Image Encryption Algorithm Using Ancient Magic Squares

Parth Parikh[1], Mrs. Geetali Saha[2], Dr. Chintan Modi[2]
[1] IEEE Gujarat Section, R10, Gujarat, India
[2] Dept. of Electronics & Communication, G. H. Patel College of Eng. & Tech, GTU, Gujarat, India
parthpparikh@yahoo.com, gitali.saha@gmail.com, chintankmodi@yahoo.com

Abstract - **Digital data is highly susceptible to malicious attacks when it is transmitted and hence needs to be heavily and reliably encrypted to prevent theft. Also, we often observe that attackers not only attempt to hack the data but also degrade it. In this paper, we use ancient magic squares to develop a location rearrangement based symmetric key image encryption algorithm that sustains different attacks of astronomical degrees to provide a visually satisfactory decrypted image. The results further this claim.**

Keywords - **Image Encryption, Ancient Magic Squares, Rearrangement, Distortive Attacks.**

I. INTRODUCTION

Cryptography has been an essential part of science since early days. The earliest known use of cryptography is some carved cipher text on stone in Egypt (1900 BC). The ancient use of cryptography was limited to sending confidential messages without the concern for additional noise.

However, with advancement in technologies, it becomes imperative to enhance the encryption methodologies. Hence protective digital cryptography comes into play.

Traditional encryption algorithms which have been used for image encryption are Data Encryption Standard (DES) [1], Blowfish Algorithm [2], and Advances Encryption Standard (AES) [3]. Few methods that employ chaos based scrambling for image encryption are [4 and 5].

Encryption algorithms are broadly classified into *symmetric-key encryption* and *public-key encryption*. Symmetric-key encryption refers to encryption methods in which both the sender and receiver share the same key. On the other hand public-key encryption methods employ a different set of key at every node in a network. In this paper, we develop a symmetric key image encryption algorithm based on permutation properties of ancient magic squares.

In this paper, we propose an algorithm that performs heavy data encryption based on multiple scrambling, providing protective redundancy and resilience to burst errors and attacks that intend to deform the encrypted data. We evaluate the performance of the algorithm by encrypting images that are subsequently subjected to resizing, salt and pepper noise, and speckle noise.

In section 2, we elaborate on the problem statement and its relevance. The key techniques used for performing encryption and thereby testing the robustness of the same are discussed in section 3. The proposed algorithm for encryption along with the performance parameters is discussed in section 4. Section 5 contains the observations of applying the various tools and their analysis. The paper is concluded in section 6 followed by references.

II. PROBLEM DEFINITION

To develop an image encryption algorithm based on ancient magic squares and rearrangement methods, which provides redundancy and security against resizing, salt and pepper noise, and speckle noise.

Most encryption algorithms are competent to serve the purpose of encryption. However, the need of the hour is an encryption algorithm that protects the data from theft as well as provides resistance to attacks that intend to alter the data. The images chosen in this paper are randomly clicked images that provide a wide range of grey scale shade variation within the image. The test images viz. *crow*, *fly* and *tub* are shown in Fig. 1.

III. TECHNIQUES USED FOR ENCRYPTION AND TESTING

A. Bit Expansion

One of the earliest methods to provide redundancy to check single bit errors as well as burst errors, induced due to channel noise while transmission, was to replicate the data. This basic principle is exploited in this paper to provide extensive redundancy that can curb up to $n/2-1$ errors if n bits of redundancy are provided.

In this paper we provide redundancy at two extreme stages in the encryption process.

Fig. 1 Specimen images
(a) Crow (236x236) (b) Fly (236x154) (c) Tub (236x121)

The total redundancy provided is *seven times* the data. This makes the algorithm highly secure against bit errors induced by intentional noise or channel.

The bit expansion used in this paper can be described by the illustration shown in Fig. 2.

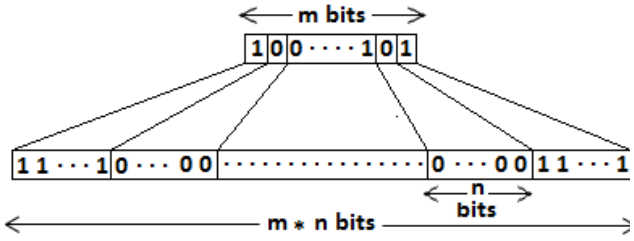

Fig. 2 Introduction of Redundancy by Bit Expansion

where,

- *m* is the length of the original data
- *n* is the expansion coefficient
- *m* x *n* is the length of the expanded data

B. Rearrangement

The most important process that makes the proposed algorithm effective is rearrangement. Rearrangement provides the required scrambling of data to protect it from being decoded upon hacking.

In this paper, rearrangement is performed *four times* using Ancient Magic Squares [6] and regular Magic Squares [6] to make the encryption ultra-robust.

The four magic squares used for scrambling are:

- Albrecht Durer's Magic Square(4x4)
- Jupiter Astronomical Matrix(4x4)
- Mercury Astronomical Matrix(8x8)
- Symmetric Magic Square(8x8)

Albrecht Durer's magic square was invented by the mathematician in the year 1514AD. Jupiter and Mercury Astronomical Matrices were developed around the same time in 1510AD by Heinrich Cornelius Agrippa. The Magic Squares are illustrated in Fig. 3.

C. Encoding

The basic process of any encryption algorithm is the encoding of data. Codes are used for data compression, cryptography, and error-correction. There are essentially two aspects to Coding theory; *Data compression* (or *source coding*) and *Error correction* (or *channel coding*).

In this paper, we employ the simplest form of channel coding wherein we perform e*xclusive-or* operation on the data for a predetermined number of times. There are two encoding procedures employed in the proposed algorithm; using the secret symmetric key and using key secretive to the algorithm. The algorithm works only when both the keys are available.

Bit expansion, Rearrangement, and Encoding are interleaved to enhance the encryption.

16	3	2	13
5	10	11	8
9	6	7	12
4	15	14	1

4	14	15	1
9	7	6	2
5	10	11	8
16	2	3	13

Fig. 3
(a) Albrecht Durer's Magic Square (b) Jupiter Astronomical Matrix

8	58	59	5	4	62	63	1
49	15	14	52	53	11	10	56
41	23	22	44	45	19	18	48
32	34	35	29	28	38	39	25
40	26	27	37	36	30	31	33
17	47	46	20	21	43	42	24
9	55	54	12	13	51	50	16
64	2	3	61	60	6	7	57

Fig. 3 (c) Mercury Astronomical Matrix

1	63	3	61	60	6	58	8
56	10	54	12	13	51	15	49
17	47	19	45	44	22	42	24
40	26	38	28	29	35	31	33
32	34	30	36	37	27	39	25
41	23	43	21	20	46	18	48
16	50	14	53	53	11	55	9
57	7	59	5	4	62	2	64

Fig. 3 (d) Symmetric Magic Square

D. Distortive attacks

When the attempts to hack the transmitted data fail, hackers resort to tampering the data, making it difficult to decrypt a meaningful message from the distorted data that is received. Also many a time, unintentional channel noise adds to the encrypted data in the form of salt and pepper or speckle noise. In this paper, we attempt at simulating extreme conditions to which the encrypted data might be subjected.

In this paper, we analyze the effectiveness of the algorithm against resizing, salt and pepper noise, and speckle noise whose attributes are stated below.

1) Resizing

Resizing is the process of increasing or decreasing the overall size of the encrypted data, which in our case is an array of binary values. The effect of resizing is that when restoring the data to the original size by interpolation, it often happens that the procedure used for interpolation tend to change the face value of the data. The effects are drastic when the resizing is

performed to a very smaller scale. In this paper, we employ resizing to five different values from resizing to an extremely small scale to very large ones to test the effectiveness of the algorithm.

2) Salt and Pepper Noise

Salt and pepper noise is a random noise caused by errors in data transmission that affect individual pixels in an image. The affected pixels are either reduced to the minimum value or enhanced to the maximum value in the image, rendering black and white pixels on the image. This gives the noise it's so called name as the affected image.

In this paper we apply varying intensities of salt and pepper noise to the encrypted data to test the robustness of the encryption algorithm. While most algorithms give in above 10% noise, the proposed algorithm endures noise at extreme levels.

3) Speckle Noise

Speckle noise is commonly found in ultrasound medical images. The existence of speckle is unwarranted since it degrades image quality and it affects the tasks of individual interpretation and diagnosis. This noise is, in fact, caused by errors in data transmission. The corrupted pixels are either set to the maximum value, which is something like a snow in image or have single bits flipped over. Speckle noise has the characteristic of multiplicative noise and follows a gamma distribution.

IV. Performance Parameters and Proposed Algorithm

A. Performance Parameters

In this subsection we discuss the various parameters sued to evaluate the performance of the encryption algorithm. The performance is evaluated on algebraic entities as well as on the holistic distribution of the pixel values of the decrypted image (histogram approach). The algebraic parameters used are *Correlation Coefficient (CC), Unified Average Changing Intensity* [7] *(UACI), Number of Pixels Change Rate* [7] *(NPCR), and Pixel Value Change Ratio (PVCR).* They are defined by the following equations:

$$CC = \frac{\sum_{i,j}(D_{i,j}-\gamma)*(O_{i,j}-\theta)}{\sqrt{\sum_{i,j}(D_{i,j}-\gamma)^2 * \sum_{i,j}(O_{i,j}-\theta)^2}} \quad (1)$$

$$UACI = \frac{1}{i*j}\left[\sum_{i,j}\frac{|D_{i,j}-O_{i,j}|}{255}\right] * 100\% \quad (2)$$

$$NPCR = \frac{(i*j)-(count)}{(i*j)} * 100\% \quad (3)$$

$$PVCR = \frac{counte/(l*w)}{count/(i*j)} * 100\% \quad (4)$$

where,

- O is the original specimen image
- D is the decrypted image.
- i, j denote the length and width of the specimen image
- l, w denote the length and width of the encrypted image
- is mean of all pixel values of decrypted image.
- is mean of all pixel values of original image.
- *count* is the number of bits in error in decrypted image
- *counte* is the number of bits in error in noisy encrypted data.

B. Proposed Algorithm

Table I enlists the various steps of the algorithm which are then explained in brief.

TABLE I: PROPOSED ALGORITHM

Step No.	Task
1	Read the specimen image, *O*.
2	Determine the size of *O.*
3	Enter secret key, *k*.
4	Rearrange *k* using Jupiter Astronomical Matrix to get *K*.
5	Encrypt each pixel of *O* using tasks described in steps 6-13.
6	Convert decimal pixel value, *p* to 8-bit binary, *b*.
7	Perform *bit expansion* on *b* with *n=2* to obtain *c*.
8	Perform encoding on *c* using *K* to get *P4*.
9	Rearrange *P* using Albrecht Durer's Magic Square to get *P1*.
10	Perform *bit expansion* on *P1* with *n=4* to obtain *P2*.
11	Rearrange *P2* using Symmetric Magic Square to get *P3.*
12	Perform encoding on *P3* using code inherent key, *Ki* to get *P4*.
13	Rearrange *P4* using Mercury Astronomical Matrix to get encrypted binary data *e*
14	Repeat steps 6-13 for all pixels in the image and reassemble to get encrypted binary image *E*.
15	Determine the size of *E.*
16	Subject *E* to various attacks to get *Ea*.
17	Extract 64-bit blocks from *Ea.*
18	Repeat steps 6-13 in reverse order on each 64 bit block to obtain decrypted image, D.
19	Calculate *CC, UACI, NPCR* and *PVCR* as described in section *IV-B,* equations 1 through 4.

First the specimen image data is read. The next step is to determine its size. The secret key is then fed to the algorithm by the user. The next step is to obtain rearranged key by using Jupiter Astronomical Matrix. Then we encrypt the image pixel wise. The decimal pixel value is converted to 8-bit binary and expanded to double the original size. This expanded data is then encoded using rearranged key. Further, rearrangement is using Albrecht Durer's Magic Square and subsequent expansion to four times the original data size. This data is then rearranged using the Symmetric Magic Square and encoded using inherent key of the algorithm. A rearrangement step using Mercury Astronomical Matrix delivers the encrypted data. The above steps are repeated for all pixels in the image. The next step is to obtain the size of the encrypted image. The 16th step involves deliberate attacks on the encrypted image. Blocks of 64-bits are extracted from the distorted

encrypted data. To perform decryption steps 6-13 of the algorithm are performed in the reverse order on all 64 bit blocks, substituting averaging operation for bit expansion. The final step involves calculating the performance parameters as explained under.

V. RESULTS AND OBSERVATIONS

This section tabulates the observations and illustrates the result images. Table II, III and IV tabulate the observations of applying Resizing, Salt and Pepper noise, and Speckle noise attacks on encrypted specimen images and subsequent decryption for crow, fly, and tub respectively. The values for the correlation coefficient for all the images are considerably large, providing evidence for the effectiveness of the algorithm. The original image, its histogram and the encrypted image for crow are shown in Fig. 4. The results for different attacks on the encrypted image shown in Fig 4(c) are shown in Fig. 5 through Fig. 8.

VI. CONCLUSIONS

We proposed a simple algorithm to encrypt image data using ancient matrices. The proposed algorithm works well on different types of test images, all of which were subjected to three types of attacks. The results for one of the images are illustrated in the paper. The simplicity of the algorithm is suitable for online encryption. In the future we intend to extend this algorithm to encompass video data encryption with higher noise.

REFERENCES

[1] Ibrahim E. Ziedan, Mohammed M. Fouad, Doaa H. Salem, "Application of Data Encryption Standard to Bitmap and JPEG Images" in Twentieth National Radio Science Conference, 2003.

[2] Nirmala Palaniswamy, Dipesh Dugar M, Dinesh Kumar Jain, Raaja Sarabhoje G, "Enhanced Blowfish algorithm using bitmap image pixel plotting for security improvisation" in 2nd International Conference on Education Technology and Computer, 2010.

[3] S.H. Kamali, R. Shakerian, M. Hedayati, M. Rahmani, "A new modified version of Advanced Encryption Standard based algorithm for image encryption" in International Conference on Electronics and Information Engineering, 2010.

[4] Yong-Hong ZHANG, Bao-Sheng KANG, Xue-Feng ZHANG, "Image Encryption Algorithm Based On Chaotic Sequence" in 16th International Conference on Artificial Reality and Telexistence, 2006

[5] Jiu-Lun FAN, Xue-Feng ZHANG, "Image Encryption Algorithm Based on Chaotic System", in Computer-Aided Industrial Design and Conceptual Design, 2006.

[6] William Symes Andrews, "Magic Squares and Cubes", Dover Publications Inc., 2000.

[7] K. Thangavel, R. Manavalan, I. Laurence Aroquiaraj, "Removal of Speckle Noise from Ultrasound Medical Image based on Special Filters: Comparative Study", in ICGST-GVIP Journal, ISSN 1687-398X, Volume (9), Issue (III), June 2009.

TABLE II: OBSERVATIONS FOR CROW

Attack \ Parameter		CC	UACI (%)	NPCR (%)	PVCR (%)
Resizing factor	0.2	0.9592	3.5346	28.3100	37.809
	0.8	0.9975	0.3557	85.2400	49.631
	3.7	1.0000	0.0000	100.000	Inf
	5.3	1.0000	0.0000	100.000	Inf
Salt and Pepper Noise	5%	0.9999	0.0130	99.7000	973.4375
	10%	0.9983	0.0983	98.6000	453.3604
	25%	0.9842	0.9585	88.3700	128.9979
	65%	0.8481	7.5003	43.8000	46.8097
Speckle Noise	5	0.9957	0.3567	93.5500	163.5733
	10	0.9757	1.7466	72.5100	70.6784
	50	0.9743	2.0664	65.7800	61.0629
	100	0.9584	2.9071	57.4600	54.1926
	500	0.9584	3.0597	54.2900	52.4355

TABLE III: OBSERVATIONS FOR FLY

Attack \ Parameter		CC	UACI	NPCR	PVCR
Resizing factor	0.2	0.8502	10.567	24.1970	35.7517
	0.8	0.9961	0.6640	82.1515	37.9947
	3.7	1.0000	0.0000	100.000	Inf
	5.3	1.0000	0.0000	100.000	Inf
Salt and Pepper Noise	5%	0.9998	0.0173	99.7576	930.584
	10%	0.9991	0.0878	98.6364	387.827
	25%	0.9873	1.0238	89.0455	109.252
	65%	0.8657	8.1299	46.9848	44.5186
Speckle Noise	5	0.9951	0.4812	93.0606	175.2771
	10	0.9774	1.9490	74.5606	70.0944
	50	0.9748	2.3993	68.2420	63.4425
	100	0.9606	3.3969	57.8333	54.8478
	500	0.9537	3.7920	54.7273	54.4955

TABLE IV: OBSERVATIONS FOR TUB

Attack \ Parameter		CC	UACI	NPCR	PVCR
Resizing Factor	0.2	0.9461	6.6473	36.1923	43.0968
	0.8	0.9943	1.0109	83.1923	43.2548
	3.7	1.0000	0.0000	100.000	Inf
	5.3	1.0000	0.0000	100.000	Inf
Salt and Pepper Noise	5%	0.9998	0.0187	99.6154	639.142
	10%	0.9986	0.1426	98.3462	351.604
	25%	0.9893	1.2532	85.2885	82.2271
	65%	0.9050	8.7288	39.0962	41.4028
Speckle Noise	5	0.9943	0.6803	90.8654	116.542
	10	0.9708	3.2561	66.5962	57.0122
	50	0.9581	4.4756	59.0385	51.5899
	100	0.9428	5.9254	49.5769	44.9385
	500	0.9383	6.5268	46.5962	43.6990

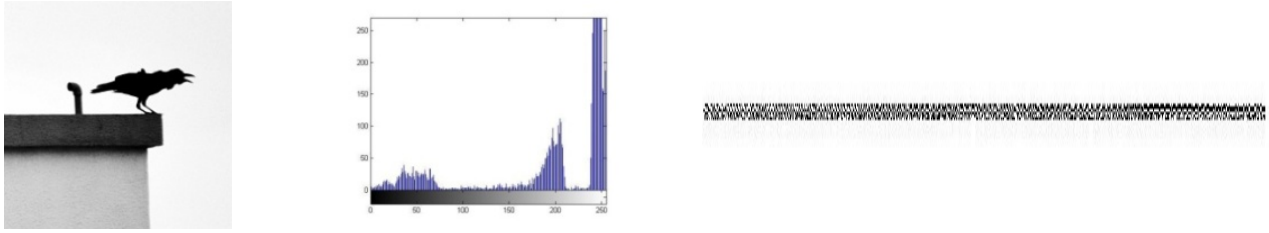

Fig. 4
(a) Specimen image, crow (b) Histogram of crow (c) Encoded binary image for crow.

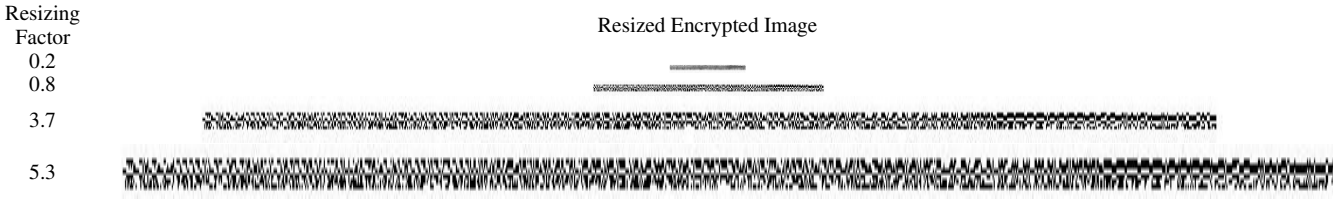

Fig. 5 Resizing factor and scaled resized encrypted image from Fig. 4(c)

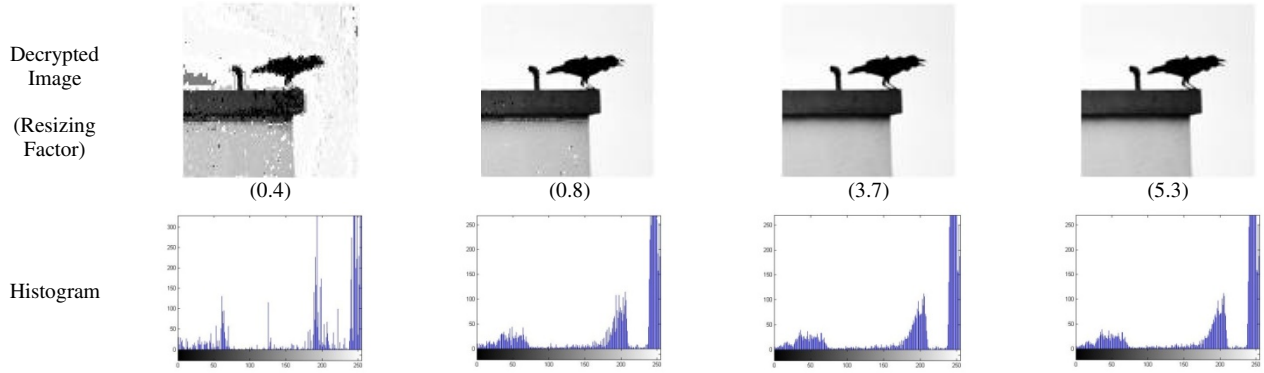

Fig. 6 Decrypted image and corresponding histogram pertaining to resizing factor mentioned.

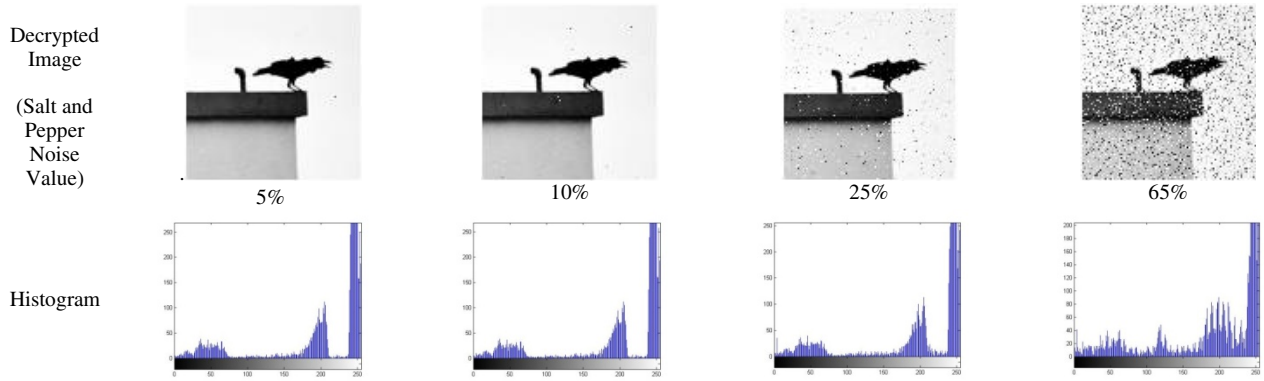

Fig. 7 Decrypted image and corresponding histogram pertaining to Salt and Pepper noise attack mentioned.

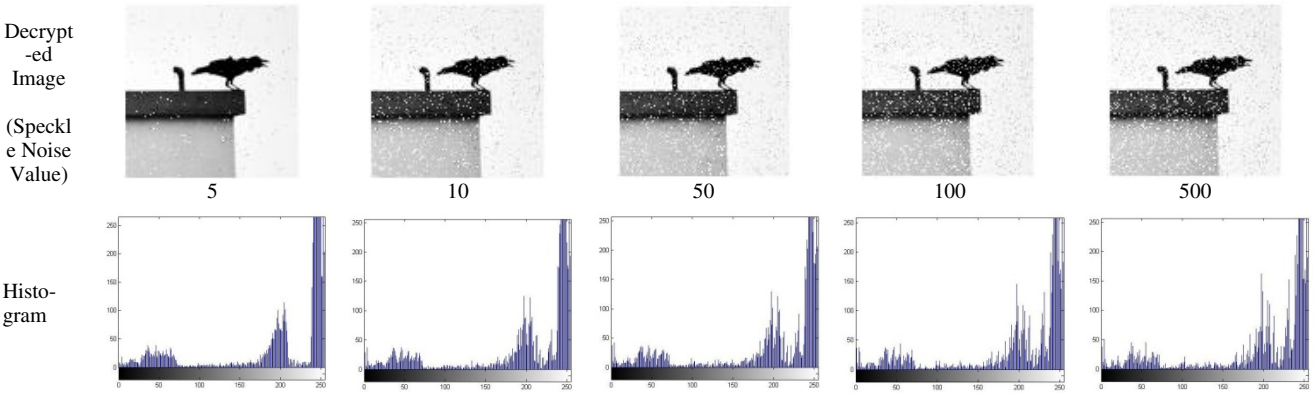

Fig. 8 Decrypted image and corresponding histogram pertaining to Speckle noise attack mentioned.