Using a new permutation and diffusion technique for Image encryption

Ahad Zaregholinejad [1], Shahin Pourzare[2], Mojtaba Hoseini[3]

[1.] Malek Ashtar University of Technology, Dep. of ICT, Tehran, Iran
[2.] Karadeniz Technical University, Dep. of Electrical and Electronics Engineering, Trabzon, Turkey
[3.] Computer Engineering Dept., Amirkabir Univ., Tehran, Iran
a.z.gholinejad@gmail.com, shaahinpour@yahoo.com, mojtabahoseini@aut.ac.ir

Abstract: In recent years, there has been a growing interest in the field of data encryption. To carry out image encryption, this paper presents an algorithm which consists of two processes, namely, the permutation process which uses two-dimensional logistic map to confuse the pixels of R, G and B components at the same time, and the diffusion process which shuffle the image pixels by using a hyper-chaotic map. A 256-bit key is used for generating the parameters of the chaotic systems for increasing the security of the proposed algorithm. The prominent features of the proposed algorithm are high security, high sensitivity, and high speed, which is applicable to encryption of color images. Security and performance aspects of the proposed algorithm were both analyzed, and satisfactory results have been achieved. It is observed that the number of pixel change rate (NPCR), the unified average changing intensity (UACI), and entropy, can satisfy security and performance. The experimental results obtained for the CVG-UGR image database reveal the fact that the proposed algorithm is suitable for practical use to protect the security of digital image information over the Internet.
[Zaregholinejad A, Pourzare S, Hoseini M. **Using a new permutation and diffusion technique for Image encryption.** *Academ Arena* 2017;9(10):41-46]. ISSN 1553-992X (print); ISSN 2158-771X (online). http://www.sciencepub.net/academia. 6. doi:10.7537/marsaaj091017.06.

Keywords: Data encryption; Hyper-chaotic map; Permutation; Security;

1. Introduction

Thanks to development of computer networks, access to digital images through multimedia networks has been more convenient. One of the major issues considering the digital transmission in this virtual environment is the security of digital images and video files. Hence, much research has been carried out to prevent access to digital images by illegal users. Encryption is used to protect data in transit, for example data being transferred via networks. Because of intrinsic features of image such as bulk data capacity and high correlation among pixels, encryption of images is different from that of texts. Conventional ciphers such as IDEA, AES, DES and RSA are not suitable for real time encryption especially for image and video because these ciphers require a large computational time and high computing power[1].

Many recent image encryption algorithms are designed considering secure communications. There exist three major types for image encryption, namely position permutation, the value transformation, and their compounding form. Among the algorithms designed for image encryption in spatial domain, tree structure-based method[2], chaos-based methods[3-10], and 2D cellular automata-based methods[11, 12] are the most popular.

So far, various image encryption schemes based on chaos-based image cryptosystems have been proposed. Owing to the exceptionally desirable properties of chaotic maps, such as randomness, sensitivity, simplicity and ergodicity, chaos-based encryption has suggested a new and efficient way to deal with the intractable problem of fast and highly secure image encryption.

In this paper, we introduce a image encryption algorithm which uses logistic and hyper chaotic maps to encrypt components of the color images with high performance. In order to increase the security of the proposed algorithm, a 256-bit long secret key is used to generate the initial conditions and parameters of the chaotic system by making some algebraic transformations to the key. These algebraic transformations are designed so that the algorithm is greatly sensitive to the changes in even a single bit in the 256-bit secret key. Moreover, according to the encryption transformation used in this paper, the pixels of plain-image are encrypted in a coupling fashion in such a way that a swift change in the original image results in a significant change in the ciphered image. As a result, these transformations along with nonlinearity structure of the chaotic system will enhance the security and sensitivity of the cryptosystem.

The remainder of this paper is organized as follows. Section 2 provides the basic theory of the proposed algorithm which consists of logistic and hyper chaotic maps. In Section 3, the proposed key schedule, encryption and decryption processes are explained. Experimental and security results are

discussed in Section 4. Finally, the conclusion is given in Section 5.

2. The basic theory of the proposed algorithm

2.1 A logestic chaotic map

In this paper, the permutation process utilizes 2D logistic map which can be defined as follows:

$$\begin{cases} x_{i+1} = \mu_1 x_i (1-x_i) + \gamma 1 y_i^2 \\ y_{i+1} = \mu_2 y_i (1-y_i) + \gamma_2 (x_i^2 + x_i y_i) \end{cases} \quad (1)$$

When $2.75 \le \mu_1 \le 3.4$, $2.75 \le \mu_2 \le 3.45$, $0.15 \le \gamma_1 \le 0.21$, $0.13 \le \gamma_2 \le 0.15$, the system is in a chaotic state and can achieve two chaotic sequences in the region (0,1]. Thanks to the system parameters μ_1 and μ_2 having a bigger value range, We set $\mu_1 = 2.85$ and $\mu_2 = 3.25$, the other parameters can be seen as secret keys.

2.2 The hyper-chaotic map

In the proposed encryption algorithm, the diffusion process uses the hyper chaotic-chaotic map generated from Chen's system, which is modeled by [13]:

$$\begin{cases} \dot{x}_1 = a(x_2 - x_1) \\ \dot{x}_2 = -x_1 x_3 + d x_1 + c x_2 - x_4 \\ \dot{x}_3 = x_1 x_2 - b x_3 \\ \dot{x}_4 = x_1 + k \end{cases} \quad (2)$$

Where a, b, c, d and k are parameters, when a=36, b=3, c=28, d=-16 and $-0.7 \le k \le 0.7$, the system is hyper-chaotic. With paarmeters a=36, b=3, c=28, d=-16 and k=0.2, its Lyapunov exponents are $\lambda_1 = 1.552, \lambda_2 = 0.023, \lambda_3 = 0, \lambda_4 = -12.573$.

3. The proposed encryption algorithm

The complete image encryption processes consist of two parts, as shown in Figure 1.

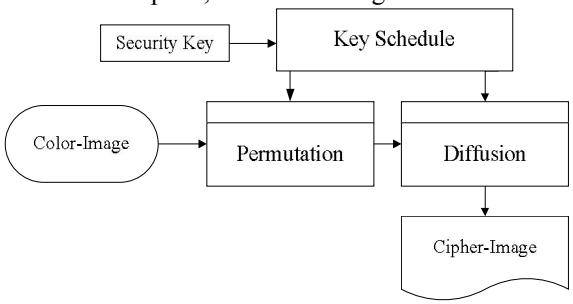

Figure1. Block diagram of the color image encryption.

3.1. Key schedule process

In order to increase the security of the proposed algorithm, a 256 bit-long secret key is used to generate the initial inputs and parameters of the chaotic system by making some algebraic transformations to the key. This key is divided into 8-bit blocks k_i referred to as session keys. The 256-bit external secret-key K is given by:

$$K = k_1, k_2, \cdots, k_{16} \quad (3)$$

The initial cinditions for logistic and hyper chaotic maps are then derived as follows:

$$x_0 = \left(\left((k_1 \oplus k_2 \oplus k_3 \oplus k_4) + \sum_{i=1}^{i=16} k_i \right) mod\ 256 \right) / 256 \quad (4)$$

$$y_0 = \left(\left((k_5 \oplus k_6 \oplus k_7 \oplus k_8) + \sum_{i=1}^{i=16} k_i \right) mod\ 256 \right) / 256 \quad (5)$$

$$\gamma 1 = \left(\left((k_9 \oplus k_{10} \oplus k_{11} \oplus k_{12}) + \sum_{i=1}^{i=16} k_i \right) mod\ 256 \right) / 256 \quad (6)$$

$$\gamma_2 = \left(\left((k_{13} \oplus k_{14} \oplus k_{15} \oplus k_{16}) + \sum_{i=1}^{i=16} k_i \right) mod\ 256 \right) / 256 \quad (7)$$

$$\dot{x}_1 = \left(\left((k_{17} \oplus k_{18} \oplus k_{19} \oplus k_{20}) + \sum_{i=1}^{i=16} k_i \right) mod\ 256 \right) / 256 \quad (8)$$

$$\dot{x}_2 = \left(\left((k_{21} \oplus k_{22} \oplus k_{23} \oplus k_{24}) + \sum_{i=1}^{i=16} k_i \right) mod\ 256 \right) / 256 \quad (9)$$

$$\dot{x}_3 = \left(\left((k_{25} \oplus k_{26} \oplus k_{27} \oplus k_{28}) + \sum_{i=1}^{i=16} k_i \right) mod\ 256 \right) / 256 \quad (10)$$

$$\dot{x}_4 = \left(\left((k_{29} \oplus k_{30} \oplus k_{31} \oplus k_{32}) + \sum_{i=1}^{i=16} k_i \right) mod\ 256 \right) / 256 \quad (11)$$

3.2 Encryption process

The detailed encryption steps for the proposed algorithm are as follows:

At first, we convert the color image P into its R, G and B components and the size of each color's (R, G or B) matrix is W × H. The detailed encryption algorithm is described as follows:

1. In order to permute the pixels of color components, combine the R, G and B matrices vertically and get matrix P1 with 3W rows and H columns and apply the external 256-bit secret key.

2. Generate the initial conditions of logistic chaotic map according to section 3.1 and iterate the two-dimensional logistic map m + 3W times, discard the former m values to avoid harmful effects and get $\{x_1, x_2, ..., x_{3W}\}$ and $\{y_1, y_2, ..., y_H\}$.

3. Sort the above values and get the new set $\{\bar{x}_1, \bar{x}_2, ..., \bar{x}_{3W}\}$ and $\{\bar{y}_1, \bar{y}_2, ..., \bar{y}_H\}$.

4. Find the position of values $\{\bar{x}_1, \bar{x}_2, ..., \bar{x}_{3W}\}$ and $\{\bar{y}_1, \bar{y}_2, ..., \bar{y}_H\}$ in $\{x_1, x_2, ..., x_{3W}\}$ and $\{y_1, y_2, ..., y_H\}$, respectively. Then,

we get the transform positions $T_x = \{t_1^x, t_2^x, ..., t_{3W}^x\}$ and $T_y = \{t_1^y, t_2^y, ..., t_H^y\}$, where $\bar{x}_i$ and $\bar{y}_i$ exactly the value of $x_{t_i^x}$ and $y_{t_i^y}$, T is used as the transform matrix.

5. Rearrange the location value of P_1, that is, move the value of P_1 (t_1^x, t_1^y) to row 1 and column 1, the value of P_1 (t_1^x, t_2^y) to row 1 and column 2, etc. until all the elements of P_1 have been moved. We get transformed matrix P'_1 whose elements in P'_1 range from 0 to 255.

6. Generate the initial conditions of hyper-chaotic map according to section 3.1 and iterate the hyper-chaotic map $\frac{3W \times H}{4}$ times and save results of output equations in a new vector E whose size is $3W \times H$.

7. We convert P_1 to vector I which has a length of $3W \times H$, and then apply the encryption transformation-on and iterate it $3W \times H$ times as the following equations:

$$C_i = I_i + E_i + Sum(C_{i-1}) + Sum(I_{i+1}) \quad (12)$$

Where C_i is the current encrypted value, I_i is the current plain value, $\mathrm{Sum}(C_{i-1})$ is the summation of encrypted values before i^{th} pixel, $\mathrm{Sum}(I_{i+1})$ is the summation of plain values after i^{th} pixel, E_i is the value of i^{th} key stream, $I_0 = 0$ and $C_0 = 0$.

3.3. Proposed decryption algorithm

The decryption process is similar to the encryption process just with reversed steps. Therefore, the remark should be considered in the decryption process as follows:

Remark. We can rewrite Eq. (12) to get the pixels' values in the RGB components:

$$I_i = C_i - E_i - Sum(C_{i-1}) - Sum(I_{i+1}) \quad (13)$$

Since the encryption and decryption processes have the similar structure, the same security key should be used. Hence, according to section 3.1, it is possible to set initial conditions and parameters.

4. Performance and security analysis

4.1 Statistical analysis

4.1.1 Histogram of Encrypted Image

Image histogram is a very important feature in image analysis. From Figure 2 it is obvious that the histogram of components of the encrypted image are nearly uniform and significantly different from the histogram of the color components of the original image. Hence, it does not provide any clue to employ any statistical analysis attack on the encrypted image.

4.1.2 Correlation of Two Adjacent Pixels

We have analyzed the correlation between two horizontally adjacent pixels, two vertically adjacent pixels, and two diagonally adjacent pixels in an image. 2000 pairs of two adjacent (in vertical, horizontal, and diagonal direction) pixels from plain-image and ciphered image were randomly selected and the correlation coefficients were calculated using [15]. Figure 3 shows the correlation distribution of two adjacent pixels in the plain-image and that in the cipher-image. Results for horizental, vertical and diagonal directions were obtained, which are shown in Table 1. Table 2 compares results of the propose method with other reported methods in the literature.

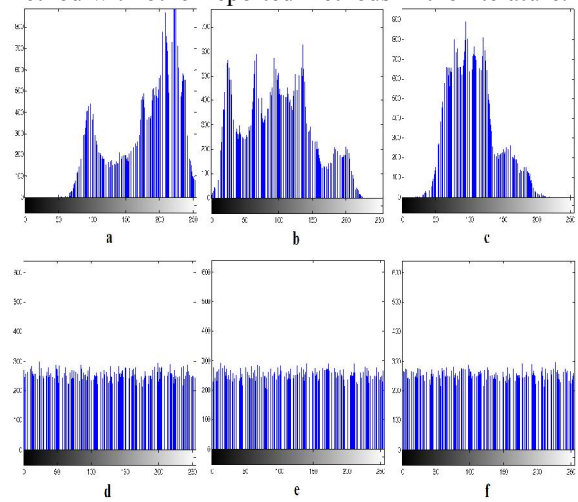

Figure 2. Histogram of the original image of Lena in the (a) red (b) green (c) blue, components, Histogram of the encrypted image of Lena in the (d) red (e) green (f) blue, components.

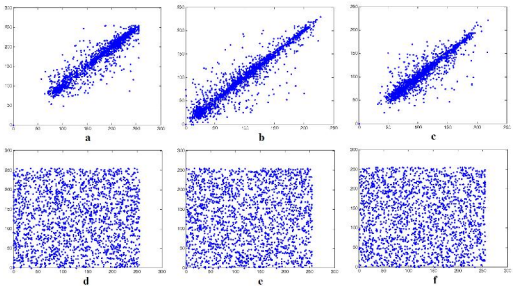

Figure 3. Correlation analysis of two horizontally adjacent pixels: Frames (a), (b) and (c), respectively, show the distribution of two horizontally adjacent pixels in the plain image of Lena in the (a) red (b) green (c) blue, components. Frames (d), (e) and (f), respectively, show the distribution of two horizontally adjacent pixels in the encrypted image of Lena in the (a) red (b) green (c) blue, components; obtained using the proposed scheme.

Table 1. Correlation coefficients of two adjacent pixels in two images.

Scan direction	Plain-image	Cipher-image
Horizontal	0.975783	0.001709
Vertical	0.985910	0.001986
Diagonal	0.960456	0.000626

Table 2. Performance analysis of the proposed method with recent methods using Lena image correlation coefficients of pairs of adjacent pixels in different directions.

Direction	Horizontal	Vertical	Diagonal
Plain-image	0.97578	0.98591	0.960456
Proposed	0.070365	0.00198	0.006262
Ref. [4]	0.00534	0.00846	0.003557
Ref. [5]	0.00681	0.00782	0.003233
Ref. [14]	0.00055	0.00083	0.001124
Ref. [15]	0.01270	0.01900	0.001200

4.2. Sensibility analysis

4.2.1 Differential sensitivity analysis

In order to test the influence of changing a single pixel in the original image on the encrypted image, we have measured the number of pixels change rate by calculating the Number of Pixel Change Rate ($NPCR$) and the Unified Average Changing Intensity ($UACI$) for the two encrypted images using [15].

Table 3. Average NPCR and UACI values for standard images.

Image	Test	256×256	512×512	1024×1024
Splash	NPCR	0.996466	0.996760	0.996824
	UACI	0.334792	0.334529	0.335279
Tiffany	NPCR	0.996424	0.996710	0.996732
	UACI	0.334725	0.334661	0.335236
Baboon	NPCR	0.996567	0.996560	0.996817
	UACI	0.334699	0.334471	0.335251
Lena	NPCR	0.996425	0.996766	0.996722
	UACI	0.334524	0.334505	0.335200
Airplane	NPCR	0.996324	0.996774	0.996727
	UACI	0.334634	0.334559	0.335281
Sailboat	NPCR	0.996559	0.996704	0.996815
	UACI	0.334686	0.334578	0.335213
Pepper	NPCR	0.996428	0.996694	0.996717
	UACI	0.334700	0.334650	0.335213
House	NPCR	0.996045	0.996679	0.996853
	UACI	0.334762	0.334692	0.335273

The several images which meet different size are employed respectively. The results in Table 3 are the average value of $NPCR$ and $UACI$, which is performed 100 trials. In proposed method, according to the encryption transformation used in Eq. (12), encrypting each pixel of the original image depends on previous encrypted values; therefore, as a result of this dependency a swift change in the original image.

Table 4. Comparison of the average NPCR and UACI values for proposed algorithm and the other algorithms.

Algorithm	NPCR	UACI
Proposed	0.996759	0.334838
Ref. [4]	0.995953	0.333586
Ref. [5]	0.996052	0.334119
Ref. [14]	0.996721	0.334904
Ref. [15]	0.996521	0.334825

will result in a significant change in the ciphered image. Table 4 compares the average value of $NPCR$ and $UACI$ for our proposed scheme, Ref. [4], Ref. [5], Ref. [14] and Ref. [15]. As it is obvious from the simulation results, the proposed cryptosystem achieves high performance by having $NPCR > 0.99675$ and $UACI > 0.33483$ and can well resist the known-plaintext and the chosen-plaintext attacks.

4.2.2 Security key analysis

Key sensitivity analysis has been performed for the proposed image encryption algorithm and the results are summarized as follows: Let us assume that two 32-character cipher keys are used as key 1: "1d,dc,23,63,ef,1d,54,67,98,ef,ee,cd,a1,b5,11,95,05,af ,34,59,5a,51,fe,b2,a4,dc,ee,49,23,30,43,45" and key 2: "1d,dc,23,63,ef,1d,54,67,98,ef,ee,cd,a1,b5,11,95, 05,af,34,59,5a,51,fe,b2,a4,dc,ee,49,23,30,43,46". A color image is first encrypted using key 1 and then key 2.

Now, these two ciphered images, encrypted by two slightly different keys, are compared. This test shows that although the two keys are different in only one bit, there is a difference of up to 0.9961 in terms of pixel grey-scale values between the image encrypted by key 1 and the image encrypted by key 2 (See Figure 4). As discussed in Section 3, transformations used in Eqs. (4) - (11) are designed so that the initial conditions and the parameters of the chaotic systems are greatly sensitive to the change even in one bit of secret key; as a result, the proposed scheme can resist against brute-force attack. The average pixel differences of some well-known images are tabulated in Table 5 using several random keys. All the cases with one-bit difference are computed for each key. Results indicate that the sensitivity obtained in the proposed method is very close to the expected value of the pixel difference on two randomly generated images ($NPCR = 0.996122, UACI = 0.334701$).

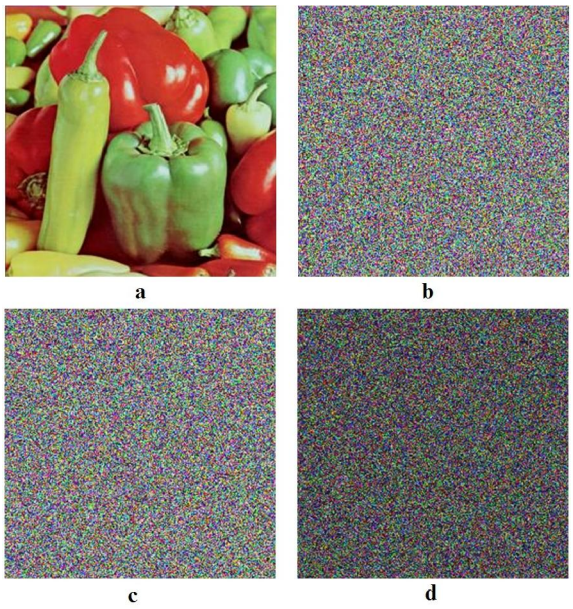

Figure 4. (a) Original image, (b) Encrypted image with key 1, (c) Encrypted image with key 2, (d) Difference image.

Table 5. Conparison of key sensitivity average values for several standard images.

Image	Test	256×256	512×512	1024×1024
Splash	NPCR	0.996032	0.996166	0.996142
	UACI	0.334530	0.334938	0.334679
Tiffany	NPCR	0.996128	0.996096	0.996128
	UACI	0.334821	0.334843	0.334632
Baboon	NPCR	0.996129	0.996160	0.996112
	UACI	0.335071	0.334682	0.334680
Lena	NPCR	0.996110	0.996152	0.996167
	UACI	0.334630	0.334558	0.334629
Airplane	NPCR	0.996047	0.996130	0.996119
	UACI	0.334730	0.334700	0.334629
Sailboat	NPCR	0.996134	0.996186	0.996129
	UACI	0.334872	0.334610	0.334718
Pepper	NPCR	0.996067	0.996169	0.996115
	UACI	0.334744	0.334920	0.334586
House	NPCR	0.996093	0.996148	0.996114
	UACI	0.334280	0.334539	0.334620

4.3 Randomness tests for the cipher

In this paper, we have used ENT test suite for testing the randomness of the cipher. The main goal of this test is to focus on different types of possible randomness in the sequence. To test the cipher randomness, a lot of initial keys are used. The results of the test is shown in Table 6. By analysing these results, it can be concluded that our proposed image encryption algorithm can successfully pass ENT test. Hence, we can claim that the generated ciphers in our cryptosystem are quite stochastic.

Table 6 Max grade of ENT test suite for the Lena image

Test name	Average value	Result
Entropy	7.999981	Success
Arithmetic mean	127.4745	Success
Monte Carlo	3.141354	Success
Chi square	254.7921	Success
SCC	0.000013	Success

5. Conclusions

In this paper, a new permutation and diffusion method based on chaotic maps for mage encryption is proposed. To achieve high security and high sensitivity, our proposed scheme presents the permutation process which uses two-dimensional logistic map to confuse the pixels of R, G and B components at the same time. In this cryptosystem, the diffusion process is designed in such a way to strengthen the security and sensitivity of cryptosystem.

We perform some security analysis to prove that the key space of the new algorithm is sufficiently large thus making the brute-force attack infeasible. Simulation results demonstrate that satisfactory performance (sensitivity and security) is achievable in our proposed algorithm.

References

1. Chen LS, Zheng GX. Multimedia Security Handbook. ed: CRC Press, 2005.
2. Cheng H, Li X. Partial encryption of compressed images and videos. IEEE Transactions on Signal Processing 2000; 48: 2439 -2451.
3. Fridrich J. Symmetric ciphers based on two dimensional chaotic maps. International Journal of Bifurcation Chaos 1998;8:1259-1284.
4. Lian S, Sun J, Wang Z. A block cipher based on a suitable use of the chaotic standard map. Chaos, Solitons and Fractals 2005;26:117-129.
5. Wong KW, Kwok BS, Law WS. A fast image encryption scheme based on chaotic standard map. Physics Letters A 2008;372:2645-2652.
6. Wang Y, Wong K, Liao X, Xiang T, Chen G. A chaos-based image encryption algorithm with variable control parameters. Chaos, Solitons and Fractals 2009;41:1773-1783.

7. Wang Y, Wong K, Liao X, Chen G. A new chaos-based fast image encryption algorithm. Applied Soft Computing 2011;11:514-522.
8. Lou DC, Sung CH. A steganographic scheme for secure communications based on the chaos and Euler theorem. Transactions on Multimedia 2004;6:501-509.
9. Feng H, Yong F. Security analysis of image encryption based on twodimensional chaotic maps and improved algorithm. Frontiers of Electrical and Electronic Engineering in China 2009;4:5-9.
10. Cokal C, Solak E. Cryptanalysis of a chaos-based image encryption algorithm. Physics Letters A 2009;373:1357-1360.
11. O. Lafe, "Data compression and encryption using cellular automata transform," Engineering Applications of Artificial Intelligence 1998;10:581-591.
12. Chen RJ, Lai JL. Image security system using recursive cellular automata substitution. Pattern Recognition 2007;40:1621-1631.
13. Gao T, Chen Z. A new image encryption algorithm based on hyper-chaos. Physics Letters A 2008;372:394-400.
14. Seyedzadeh SM, Mirzakuchaki S. A fast color image encryption algorithm based on coupled two-dimensional piecewise chaotic map. Signal Processing 2011;92:1202-1215.
15. Liao X, Lai S, Zhou Q. A novel image encryption algorithm based on self-adaptive wave transmission. Signal Processing 2010;90:2714-2722.

10/18/2017