1.4279 \\ -0.8175 & -1.7658 & -1.0791 \end{bmatrix}$$

$$f = \begin{bmatrix} aa_1 \\ aa_2 \end{bmatrix} = \begin{bmatrix} (1.0e+5)*\begin{bmatrix} -1.7958 & -1.0900 & -1.6459 \\ -1.9075 & -2.6378 & -0.9919 \end{bmatrix} \\ (1.0e+4)*\begin{bmatrix} -0.3161 & -0.3488 & -1.4279 \\ -0.8175 & -1.7658 & -1.0791 \end{bmatrix} \end{bmatrix}$$

Decryption:

$$f = \begin{bmatrix} (1.0e+5)*\begin{bmatrix} -1.7958 & -1.0900 & -1.6459 \\ -1.9075 & -2.6378 & -0.9919 \end{bmatrix} \\ (1.0e+4)*\begin{bmatrix} -0.3161 & -0.3488 & -1.4279 \\ -0.8175 & -1.7658 & -1.0791 \end{bmatrix} \end{bmatrix}$$

$$aa_1 = (1.0e+5)*\begin{bmatrix} -1.7958 & -1.0900 & -1.6459 \\ -1.9075 & -2.6378 & -0.9919 \end{bmatrix}$$

$$aa_2 = (1.0e+4)*\begin{bmatrix} -0.3161 & -0.3488 & -1.4279 \\ -0.8175 & -1.7658 & -1.0791 \end{bmatrix}$$

$$[uu_1, vv_1, xx_1, cc_1, ss_1] = \text{gsvd}(aa_1, b)$$

$$uu_1 = \begin{bmatrix} 0.9996 & 0.0267 \\ -0.0267 & 0.9996 \end{bmatrix}$$

$$vv_1 = \begin{bmatrix} -0.5690 & -0.8224 \\ -0.8224 & 0.5690 \end{bmatrix}$$

$$xx_1 = (1.0e+5)*\begin{bmatrix} -0.0032 & -1.7470 & -1.9548 \\ -0.0015 & -1.0193 & -2.6659 \\ -0.0004 & -1.6189 & -1.0354 \end{bmatrix}$$

$$cc_1 = \begin{bmatrix} 0 & 1.0000 & 0 \\ 0 & 0 & 1.0000 \end{bmatrix}$$

$$ss_1 = \begin{bmatrix} 1.0000 & 0 & 0 \\ 0 & 0.0003 & 0 \end{bmatrix}$$

$$[uu_2, vv_2, xx_2, cc_2, ss_2] = \text{gsvd}(aa_2, b)$$

$$uu_2 = \begin{bmatrix} 0.9999 & 0.0169 \\ -0.0169 & 0.9999 \end{bmatrix}$$

$$vv_2 = \begin{bmatrix} -0.6722 & -0.7404 \\ -0.7404 & 0.6722 \end{bmatrix}$$

$$xx_2 = (1.0e+4)*\begin{bmatrix} -0.0320 & -0.3023 & -0.8227 \\ -0.0155 & -0.3190 & -1.7714 \\ -0.0041 & -1.4095 & -1.1030 \end{bmatrix}$$

$$cc_2 = \begin{bmatrix} 0 & 1.0000 & 0 \\ 0 & 0 & 1.0000 \end{bmatrix}$$

$$ss_2 = \begin{bmatrix} 1.0000 & 0 & 0 \\ 0 & 0.0030 & 0 \end{bmatrix}$$

$$a_{1new} = uu_1 * cc_2 * xx_1^T$$

$$= \begin{bmatrix} 0.9996 & 0.0267 \\ -0.0267 & 0.9996 \end{bmatrix} * \begin{bmatrix} 0 & 1.0000 & 0 \\ 0 & 0 & 1.0000 \end{bmatrix}$$

$$* \left((1.0e+5) * \begin{bmatrix} -0.0032 & -1.7470 & -1.9548 \\ -0.0015 & -1.0193 & -2.6659 \\ -0.0004 & -1.6189 & -1.0354 \end{bmatrix} \right)^T$$

$$= (1.0e+5) * \begin{bmatrix} -1.7985 & -1.0900 & -1.6459 \\ -1.9075 & -2.6378 & -0.9919 \end{bmatrix}$$

$$a_{2new} = uu_2 * cc_1 * xx_2^T$$

$$= \begin{bmatrix} 0.9999 & 0.0169 \\ -0.0169 & 0.9999 \end{bmatrix} * \begin{bmatrix} 0 & 1.0000 & 0 \\ 0 & 0 & 1.0000 \end{bmatrix}$$

$$* \left((1.0e+4) * \begin{bmatrix} -0.0320 & -0.3023 & -0.8227 \\ -0.0155 & -0.3190 & -1.7714 \\ -0.0041 & -1.4095 & -1.1030 \end{bmatrix} \right)^T$$

$$= (1.0e+4) * \begin{bmatrix} -0.3161 & -0.3488 & -1.4279 \\ -0.8175 & -1.7658 & -1.0791 \end{bmatrix}$$

$$a_{12new} = \frac{a_{1new}}{-10*c}$$

$$= \frac{(1.0e+5) * \begin{bmatrix} -1.7985 & -1.0900 & -1.6459 \\ -1.9075 & -2.6378 & -0.9919 \end{bmatrix}}{-10*109}$$

$$= \begin{bmatrix} 164.9986 & 99.9992 & 150.9987 \\ 175.000 & 242.0000 & 91.0000 \end{bmatrix}$$

$$a_{new} = (a_{12new} - b) \bmod (256)$$

$$= \left(\begin{bmatrix} 164.9986 & 99.9992 & 150.9987 \\ 175.000 & 242.0000 & 91.0000 \end{bmatrix} - \begin{bmatrix} 222 & 111 & 59 \\ 231 & 108 & 2 \end{bmatrix} \right)_{mod(256)}$$

$$= \begin{bmatrix} 198.9986 \approx 199 & 244.9992 \approx 245 & 91.9987 \approx 92 \\ 200.0000 \approx 200 & 134.0000 \approx 134 & 89.0000 \approx 89 \end{bmatrix}$$

$$= \begin{bmatrix} 199 & 245 & 92 \\ 200 & 134 & 89 \end{bmatrix} = a$$

Application for the proposed algorithm: Our algorithm is generally applicable to all kinds of color, gray, military, medical and so on. And more than that can be developed to apply to the coding of texts in any language and it's the feature for all the researches we have written previously.

Below are the results, we have obtained and we will only have two images, Lena and Baboon. Where the results were obtained using MATLAB (Knuth, 1997). It's noticed from the above results that this algorithm does not lose any data. This is illustrated by the match of the original images with the image after decoding. In addition, encryption and decryption time is in line with modern business in the field of encryption and information security. The strength of the algorithm in the time required to decode the code is a time that can never be estimated because it is too large not least than the estimated time in our previous search. That it can be said that the algorithm is very powerful and immune against hackers.

RESULTS AND DISCUSSION

Figure 1 and 2 show the exact match between the original image and the image after decoding. This is very clear that the graph equation (Gonzalez and Woods, 2008; Jain, 1989) matches both images. This assures the quality

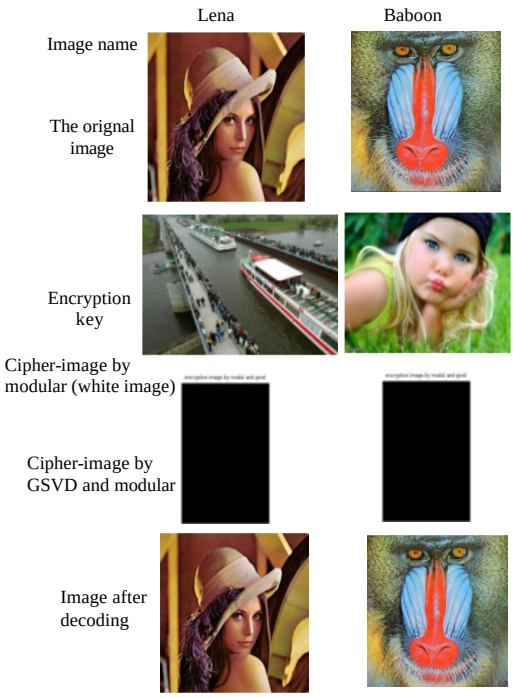

Fig. 1: Sample data base for four images, keys, cipher-images and images after decoding

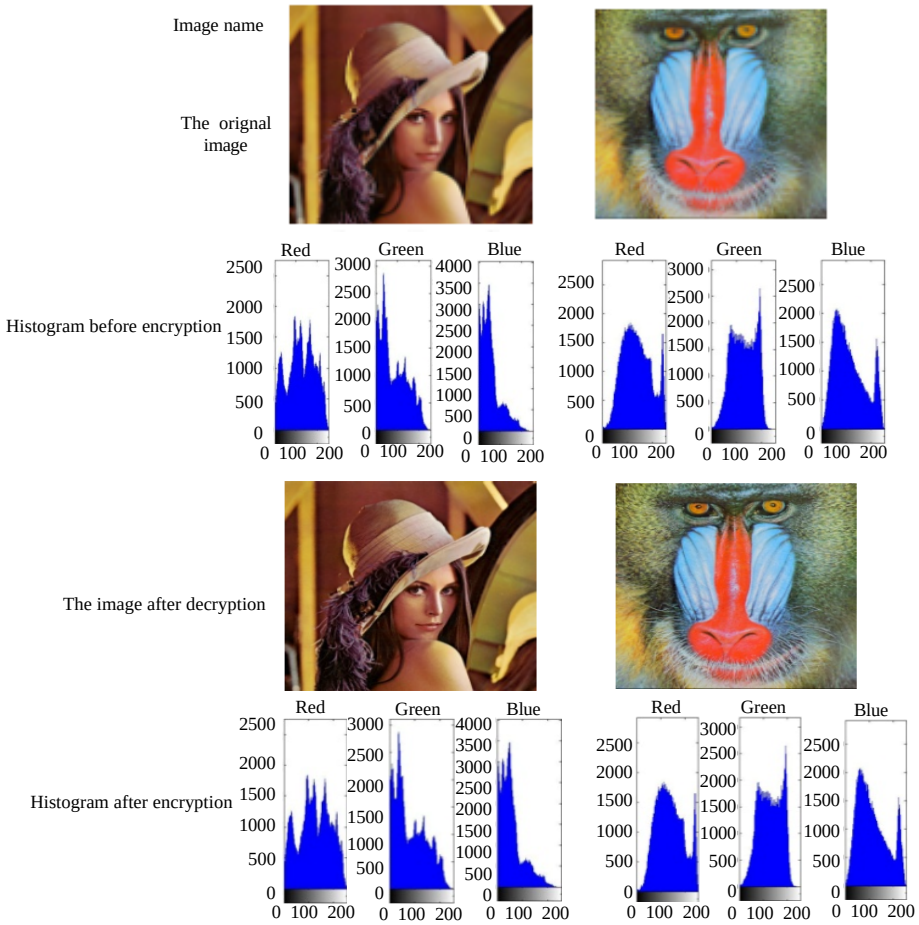

Fig. 2: Images of both Lena and Baboon with histograms of each other before encoding and after decoding

Table 1: Encryption and decryption time, Mean Error, (MSE)* for (Baboon, and Lena) images

Name of image	Baboon	Lena
Encryption (time/sec)	3.46	3.35
Decryption (time/sec)	3.43	3.57
Mean error	0.0013	8.1298e-004
MSE	3.8846e-006	1.9491e-006
PSNR	27.1084	28.5598

of the algorithm and its advantage over a lot of algorithms To identify these statistical concepts can be back to search (Kolman, 1984).

Table 1 shows that, the readings of the coding, decoding time and other accuracy criteria fall within the accepted field in the similar algorithms. Although, it is not ideal, we accept it if we take into account the accuracy and complexity of the algorithm which exceeds all perceptions.

Which assures that this algorithm is very acceptable. We have taken into consideration readings of other universal standards of accuracy used by us in previous algorithms (Jain, 1989; Chaudhari, 2010) we will summarize them in Table 2.

Table 2: Readings for the global standards accuracy before encryption and after decryption for (Baboon and Lena) images

Name of image	Baboon	Lena
EBE	7.7624	7.7275
EAD	7.7632	7.7275
EEI	0	0
SDBE	0.2204	0.2503
SDAD	0.2204	0.2503
SDEI	468.5578	4362442
CCOAD	1	1
CCOE	-0.9851	-0.9884
NPCR	100%	100%
UACI	489.8741	4594327

1[EBE]: Entropy Before Encryption; 2[EAD]: Entropy After Decryption; 3[EEI]: Entropy for Encryption Image; 4[SDBE]: Standard Deviation Before Encryption; 5[SDAD]: Standard Deviation After Decryption; 6[SDEI]: Standard Deviation for Encryption Image; 7[CCOAD]: Correlation Coefficient between Original image and image After Decryption; 8[CCOE]: Correlation Coefficient between Original image and Encrypted image

The entropy of the image, the standard deviation for a separate divider (the probable density function), the correlation coefficient, the tone of the change of the pixel rate and the WSI is the average of variable intensity. Table 2 shows the comparison between the readings of statistical standards before encryption and after

Table 3: Comparing the proposed algorithm with other algorithms

	Encryption (time/sec) (image)		Decryption (time/sec) (image)	
Algorithm	Lena	Baboon	Lena	Baboon
MIE	5.000	9.230	5.16	9.23
VC	4.560	8.350	****	****
Mk-1	2.224	2.287	3.11	3.166
Mk-2	5.368	5.508	6.013	6.104
Mk-3	1.456	1.459	2.138	2.159
Mk-4	5.540	5.567	6.265	6.382
Mk-5	2.522	2.338	2.924	3.104
Mk A-6	7.950	8.240	2.13	2.07
MKHAH-7	3.530	3.450	3.57	3.3
Mk-8	1.799	1.898	0.73	0.74
Our algorithm Mk-9	3.350	3.460	3.57	3.43

(***)means that the address is not calculated

Table 4: Comparing the results of the global standards of accuracy (MSE) and (PSNR) with other works of image processing in general

	MSE (image)		PSNR (image)	
Algorithm	Lena	Baboon	Lena	Baboon
SKM (Abdulla, 2010) JPEG	****	****	21.2	****
SKM (Abdulla, 2010) BPP	****	****	22.7	****
NKP (El-said *et al*., 2010)	****	****	8.67	9.076
DSA (Pareek *et al*., 2006)]	3.81e-6	****	54.18	****
TTS (Aggarwal *et al*., 2010)	0.078	****	29.6041	****
AZA (Trinadh and Narayana, 2012)	****	0.132977	****	56.893054
NDD (Nori *et al*., 2010)	0.0012	****	77.4586	****
Mk-1 (Abdul-Hameed and Al-Kufi, 2014)	9.9137e-26	7.9003e-26	125.0188	125.5118
Mk-2 (Abdul-Hameed and Al-Kufi, 2014)	9.9137e-26	7.9003e-26	125.0188	125.5118
Mk-3 (Abdul-Hameed and Al-Kufi, 2014)	7.3078e-27	8.9787e-27	130.6811	130.2339
Mk-4 (Abdul-Hameed and Al-Kufi, 2014)	9.9137e-26	7.9003e-26	125.0188	125.5118
Mk-5 (Abdul-Hameed and Al-Kufi, 2014)	****	****	140.4432	143.2098
MkA-6 (AL-Rammahi and Al-Kufi, 2016)	0	0	Inf	Inf
MkHAH-7 (Al-Kufi *et al*., 2017)	7.8839e-30	****	145.5163	****
Mk-8 (Al-Kufi *et al*., 2017)	5.0193e-20	4.4692e-21	144.6276	149.8797
Our algorithm Mk-9	1.9491e-006	3.8846e-006	28.5598	27.1084

decryption it indicates that there is no loss of information by virtue of equal readings before and after decryption. As shown in Table 2 readings of other statistical standards.

Furthermore, Table 3 shows a comparison between the readings of encryption and decryption time of our proposed algorithm with other algorithms such as MIE (Mirror-like Image Encryption) (Leung *et al*., 2001); VC (Visual Cryptography) (Yen and Guo, 2000); Mk 1-4 (Al-Kufi *et al*., 2017 level 1-4) (Abdul-Hameed and Al-Kufi, 2014); Mk 5 (Abbadi *et al*., 2014); MkA 6 (AL-Rammahi and Al-Kufi, 2016); MkHAH 7 (Al-Kufi *et al*., 2017), Mk 8 (Kolman, 1984). To compare the readings of the international accuracy standards of our algorithm and some other algorithms (Table 4).

This algorithm is the development of a previous algorithm, we have completed and published with the title (image cryptography via. SVD modular numbers). Where we used in the first algorithm SVD decomposition but in this algorithm (image encryption and decription via. (gsvd-modular numbers), we used gsvd decomposition. we can combined between The first algorithm (Image cryptography via. SVD modular numbers) and this algorithm (image encryption and decription via. (GSVD-modular numbers)) to produce a more complex algorithm combining the decompositions SVD, GSVD and modular numbers.

The encryption and decryption time has been determined based on the MATLAB Version used and the speed of the user's computer. These are scalable readings if a newer MATLAB Version and a faster computer are used.

As we have done in our previous researches which adopted a generation or a new approach to encryption where the text or image is converted into a digital matrix that is difficult to recover except from the authoritative, not as in the old style encryption which ensures that the encryption image is rubbish image. Our algorithm was within the same approach where the color values of the original image have been dispersed to a digital matrix, the intruders can not find any connection between them and the original.

We are adding another mathematical tool on these Algebraic tools (SVD, GSVD, modular numbers). It is a logistic function. Another complication will be added to the algorithm because of the property of this function.

We have made use of multiple statistical standards to check the quality of the algorithm for the purpose of allowing researchers to compare with the results of this algorithm later.

The computer program has been tested for this algorithm by MATLAB for many images and of all kinds (colored, gray, military and medical). The result was excellent, allowing the possibility of use of the beneficiaries immediately after the agreement with us.

The algorithm in MATLAB program: For the purpose of demonstrating the rigor of our work and our algorithm, below is a program for the algorithm by MATLAB in two parts (encryption and decryption) which can be converted by specialists to an operational program that it can be used to apply this algorithm on the ground and in the field of practical application.

Algorithm 1; MATLAB program:

```
clc
clear all
keyc = input('keyc=')
a = imread('C:\Users\baqir\Desktop\5555.png')
[tt1,tt2,tt3] = size(a)
if tt1>tt2
 ttt1 = [tt1 600]
a = imresize(a, [min(ttt1) (tt2*min(ttt1))/tt1])
else
   ttt2 = [tt2 600]
   a = imresize(a [(tt1*min(ttt2))/tt2 min(ttt2)])
end
subplot(1, 1, 1); imshow(a); title ('the origin image')
flag = input('flag=')
a123 = a
subplot(2, 2, 1); imshow(a); title ('origin image')
subplot(2, 2, 2); imhist(a(:,:,1)); title ('red')
subplot(2, 2, 3); imhist(a(:,:,2)); title ('green')
subplot(2, 2, 4); imhist(a(:,:,3)); title ('blue')
%%%%%%%%%%%%%%%%
flag = input('flag=')
key = imread('C:\Users\baqir\Desktop\3.jpg')
subplot(1, 1, 1); imshow(key); title ('the key image')
[tt1, tt2, tt3] = size(a)
key1 = imresize(key[tt1 tt2])
flag = input('flag=')
a1 = im2double(a)
aaaaa = a1
keyy = im2double(key1)
a1 = a1+1; key1 = floor(keyy+1)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
time1 = clock
am1 = a1+key1
am2 = a1+5*key1
amodul1 = mod(am1, 256)
amodul2 = mod(am2, 256)
amodul = [amodul1; amodul2]
aaa1 = amodul1
aaa2 = amodul2
aaa1 = -10*keyc*aaa1
aaa2 = -keyc*aaa2
for p = 1:tt3
   kkey1 = key1(:,:,p)
   aaa11 = aaa1(:,:,p);aaa21 = aaa2(:,:,p)

   [u1, v1, x1, c1, s1] = gsvd(aaa11, kkey1)
   [u2, v2, x2, c2, s2] = gsvd(aaa21, kkey1)
aa1 = u1*c2*x1'; aa2 = u2*c1*x2'
f(:,:,p) = [aa1; aa2]
end
time2 = clock
subplot(1, 1, 1); imshow(amodul); title ('encryption image by modul')
flag = input('flag=')
subplot(1, 1, 1); imshow(f); title ('encryption image by modul and gsvd')
flag = input('flag =')
subplot(2, 2, 1); imshow(f); title ('Encrypted image by modul and gsvd')
subplot(2, 2, 2); imhist(f(:,:,1)); title ('red')
subplot(2, 2, 3); imhist(f(:,:,2)); title ('green')
subplot(2, 2, 4); imhist(f(:,:,3)); title ('blue')
%%%%%%%%%%%%%%%%%%%%%%%%%%
imwrite(f, 'C:\Users\baqir\Desktop\gsvd and modul incimage.jpg')
flag = input('flag=')
%%%%%%%%%%%%%%%%%%%%%%
time3 = clock
[t1, t2, t3] = size(f)
key2 = key1
[sz1, sz2, sz3] = size(key2)
for k = 1:t3
aa1 = f([1:end/2],:,k)
aa2 = f([end/2+1:end],:,k)
key3 = key2(:,:,k)
[u1, v1, x1, c1, s1] = gsvd(aa1 ,key3)
[u2, v2, x2, c2, s2] = gsvd(aa2, key3)
a111 = u1*c2*x1'
aaaa(:,:,k) = a111
end
aaaa = aaaa/(-10*keyc)
am2 = aaaa-key1
a2 = mod(am2, 256)
a2 = a2-1
subplot(1, 1, 1); imshow(a2); title ('image after decryption')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
time4 = clock
flag = input('flag=')
subplot(2, 2, 1); imshow(a2); title ('image after decryption')
subplot(2, 2, 2); imhist(a2(:,:,1)); title ('red')
subplot(2, 2, 3); imhist(a2(:,:,2)); title ('green')
subplot(2, 2 ,4); imhist(a2(:,:,3)); title ('blue')
flag = input('flag=')
incriptiontime = time2-time1
decryptiontime = time4-time3
aaaa = a2
error = abs(aaaa-aaaaa)
meaneror = mean(error(:))
[t1, t2, t3] = size(aaaa)
sumo = 0
      for i = 1:t3
       summ(i) = 0
         for j = 1:t1
            for p = 1:t2
               summ(i) = summ(i)+(aaaa(j, p, I)-aaaaa(j, p, i))^2
            end
         end
         avs(i) = summ(i)/(t1*t2)
         sumo = sumo+avs(i)
      end
      mse = sumo/t3
```

```
psnr1 = 10*log10(max(aaaa(:))^2/sqrt(mse))
mse1 = (aaaa-aaaaa).^2
mse2 = sum(mse1(:))/(t1*t2*t3)
psnr2 = 10*log10(max(aaaa(:))^2/sqrt(mse2))
entropy_before = entropy(aaaaa)
entropy_after = entropy(aaaa)
entropy_encryption_image = entropy(f)
[t1, t2, t3] = size(aaaa)
aa123 = (aaaa-mean(aaaa(:))).^2
standarddeviation_before = sqrt((1/((t1*t2*t3)-1))*sum(aa123(:)))
aa1234 = (aaaaa-mean(aaaaa(:))).^2
standarddeviation_after = sqrt((1/((t1*t2*t3)-1))*sum(aa1234(:)))
[tt1, tt2, tt3] = size(f)
aa12345 = (f-mean(f(:))).^2
standarddeviation_encryption_image = sqrt((1/(((tt1*tt2*tt3)-
1))*sum(aa12345(:)))
nnn = t1*t2*t3
r1 = aaaaa.*aaaa
r2 = aaaaa.^2
r3 = aaaa.^2
r = (nnn*sum(r1(:))-sum(aaaaa(:))*sum(aaaa(:)))/(sqrt(nnn*sum(r2(:))-
(sum(aaaaa(:)))^2)*sqrt(nnn*sum(r3(:))-(sum(aaaa(:)))^2))
Correlation_coefficient = r
nn1 = f([1:end/2],:,:)
nn2 = f([end/2+1:end],:,:)
rr11 = aaaaa.*nn1
rr12 = aaaaa.*nn2
rr2 = aaaaa.^2
rr31 = nn1.^2
rr32 = nn2.^2
rrr1 = (nnn*sum(rr11(:))-sum(aaaaa(:))*sum(nn1(:)))/(sqrt(nnn*sum(rr2(:))-
(sum(aaaaa(:)))^2)*sqrt(nnn*sum(rr31(:))-(sum(nn1(:)))^2))
rrr2 = (nnn*sum(rr12(:))-sum(aaaaa(:))*sum(nn2(:)))/(sqrt(nnn*sum(rr2(:))-
(sum(aaaaa(:)))^2)*sqrt(nnn*sum(rr32(:))-(sum(nn2(:)))^2))
Correlation_coefficient_origon_encryption1 = rrr1
Correlation_coefficient_origon_encryption2 = rrr2
nn1 = f([1:end/2],:,:)
nn2 = f([end/2+1:end],:,:)
d1 = aaaaa~=nn1
d2 = aaaaa~ = nn2
d1 = double(d1)
d2 = double(d2)
NPCR1 = (sum(d1(:))/(t1*t2*t3))*100
NPCR2 = (sum(d2(:))/(t1*t2*t3))*100
dd1 = abs(aaaaa-nn1)./(t1*t2*t3*255)
dd2 = abs(aaaaa-nn2)./(t1*t2*t3*255)
UACI1 = sum(dd1(:))*100
UACI2 = sum(dd2(:))*100
Dimension_image=size(aaaa)
```

CONCLUSION

In this algorithm (image encryption and description via. GSVD modular numbers), we have focused on the complexity of dispersing the chromatic values of the image and passing through the two encoding stages as it is clear, making the possibility of breaking the code impossible and never considered. Therefore, our algorithm has become a qualitative leap in the world of information security and is ready for entry into the application as well as it can be added simple touches and develop it to be ready for use in text encryption. The properties of this algorithm can be summarized in the following points: the algorithm is a complex coding process where the second process encrypts the encrypted image and increases the complexity. Adopt an image as a cryptographic key in the first and second stage of encryption and this makes it very difficult to break the code.

We can use this algorithm to encrypt any type of colored images and gray/military, medical and other. The possibility of developing this algorithm to apply to text encryption and in any language. The encoding time and the decoding time are within the acceptable rate which is in line with similar algorithms which is clear from Table 1. The relationship between the original image and the decoded image is equal to 1 and it's the highest possible value indicating the strength of the algorithm.

ACKNOWLEDGEMENTS

We extend our thanks to Mss. 'Bushra Lateef Saddam'/Director of Elementary School (Almoumenat) for girls, affiliated to the Directorate of Education Kufa in Najaf for the effort made as in previous research in our support and help us by its means available where it had a major role in the extraction of this research in its current form.

And we also extend our sincere thanks to the distinguished educational teacher (Nagham Salem Hussein Al-Mawashi) which it has had a significant role in our support for the completion of this research. Mohammed Abdul Hameed Jassim developed the algorithm idea and programming them on MATLAB and extracting the results, making comparisons, readings, tables and writing research. Mohammed Abdul Hameed Jassim and Dheiaa Shakir Redhaa are translated the research and reviewed it according to grammar. Mohammed Abdul Hameed Jassim, Mohammed Mundher Neamah and Ali Mohammed Taher are organized a search form as required by the journal form. Mohammed Abdul Hameed Jassim reviewed the final design and modified it to make the search the best formula.

REFERENCES

Abbadi, N.K.E., A. Mohamad and M. Abdul-Hameed, 2014. Image encryption based on singular value decomposition. J. Comput. Sci., 10: 1222-1230.

Abdul-Hameed, M. and J. Al-Kufi, 2014. Image encryption with singular values decomposition aided. MSc Thesis, University of Kufa, Kufa, Iraq.

Abdulla, S., 2010. New visual cryptography algorithm for colored image. J. Comput., 2: 21-25.

Aggarwal, E., E. Kaur and E. Anantdeep, 2010. An efficient watermarking algorithm to improve payload and robustness without affecting image perceptual quality. J. Comput., 2: 105-109.

Al-Kufi, M.A.H.J., H.R. Hashim, A.M. Hussein and H.R. Mohammed, 2017. An algorithm based on GSVD for image encryption. Math. Comput. Appl., 22: 1-8.

Al-Rammahi, A. and M. Al-Kufi, 2016. Image cryptography via SVD modular numbers. Eur. J. Sci. Res., 138: 1-2.

Chaudhari, J.C., 2010. Design of artificial back propagation neural network for drug pattern recognition. Intl. J. Comput. Sci. Eng., 2010: 1-6.

Divya, V.V., S.K. Sudha and V.R. Resmy, 2012. Simple and secure image encryption. Intl. J. Comput. Sci. Issues, 9: 286-289.

El-said, S.A., K.F. Hussein and M.M. Fouad, 2010. Securing image transmission using in-compression encryption technique. Intl. J. Comput. Sci. Secur., 4: 466-481.

Gonzalez, R.C. and R.E. Woods, 2008. Digital Image Processing. 3rd Edn., Prentice hall, Upper Saddle River, New Jersey, USA., ISBN:9780135052679, Pages: 954.

Hameed, M.A. and J. Al-Kufi, 2018. An a new algorithm based on (General Singular values Decomposition) for image cryptography. Elixir Digital Process., 114: 49604-49609.

Hansen, P.C., 1989. Regularization, GSVD and truncated GSVD. BIT. Numer. Math., 29: 491-504.

Jain, A.K., 1989. Fundamentals of Digital Image Processing. 4th Edn., Prentice Hall, Upper Saddle River, New Jersey, USA., ISBN:9780133361650, Pages: 569.

Jassim, M.A.H., 2016. Text and image encryption via text and image keys using singular value decomposition. Intl. J. Eng. Future Technol., 1: 1-16.

Knuth, D.E., 1997. The Art of Computer Programming: Sorting and Searching. Addison-Wesley, Boston, Massachusetts, USA., ISBN:9780201896855, Pages: 780.

Kolman, B., 1984. Introductory Linear Algebra with Applications. 3rd Edn., Macmillan Publishers, Basingstoke, USA., ISBN:9780023660108, Pages: 455.

Leung, L.W., B. King and V. Vohora, 2001. Comparison of image data fusion techniques using entropy and INI. Proceedings of the 22ndAsian Conference on Remote Sensing Vol. 5, November 5-9, 2001, National University of Singapore, Singapore, pp:1-6.

Macq, B.M. and J.J. Quisquater, 2005. Cryptology for digital TV broadcasting. Proc. IEEE, 83: 944-957.

Nori, A.S., Z.M. Taha and A.B. Sallow, 2010. An investigation for steganography using different color system. AL. Rafidain J. Comput. Sci. Math., 7: 91-108.

Pareek, N.K., V. Patidar and K.K. Sud, 2006. Image encryption using chaotic logistic map. Image Vision Comput., 24: 926-934.

Parker, T.S. and L.O. Chua, 1987. Chaos: A tutorial for engineers. Proc. IEEE, 75: 982-1008.

Sathishkumar, G.A., K.B. Bagan and N. Sriraam, 2011. Image encryption based on diffusion and multiple chaotic maps. Int. J. Network Secur. Applic., 3: 181-194.

Shah, J. and V. Saxena, 2011. Performance study on image encryption schemes. Intl. J. Comput. Sci. Issues, 8: 349-355.

Tadala, T. and S.E.V. Narayana, 2012. A novel PSNR-B approach for evaluating the quality of De-blocked images. IOSR. J. Comput. Eng., 4: 40-49.

Wei, Y., P. Xie and L. Zhang, 2016. Tikhonov regularization and randomized GSVD. SIAM. J. Matrix Anal. Appl., 37: 649-675.

Wu, C.W. and N.F. Rul'kov, 1993. Studying chaos via. 1-D maps-A tutorial. IEEE. Trans. Circuits Syst. I. Fundam. Theory Appl., 40: 707-721.

Yang, M., N. Bourbakis and S. Li, 2004. Data-image-video encryption. Potentials, 23: 28-34.

Yen, J.C. and J.I. Guo, 2000. A new chaotic mirror-like image encryption algorithm and its VLSI architecture. Pattern Recognit. Image Anal. Adv. Math. Theory Appl., 10: 236-247.

Yi, X., C.H. Tan, C.K. Slew and M.R. Syed, 2001. Fast encryption for multimedia. IEEE. Trans. Consum. Electron., 47: 101-107.