New modified map for digital image encryption and its performance

Suryadi MT, Maria Yus Trinity Irsan, and Yudi Satria
Department of Mathematics, Faculty of Mathematics and Natural Sciences
Universitas Indonesia, Depok 16424
E-mail: yadi.mt@sci.ui.ac.id, maria.irsan@sci.ui.ac.id, ysatria@sci.ui.ac.id

Abstract. Protection to classified digital data becomes so important in avoiding data manipulation and alteration. The focus of this paper is in data and information protection of digital images form. Protection is provided in the form of encrypted digital image. The encryption process uses a new map, \( x_{n+1} = \frac{r \lambda x_n}{1 + \lambda(1 - x_n)^2} \mod 1 \), which is called MS map. This paper will show: the results of digital image encryption using MS map and how the performance is regarding the average time needed for encryption/decryption process; randomness of key stream sequence with NIST test, histogram analysis and goodness of fit test, quality of the decrypted image by PSNR, initial value sensitivity level, and key space. The results show that the average time of the encryption process is relatively same as the decryption process and it depends to types and sizes of the image. Cipherimage (encrypted image) is uniformly distributed since: it passes the goodness of fit test and also the histogram of the cipherimage is flat; key stream, that are generated by MS map, passes frequency (monobit) test, and runs test, which means the key stream is a random sequence; the decrypted image has same quality as the original image; and initial value sensitivity reaches \( 10^{-17} \), and key space reaches \( 3^{24} \times 10^{634} \). So, that encryption algorithm generated by MS map is more resistant to brute-force attack and known plaintext attack.

1. Introduction

Encryption algorithm of digital image based on chaotic system has been widely introduced, as in [1–3]. The study of chaotic systems and their possible applications to cryptography has received considerable attention during the last years in part of the scientific community due to the properties of chaotic system [4]. The properties of chaotic system was described in [4], that is ergodicity, sensitivity of initial condition/ control parameter, mixing property, deterministic dynamic, and structure complexity. [1] has introduce an encryption algorithm of digital image and its performance by using logistic map as a chaotic map/ key stream generator. In this paper we use MS map [5] as our key stream generator. MS map shown at equation (1)

\[
x_{n+1} = \frac{r \lambda x_n}{1 + \lambda(1 - x_n)^2} \mod 1 \tag{1}
\]

where \( r, \lambda \in \mathbb{R} \) and \( x_n \in (0, 1) \).

The value of all parameters chosen by looking into the bifurcation diagram of MS map. Figure 1 shows the bifurcation diagram for \( r = 3.5 \). [6]

First 100 iteration from MS map shown on Figure 2 with \( r = 3.5; \lambda = 2.1; x_0 = 0.3 \). [6]
2. MAIN RESULT

2.1. Encryption and Decryption Algorithm

The encryption process shown on Figure 3, and the decryption process shown on Figure 4. [5]

Let the original image has pixel size $p \times q$. The key stream ($K_i$) obtained by:

$$K_i = (1000 \times x_i) \mod 256$$

(2)
where $x_i$ generated by MS map and $i = 0, 2, 3pqd - 1$, $d$ is the image dimension. $d = 1$ for greyscale image and $d = 3$ for color image.

2.2. Testing Data and Encryption Result
We use baboon.jpg in two version (greyscale and color) as our testing data with 5 different pixel size for each version [6]. The data testing shown below:

![Figure 5: Display Data](image)

![Table 1: Data Testing](image)

| Image Version | Data | Size (Pixel) |
|---------------|------|--------------|
| Greyscale     | 1    | $77 \times 77$ |
| Greyscale     | 2    | $212 \times 212$ |
| Greyscale     | 3    | $512 \times 512$ |
| Greyscale     | 4    | $1590 \times 1590$ |
| Greyscale     | 5    | $2783 \times 2783$ |
| Color         | 6    | $77 \times 77$ |
| Color         | 7    | $212 \times 212$ |
| Color         | 8    | $512 \times 512$ |
| Color         | 9    | $1590 \times 1590$ |
| Color         | 10   | $2783 \times 2783$ |

Example of encryption process result shown on Figure 6, and the average time of encryption and decryption process of each data shown on Figure 7.

![Figure 6: Encrypted image of data 3 (left side) and data 8 (right side)](image)
2.3. Algorithm’s Performance

2.3.1. Randomness Key Stream Analysis  The procedure of this test was described on [7]. For $r = 3.5; \lambda = 2.1; x_0 = 0.3$, and generated by 100 terms of key stream, we obtained:

(a) Frequency (monobit) test

1. $n = 100 \times 8 = 800$ bit
2. $|S_{800}| = 12$
3. $s_{obs} = \frac{|S_{800}|}{\sqrt{800}} = \frac{12}{\sqrt{800}} = 0.4242640687119285$
4. $P_{value} = \text{erfc}(\frac{s_{obs}}{\sqrt{2}}) = \text{erfc}(\frac{0.49497474683}{\sqrt{2}}) = 0.6713732405408726$

Since $P_{value} \geq 0.01$, then we accept the key stream sequence as random.

(b) The Run Test

1. $n = 100 \times 8 = 800$ bit
2. $\tau = \frac{2}{\sqrt{800}} = 0.07071067811865475$
3. $\pi = 0.4925$
4. $V_{800}(\text{obs}) = 419$
5. $P_{value} = \text{erfc}(\frac{|V_{800}(\text{obs}) - 2n\pi(1-\pi)|}{2\sqrt{2n\pi(1-\pi)}}) = 0.34337010342129703$

Since $P_{value} \geq 0.01$, then we accept the key stream sequence as random.

2.3.2. Histogram Analysis  We will test the distribution of cipherimage pixel’s values using Goodness of fit test [8]. The test result shown on Table 2 for grayscale image and Table 3 for color image.

Table 2: Test Statistic Value of Grayscale Image

| Data | Test Statistic Value |
|------|----------------------|
| 1    | 267.892561983        |
| 2    | 260.573869705        |
| 3    | 270.802734375        |
| 4    | 239.502114632        |
| 5    | 276.859198777        |

With degrees of freedom $256 - 1 = 255$ and 1% significance level, then from Chi-Squared table we obtained the critical value is 310.4573882199. Since all the test statistic values are less than the critical value, then we conclude that the distribution of cipherimage pixel’s values is uniform. Figure 8 shown the histogram of cipherimage of Data 3 and Figure 9 shown the histogram of cipherimage of Data 8.
Table 3: Test Statistic Value of Color Image

| Data | Test Statistic Value |
|------|----------------------|
|      | Red                  | Green                | Blue                  |
| 6    | 232.832349469        | 234.041322314        | 240.604317760         |
| 7    | 238.530437878        | 248.714845141        | 287.606977572         |
| 8    | 262.916015625        | 228.000000000        | 220.248046875         |
| 9    | 222.893353902        | 246.747802698        | 275.719416162         |
| 10   | 249.128751264        | 271.990065318        | 259.869918473         |

Figure 8: Histogram of cipherimage of Data 3

Figure 9: Histogram of cipherimage of Data 8

2.3.3. Quality of Image Tests We will compare the quality of original image and decrypted image, by Peak Signal to Noise Ratio (PSNR) which is described in [9] with formula $PSNR = 10 \log_{10} \frac{L^2}{MSE}$, where $MeanSquaredError(MSE) = \frac{1}{N} \sum_{i=0}^{p-1} \sum_{j=0}^{q-1} (x_{i,j} - y_{i,j})^2$, $N = pq$, $x_{i,j}$ and $y_{i,j}$ are the $(i,j)$ entry of original image matrix and decrypted image matrix, respectively. (assume that the image size is $(p \times q)$ pixel). If $PSNR = \infty$ then we conclude that the original image and the decrypted image has same quality. Using python we obtained :

Table 4: PSNR Result

| Data | MSE | PSNR |
|------|-----|------|
| 1    | 0   | $\infty$ |
| 2    | 0   | $\infty$ |
| 3    | 0   | $\infty$ |
| 4    | 0   | $\infty$ |
| 5    | 0   | $\infty$ |
| 6    | 0   | $\infty$ |
| 7    | 0   | $\infty$ |
| 8    | 0   | $\infty$ |
| 9    | 0   | $\infty$ |
| 10   | 0   | $\infty$ |

From Table 4 we conclude that the quality between the original image and the decrypted image is same.
2.3.4. Key Sensitivity Analysis  We will decrypt Data 8 with different $x_0$. In encryption process we used $x_0 = 0.3$, in this part we will decrypt the cipherimage with three different $x_0$ as shown on Table 5.

Table 5: Key Sensitive Test Result

| $x_0 = 0.3 \times 10^{-5}$ | $x_0 = 0.3 \times 10^{-15}$ | $x_0 = 0.3 \times 10^{-17}$ |
|--------------------------|---------------------------|-----------------------------|
| ![Image](image1.png)     | ![Image](image2.png)      | ![Image](image3.png)        |
| ![Histogram](histogram1.png) | ![Histogram](histogram2.png) | ![Histogram](histogram3.png) |

From Table 5 we seen the original image will be produced when $x_0 = 0.1 + 10^{-17}$, means that the sensitivity of initial value of this algorithm is reaches $10^{-17}$.

2.3.5. Key Space Analysis  In encryption and decryption algorithm using MS map, there are three parameters, $x_0, r, \lambda$, where $x_0 \in (0, 1)$ and $r\lambda \in$. In Python, the maximum value of float is $1.7976931348623157 \times 10^{308}$; And for real number in interval $(0, 1)$, the precision level reaches $10^{-19}$. So the key space of our algorithm is,

$$1.8 \times 10^{308} \times 1.8 \times 10^{308} \times 10^{18} = 3.24 \times 10^{634}$$

Probabilistically, the probability to obtain a right key pair is

$$\frac{1}{1.8 \times 10^{308}} \times \frac{1}{1.8 \times 10^{308}} \times \frac{1}{10^{18}} = 3.24 \times 10^{634} \approx 0.31 \times 10^{-634}$$

3. Conclusion

From the result above, we conclude that:

(i) MS map can be used on encryption digital image

(ii) Performance of algorithm are:

(a) The encryption algorithm is very difficult to be cracked by known plaintext attack, due to the distribution of cipherimage pixel’s values is uniform (by goodness of fit test and also by histogram of cipherimage). And so the key streams were generated, proved random by the frequency (monobit) test with $P_{value} = 0.6713732405408726 \geq 0.01$; and by the run test with $P_{value} = 0.34337010342129703 \geq 0.01$.

(b) Key sensitivity of the algorithm reaches $10^{-17}$ and key space reaches $3.24 \times 10^{634}$. So that, the encryption algorithm which developed by MS map is resistant to brute-force attack.

(c) The original image and the decrypted image have same quality by PSNR.

(d) Encryptions time and decryptions time is relative same
Acknowledgment

The authors would like to thank Kemenristekdikti and DRPM UI for the grant PUPT Universitas Indonesia 2016 who support us in this research.

References

[1] Suryadi M T, Nurpeti E, and Widya 2014 Performance of Chaos-Based Encryption Algorithm for Digital Image TELKOMNIKA Telecommunication, computing, electronics, and control 12(3)
[2] Sukirman E, Suryadi M T, and Mubarak M A 2014 The Implementation of Henon Map Algorithm for Digital Image Encryption TELKOMNIKA Telecommunication, computing, electronics, and control 12(3)
[3] Nurpeti E and Suryadi M T 2013 Chaos-Based Encryption Algorithm for Digital Image Proceedings IndoMS International Conference on Mathematics and Its Application 2013, Indonesia Mathematical Society
[4] Kocarev L and Lian S Chaos-Based Cryptography: theory, algorithms, and applications Springer Science & Business Media 354
[5] Suryadi M T, Maria Y T I, and Yudi S 2015 Encryption algorithm using new modified map for digital image IndoMS International Conference of Mathematics and Application (IICMA).
[6] Maria Y T I 2016 Analisis Kinerja New Modified Map pada Enkripsi Digital Thesis of Mathematics FMIPA Universitas Indonesia
[7] Rukthin A, Soto J, Nechvatal J, Barker E, Leigh S, Levenson M, Banks D, Heckert d A, Dray J, and Vo S 2010 Statistical test suite for random and pseudorandom number generators for cryptographic applications NIST special publication
[8] Walpole R E, Myers R H, Myers S L and Ye K 2012 Probability and Statistics for Engineers & Scientists (9th edition) Boston:Prentice Hall
[9] Verma A and Kaur I 2013 Techniques and Algorithm Design for the Detection of PSNR in Digital Image International Journal of Computer Applications (0975-8887): National Conference on Structuring Innovation Through Quality SITQ-2013