Impact of Rotations in the Salsa20/8 Image Encryption Scheme

Alireza Jolfaei, *Member, IACSIT*, Abdolrasoul Mirghadri, and Ahmadreza Vizandan

***Abstract*—Salsa20 uses a single set of rotation distances within its core function. To our knowledge, no design rationales on the choice of rotation distances are given on the Salsa20 core function. This is the first paper that analyzes rotations influence over the output sequence in the Salsa20 image encryption scheme. We focus on the Salsa20/8 image encryption scheme and use recent developments in the analysis of the Salsa20 core function to evaluate the security implications of using different sets of rotation distances in the core function. Finally, we give some observations on the some sets of rotation distances used in the Salsa20/8 image encryption scheme.**

***Index Terms*—Salsa20, image encryption, rotation distance.**

I. Introduction

Along with the rapid growth of image transmission through communication substructures such as mobile networks and the Internet, the security of digital images has become a major concern. So, Image encryption is urgently needed to make visual communication incomprehensible to those who do not possess the right keys.

Image encryption is quite different from text encryption due to some intrinsic features of images such as bulky data capacity and high correlation among pixels, which are generally difficult to handle by traditional techniques. In the last two decades, a growing number of image encryption algorithms, adopting some related nonlinear theories, have been proposed for use in cryptographic applications. Most of these systems were subject to cryptanalytic attacks and many of them were shown to suffer from a lack of security.

Recent studies have shown that stream ciphers are potential candidates for image encryption [1]–[3]. Stream ciphers play an important role in cryptographic practices such as medical and military applications. A critical point in the design of a stream cipher is to generate a long unpredictable binary sequence from a short secret key. Unpredictable sequences are desirable in cryptography because it is rather impossible, given a reasonable segment of binary sequence, to find out more about them. Generally, the security of stream ciphers cannot be proved. Instead, the trust in a cipher is merely based on the fact that no weaknesses have been found after a long and thorough evaluation phase.

Recently, as a response to the lack of efficient and secure stream ciphers, The European Network of Excellence for Cryptography, ECRYPT, issued a call for stream cipher proposals named eSTREAM project to find suitable stream ciphers for widespread adoption. This is a good opportunity for the cryptographic community to settle on a new encryption standard that simultaneously provides higher confidence and higher speed than AES. The hope is that, by doing so, the algorithms and methods that are likely to be standardized at some point during the next years or so will be subjected to rigorous inspection by the cryptographic community. Salsa20, one of the e STREAM candidates, is a synchronous stream cipher proposed by Bernstein [4]. The author justified the use of very simple operations (addition, XOR, constant-distance rotation) and the lack of multiplication or S-boxes. This helps to develop a very fast primitive that is also, by construction, immune to timing attacks. There are still several open questions about the design of Salsa20 core function. One of these questions is about choice of rotation distances in the design of Salsa20 core function. The rotations provide the diff usion between the various bit positions in the state words. In [5], Bernstein answered to this question and stated that the exact choice of distances does not seem very important.

In [1], we surveyed a successfully efficient implementation of three variants of Salsa20 stream cipher: Salsa20/8, Salsa20/12 and Salsa20/20, for digital image encryption and compared the results. In this paper, we describe an observation about the effect of choosing different set of rotation distances in the Salsa20/8 image encryption algorithm. We show that choosing different rotation distances causes a change in the statistical properties of output cipher-image and alters the encryption speed. To our knowledge this is the first article which deals with the issue of rotation distances in Salsa20 core function. The outline and main contributions of this article are as follows. In Section 2, we give a short description of the Salsa20/8 image encryption scheme. Afterwards, in Section 3, we describe our test method. In Section 4, we analyze the efficiency of the Salsa20/8 image encryption scheme using different sets of rotation distances. Finally, Section 5 concludes the paper.

A. Used Notation and Terminology

Table I contains a description of symbols used throughout this article.

TABLE I: Used Notation

Notation	Description
$x \oplus y$	Addition of x and y modulo 2 (XOR)
$x + y$	Addition of x and y modulo 2^{32}
$x \lll n$	Bit-rotation of x by n positions to the left, $0 \leq n \leq 31$
$\lfloor x \rfloor$	The largest integer not greater than x
erfc (x)	The complementary error function for the value of x
igamc (a, x)	The incomplete gamma function as defined in [6]
P-value	The probability of obtaining a test statistic at least as extreme as the one that was actually observed, assuming that the null hypothesis is true
Pr (x)	Probability of observing event x
H	Number of rows in the image matrix
W	Number of columns in the image matrix

Manuscript received August 20, 2012; revised October 10, 2012.

A. Jolfaei is with the Department of Information and Communication Technology (ICT), Griffith University, Gold Coast campus, Queensland, Australia (e-mail: Jolfaei@yahoo.com).

A. Mirghadri and A. Vizandan are with the the Faculty and Research Center of Communication and Information Technology, IHU, Tehran, Iran.

II. Salsa20/8 Image Encryption Algorithm

The Salsa20/8 image encryption algorithm is defined as follows [1]:

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} = quarterround \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} \rightarrow \begin{cases} y_1 = x_1 \oplus ((x_0 + x_3) <<< 7) \\ y_2 = x_2 \oplus ((y_1 + x_0) <<< 9) \\ y_3 = x_3 \oplus ((y_2 + y_1) <<< 13) \\ y_0 = x_0 \oplus ((y_3 + y_2) <<< 18) \end{cases} \quad (1)$$

$$\begin{pmatrix} x_0 & x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 & x_7 \\ x_8 & x_9 & x_{10} & x_{11} \\ x_{12} & x_{13} & x_{14} & x_{15} \end{pmatrix} \xrightarrow{Rowround\ Operations} \begin{cases} (y_0, y_1, y_2, y_3) = quarterround(x_0, x_1, x_2, x_3) \\ (y_5, y_6, y_7, y_4) = quarterround(x_5, x_6, x_7, x_4) \\ (y_{10}, y_{11}, y_8, y_9) = quarterround(x_{10}, x_{11}, x_8, x_9) \\ (y_{15}, y_{12}, y_{13}, y_{14}) = quarterround(x_{15}, x_{12}, x_{13}, x_{14}) \end{cases} \quad (2)$$

$$\begin{pmatrix} x_0 & x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 & x_7 \\ x_8 & x_9 & x_{10} & x_{11} \\ x_{12} & x_{13} & x_{14} & x_{15} \end{pmatrix} \xrightarrow{Columnround\ Operations} \begin{cases} (y_0, y_4, y_8, y_{12}) = quarterround(x_0, x_4, x_8, x_{12}) \\ (y_5, y_9, y_{13}, y_1) = quarterround(x_5, x_9, x_{13}, x_1) \\ (y_{10}, y_{14}, y_2, y_6) = quarterround(x_{10}, x_{14}, x_2, x_6) \\ (y_{15}, y_3, y_7, y_{11}) = quarterround(x_{15}, x_3, x_7, x_{11}) \end{cases} \quad (3)$$

$$Salsa20/8(x) = x + (rowround(columnround(x)))^4 \quad (4)$$

Image matrix is a binary sequence of $8 \times H \times W$ length. The plain-image is first partitioned into blocks of 8×8 pixels. Then, Salsa20/8 algorithm generates a set of pseudo-random 64-byte stream known as keystream, equal length to image matrix size. Afterwards, each 64-byte block is XOR-ed with its corresponding 64-byte block in the plain-image as follows:

$$Plain\_image(i) \oplus Keystream(i) = Cipher\_image(i) \quad (5)$$

where, $i \in \{0, 1, 2, ..., 2^{64} - 1\}$. This procedure is shown in Fig. 1. As it is shown in Fig. 1, each 8×8 block (64-byte) of keystream is XOR-ed with its corresponding 8×8 block of plain-image to produce cipher-image. For decryption, cipher-image is XOR-ed with keystream. The Salsa20 key is a uniform random sequence of bytes, and the same nonce is never used for two different blocks of messages.

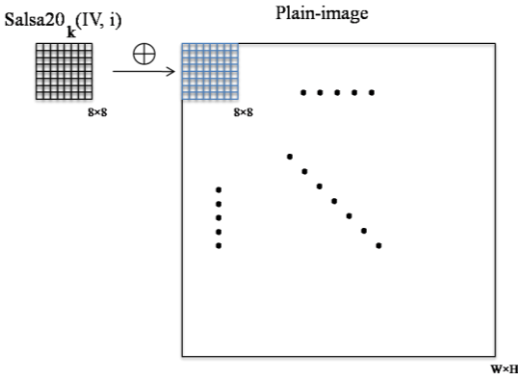

Fig. 1. Salsa20/8 image encryption scheme

III. Test Method

In this section, we briefly describe our test method. A gray image of size 512×512 (= 2,097,152 bits), called "Couple", is selected as the plain-image that is shown in Fig. 2. The experiments are all performed using MATLAB 7.10 on a personal computer (PC) with a 2.0 GHz Intel dual-core processor and 2 GB RAM. We use K = 0x0123456789abcdef0123456789abcdef as the secret key and IV = 0 as the initial vector. The default rotation distances in the Salsa20/8 encryption scheme are 7, 9, 13 and 18. So, we encrypt the plain-image using all other possible choices of fewer rotation distances in the encryption scheme, i.e. $7 \times 9 \times 13 \times 18 = 14742$ number of options, and analyze the randomness of corresponding cipher-images' bit strings using frequency test and frequency test within a block. Then, we analyze the uniformity of cipher-images using Pearson's chi-square test. Our purpose is to find out whether there is any other possible choice of rotation distances that leads to a faster and more efficient design compared to the design with the default rotation constants.

Fig. 2. Plain-image

We used the following code in our test method to encrypt plain-image using all possible fewer rotation distances.

```
Counter = 0;
for i = 1:7
 for ii = 1:9
  for iii = 1:13
   for iiii = 1:18
   rotation distance=(i,ii,iii,iiii);
    Cipher=Salsa20/8(plain-image,
                rotation distance);
    Counter = Counter + 1;
  end
  end
 end
end
```

A. Frequency Test (Monobit Test)

This test focuses on the proportion of zeroes and ones for the entire cipher-image bit sequence. So, it counts the number of zeros and ones in a sequence and determines whether the proportion of zeros and ones in a sequence are approximately the same as would be expected for a truly random sequence. Note that given a random generated bit string, we would expect approximately half the bits in the string to map to ones and approximately half to map to zeros. Let the length of the encrypted bit string be n and let the generated bit sequence be given as $\varepsilon = \varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n$. Where $\varepsilon_i \in \{0, 1\}$. The entire encrypted bit strings are added together to produce $S_n = X_1 + \ldots + X_n$, where $X_i = 2\varepsilon_i - 1$. Now, the P-value can be computed as follows [6]:

$$P-value = erfc\left(\frac{S_n}{\sqrt{2n}}\right) \quad (6)$$

If the computed *P*-value is < 0.01, then conclude that the sequence is non-random. Otherwise, conclude that the sequence is random. Fig. 3 shows the P-value of monobit test for the bit string of the final encrypted image of the cryptosystem under study. Due to saving a large scale data, it is rather difficult to distinguish the exact results from the figure. So, for a better comparison, we have selected some set of rotation distances and listed their corresponding monobit test *P*-values in Table II. According to the monobit test results, there are other sets of rotation distances, if chosen; we can get a random bit string with fewer rotations.

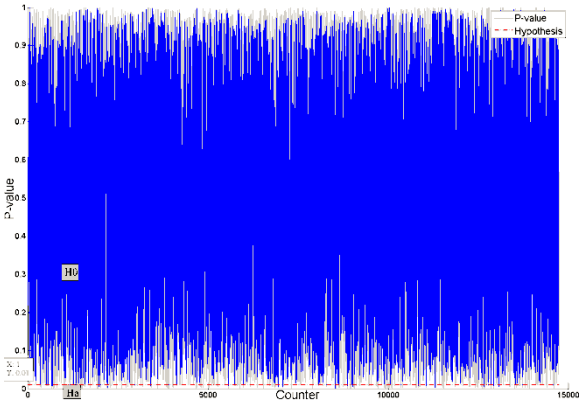

Fig. 3. Monobit test results

TABLE II: MONOBIT TEST RESULTS FOR SOME SETS OF ROTATION DISTANCES

Rotation distance set	Counter	P-value	Result
[7, 9, 13, 18]	14742	0.1020	Pass
[7, 8, 12, 16]	14488	0.5334	Pass
[7, 8, 8, 12]	14412	0.0853	Pass
[7, 7, 2, 7]	14065	0.3693	Pass
[6, 9, 6, 2]	12494	0.8445	Pass
[6, 7, 10, 15]	12111	0.5489	Pass
[4, 5, 6, 7]	7351	0.2306	Pass
[2, 8, 2, 4]	3766	0.3227	Pass
[2, 6, 9, 8]	3428	0.0035	*Fail*
[1, 3, 2, 12]	498	0.9329	Pass
[1, 1, 4, 5]	59	0.4239	Pass
[1, 1, 3, 4]	40	0.0070	*Fail*
[1, 1, 1, 1]	1	0.0152	Pass

B. *Frequency Test within a Block*

This test focuses on the proportion of one's within *M*-bit blocks of cipher-image bit sequence. The purpose of this test is to determine whether the frequency of ones in an *M*-bit block is approximately $^{M}/_{2}$, as would be expected under an assumption of randomness. For block size $M = 1$, this test degenerates to the monobit test. Let the length of the encrypted bit string be n and let the generated bit sequence be given as $\varepsilon = \varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n$. The chi-square is computed as follows [6]:

$$\chi^2_{test} = 4\sum_{i=1}^{\left\lfloor \frac{n}{M} \right\rfloor}\left(\sum_{j=1}^{M}\varepsilon_{(i-1)M+j} - \frac{M}{2}\right)^2 \quad (7)$$

Now, the P-value can be computed as follows [6]:

$$P-value = \text{igamc}\left(\left\lfloor \frac{n}{M} \right\rfloor, \frac{\chi^2_{test}}{2}\right) \quad (8)$$

If the computed *P*-value is < 0.01, then conclude that the sequence is non-random. Otherwise, conclude that the sequence is random. This test depends on the passing of monobit test. Let $M = 3$. Fig. 4 shows the *P*-value of frequency test within a block for the bit string of the final encrypted images of the cryptosystem under study. Due to saving a large scale data, it is rather difficult to distinguish the exact results from the figure. So, we have selected some set of rotation distances and listed their corresponding *P*-values in Table III. According to the results of the frequency test within a block, there are a lot of rotation distance sets, if chosen; we can get a random bit string with fewer rotations.

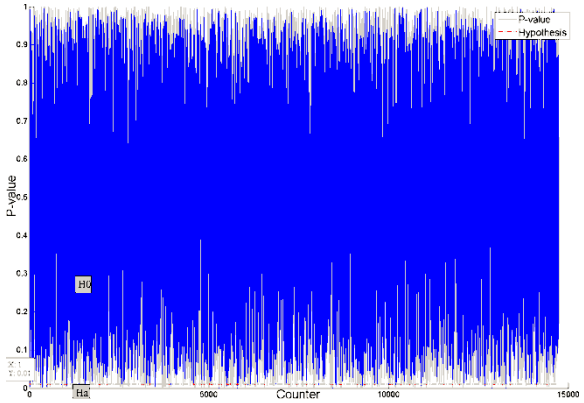

Fig. 4. Results of frequency test within a block

TABLE III: THE RESULTS OF THE FREQUENCY TEST WITHIN A BLOCK FOR SOME SETS OF ROTATION DISTANCES

Rotation distance set	Counter	*P*-value	Result
[7, 9, 13, 18]	14742	0.2793	Pass
[7, 8, 12, 16]	14488	0.8982	Pass
[7, 8, 8, 12]	14412	0.0040	*Fail*
[7, 7, 2, 7]	14065	0.4009	Pass
[6, 9, 6, 2]	12494	0.1466	Pass
[6, 7, 10, 15]	12111	0.1802	Pass
[4, 5, 6, 7]	7351	0.2049	Pass
[2, 8, 2, 4]	3766	0.5242	Pass
[2, 6, 9, 8]	3428	*Monobit test failed*	
[1, 3, 2, 12]	498	0.0003	*Fail*
[1, 1, 4, 5]	59	0.5051	Pass
[1, 1, 3, 4]	40	*Monobit test failed*	
[1, 1, 1, 1]	1	0.1846	Pass

C. *Chi-square Test*

Digital image is a two-dimensional matrix and its smallest unit is a byte (pixel) not a bit. Also, in the image matrix each pixel is adjacent to 8 surrounding pixels. So, unlike the textual data, there exists a lot of correlation among image pixels. Despite the randomness of cipher-image bit sequence, there should be no appearance of pattern or textured zone recognized by visual inspection. Appearance of homogeneous zones in the cipher-image is a security weakness and prevents image to reach the maximal entropy.

In statistics and probability theory, the discrete uniform distribution is an equally likely probability distribution whereby every one of *n* observed values has equal probability $^{1}/_{n}$. Uniformity caused by an encryption function may be justified quantitatively by the Pearson's chi-square test [7]. The chi-square distribution is a very powerful statistical test. Its distribution can be used to compare the goodness-of-fit of the observed frequencies of events to their expected frequencies under a hypothesized distribution [8]. The test statistics for this test is given by

$$\chi^2_{test} = \sum_{k=0}^{255} \frac{(o_k - e_k)^2}{e_k} \quad (9)$$

where k is the number of gray levels (256), o_k is the observed occurrence frequencies of each gray level (0–255), and e_k is the expected occurrence frequency of each gray level. For example, if plain-image size is $H \times W$ then $e_k = H \times W/256$. Assuming a significant level of 0.01, χ^2 (255, 0.01) = 310.4574. For any cipher-image if $\chi^2_{test} < \chi^2$ (255, 0.01), then it implies that the null hypothesis is not rejected and the distribution of the encrypted image histogram is uniform. Fig. 5 shows the chi-square test value for the final encrypted image of the cryptosystems under study. Due to saving a large scale data, it is rather difficult to distinguish the exact results from the figure. So, for a better comparison, we have selected some set of rotation distances and listed their corresponding chi-square values in Table IV. Test results show that there are other choices of rotation distances, if chosen; we can get a random cipher-image with fewer rotations.

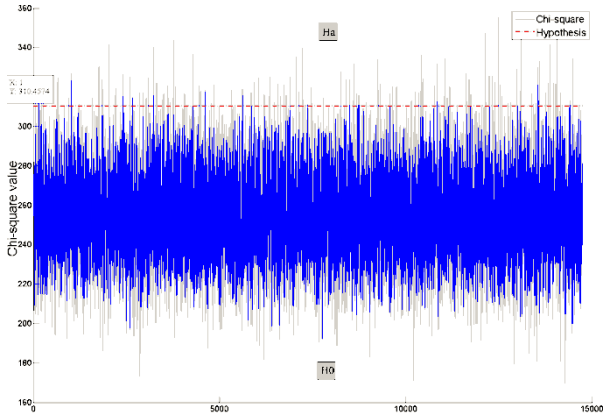

Fig. 5. Chi-square test value

TABLE IV: CHI-SQUARE TEST VALUES FOR SOME SETS OF ROTATION DISTANCES

Rotation distance set	Counter	Chi-square	Result
[7, 9, 13, 18]	14742	245.5020	Pass
[7, 8, 12, 16]	14488	253.0605	Pass
[7, 8, 8, 12]	14412	*Frequency test within a block failed*	
[7, 7, 2, 7]	14065	353.3379	*Fail*
[6, 9, 6, 2]	12494	355.1484	*Fail*
[6, 7, 10, 15]	12111	231.9316	Pass
[4, 5, 6, 7]	7351	252.6836	Pass
[2, 8, 2, 4]	3766	343.7148	*Fail*
[2, 6, 9, 8]	3428	*Monobit test failed*	
[1, 3, 2, 12]	498	*Frequency test within a block failed*	
[1, 1, 4, 5]	59	323.0176	*Fail*
[1, 1, 3, 4]	40	*Monobit test failed*	
[1, 1, 1, 1]	1	228.0234	Pass

IV. ANALYSIS

In this section, we performed a series of tests to justify and compare the efficiency of the Salsa20/8 image encryption scheme using different sets of rotation distances. The evaluation consisted of theoretical derivations and practical experimentation.

A. Entropy Analysis

Information theory is the mathematical theory of data communication and storage founded in 1949 by Shannon [9]. Shannon redefined the entropy as a measure of the amount of information in a source [10]. The concept of entropy is associated with the amount of disorder and uncertainty in a physical system. Today, the modern information theory is concerned with error-correction, data compression, cryptography, communications systems, and related topics. The entropy of an image is an estimation of randomness and it is frequently used to measure sharpness of the histogram peaks, which is directly related with better defined structural information. It is well known that the Shannon entropy $H(s)$ of a message source s can be calculated as:

$$H(s) = \sum_{i=0}^{2N-1} \Pr(s_i) \log_2 \frac{1}{\Pr(s_i)} \quad (10)$$

The entropy is expressed in bits. Let us suppose that the source emits 2^N symbols with equal probability, i.e., $s = \{s_1, s_2, ..., s_{2^N}\}$. After evaluating equation (10), we obtain its entropy $H(s) = N$, corresponding to a truly random source. Actually, given that a practical information source seldom generates random messages, in general, its entropy value is smaller than the ideal one. However, when the messages are encrypted, their entropy should ideally be 8. If the output of such a cipher emits symbols with entropy less than 8, there exists certain degree of predictability, which threatens its security. Let us consider the cipher-image of test image, the number of occurrence of each gray level is recorded and the probability of occurrence is computed. The entropy test results are listed in Table V. The values obtained are very close to the theoretical value of 8. This means that the information leakage in the encryption process is negligible and the encryption system is secure upon the entropy attack. Experiments show that there is a negligible difference between information entropy of cipher-images of surveyed image encryption algorithm using different sets of rotation distances. This shows that we can approach the maximal entropy using fewer rotations.

TABLE V: ENTROPY VALUE FOR SOME SETS OF ROTATION DISTANCES

Rotation distance set	Counter	Entropy
[7, 9, 13, 18]	14742	7.9994
[7, 8, 12, 16]	14488	7.9993
[6, 7, 10, 15]	12111	7.9992
[4, 5, 6, 7]	7351	7.9993
[1, 1, 1, 1]	1	7.9994

B. Correlation Coefficients Analysis

In the image data, each pixel is highly correlated with its adjacent pixels. An ideal encryption algorithm should produce the cipher-images with no such correlation in the adjacent pixels. Following equations are used to study the correlation between two adjacent pixels in horizontal, vertical and diagonal orientations [11].

$$r_{xy} = \frac{Cov(x,y)}{\sqrt{D(x)}\sqrt{D(y)}} \quad (11)$$

$$D(x) = \frac{1}{N}\sum_{j=1}^{N}(x_j - \frac{1}{N}\sum_{j=1}^{N}x_j)^2 \quad (12)$$

$$Cov(x,y) = \frac{1}{N}\sum_{j=1}^{N}(x_j - \frac{1}{N}\sum_{j=1}^{N}x_j)(y_j - \frac{1}{N}\sum_{j=1}^{N}y_j) \quad (13)$$

where *x* and *y* are intensity values of two neighboring pixels in the image and *N* is the number of adjacent pixels selected from the image to calculate the correlation. We consider Lena as correlation test image that is depicted in Fig. 2. Fig. 6 shows the correlation distribution of two adjacent pixels in the plain-image and cipher-image. It is observed that neighboring pixels in the plain-image are correlated too much, while there is a little correlation between neighboring pixels in the encrypted images. Table VI shows the results for correlation coefficients of surveyed cryptosystem. The correlation coefficients of cipher-images are far apart from plain-image. Results show that the Salsa20/8 image encryption scheme can dissipate the correlation among pixels using fewer rotations.

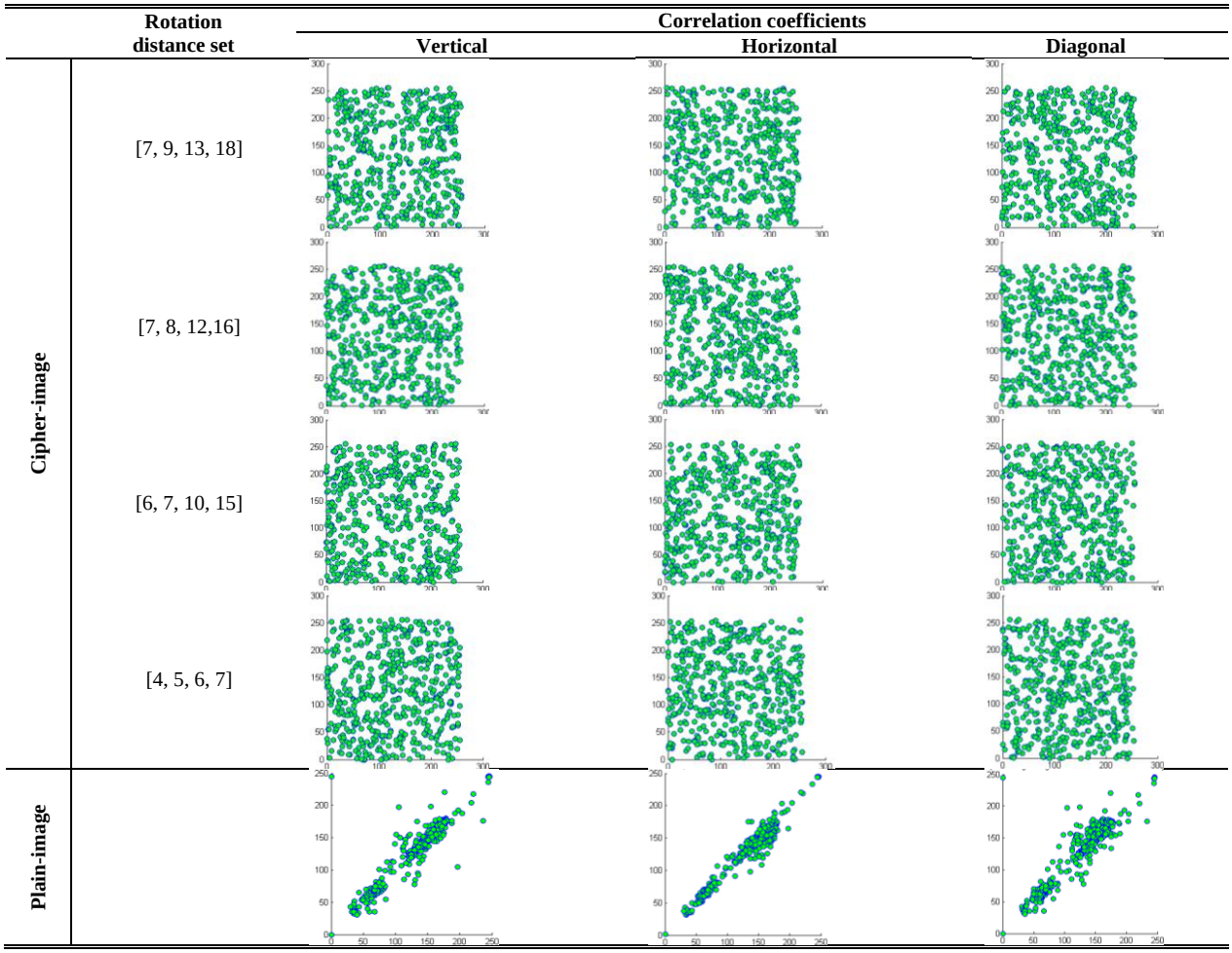

Fig. 6. Correlation analysis and distribution of two adjacent pixels

TABLE VI: CORRELATION COEFFICIENTS OF TWO ADJACENT PIXELS IN PLAIN-IMAGE AND CIPHER-IMAGE.

Image	Rotation distance set	Counter	Correlation coefficients		
			Vertical	Horizontal	Diagonal
Cipher-image	[7, 9, 13, 18]	14742	0.0182	-0.0452	-0.0217
	[7, 8, 12, 16]	14488	0.0395	-0.0714	0.0464
	[6, 7, 10, 15]	12111	-0.0294	0.0852	-0.0268
	[4, 5, 6, 7]	7351	0.0805	-0.0730	-0.0285
	[1, 1, 1, 1]	1	-0.0073	-0.0876	0.0143
Plain-image	–	–	0.9384	0.9798	0.9260

C. Performance Analysis

Apart from the security consideration, some other issues on image encryption are also important. This includes the encryption speed for real-time processes. In general, encryption speed is highly dependent on the CPU structure, memory size, OS platform, the programming language and also on the compiler options. So, it is pointless to compare the encryption speeds of two ciphers without using the same developing environment and optimization techniques. Despite of the mentioned difficulty, we have undertaken an analysis for the explicit comparison between the encryption speeds of the cryptosystems using different set of rotation distances.

We evaluated the performance of the surveyed image encryption scheme with an un-optimized MATLAB code. In addition, to improve the accuracy of timing measurements, each set of the timing tests was executed 10 times, and the average of the times was reported. Table VII summarizes the encryption speeds for the encryption scheme under study on images of different sizes. According to the performance analysis, employing fewer rotations leads to a faster encryption.

TABLE VII: EXPLICIT COMPARISON BETWEEN THE ENCRYPTION SPEEDS OF CRYPTOSYSTEMS USING DIFFERENT SETS OF ROTATION DISTANCES

Plain-image size (Pixel)	Type	Encryption time (sec)				
		[7, 9, 13, 18]	[7, 8, 12, 16]	[6, 7, 10, 15]	[4, 5, 6, 7]	[1, 1, 1, 1]
128×128	Gray	0.4541	0.4453	0.4323	0.4173	0.3992
256×256	Gray	1.6340	1.6063	1.5905	1.5146	1.4055
512×512	Gray	6.4123	6.2436	6.1193	5.8311	5.4424
1024×1024	Gray	25.2552	24.7106	24.3122	23.1552	21.6083

V. CONCLUSION AND FUTURE DIRECTIONS

In this paper, we have analyzed the effect of choosing different rotation distances in the Salsa20/8 image encryption scheme and compared them to the default set of rotation distances used in the cryptosystem. According to the performed analysis, we have finally observed that choosing most of the fewer rotations in the encryption scheme provide the same security as the default scheme. So, based on our limited analysis, the specific choice of rotation distances does not have a strong impact on the security of the Salsa20/8 image encryption scheme. The benefit of choosing fewer rotations is the encryption speed. The future research directions will be directed to a more detailed study of security analysis of the Salsa20/8 image encryption scheme using fewer rotation distances. We plan to use standard cryptanalytic tools, such as differential and linear cryptography to further assure the safety and robustness of the Salsa20/8 image encryption scheme that use fewer rotations. Finally, we hope that this paper will spur additional research in the analysis of image encryption stream ciphers.

REFERENCES

[1] A. Jolfaei and A. Mirghadri, "Survey: Image encryption using salsa20," *International Journal of Computer Science Issues*, vol. 7, no. 5, pp. 213–220, 2010.

[2] A. Jolfaei and A. Mirghadri, "Survey: Image encryption using A5/1 and W7," *Journal of Computing*, vol. 2, no. 8, 2010.

[3] H. J. Li and J. S. Zhang, "A novel chaotic stream cipher and its application to palmprint template protection," *Chinese Phys. B.*, vol. 19, no. 4, 2010.

[4] D. J. Bernstein, "The salsa20 stream cipher," in *Symmetric Key Encryption Workshop* (*SKEW 2005*), Workshop Record, 2005.

[5] D. J. Bernstein. Salsa20 Design. [Online]. Available: http://cr.yp.to/snuffle/design.pdf, 2005.

[6] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray, and S. Vo, "A statistical test suite for random and pseudorandom number generators for cryptographic applications," *NIST Special Publication*, pp. 800-22, 2010.

[7] P. L'ecuyer and R. Simard, "TestU01: A C library for empirical testing of random number generators," *ACM Transactions on Mathematical Software*, vol. 33, no. 4, Article 22, 2007.

[8] A. J. Menezes, V. P. C. Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*, New York: CRC Press, 1997.

[9] D. R. Stinson, *Cryptography: Theory and Practice*. CRC Press, ISBN: 0849385210, 1995.

[10] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 227, pp. 379–423, pp. 623–656, 1948.

[11] A. Jolfaei and A. Mirghadri, "Substitution-permutation based image cipher using chaotic henon and baker's maps," *International Review on Computers and Software* (*IRECOS*), vol. 6, no. 1, 2011.

A. Jolfaei is with the Department of Information and Communication Technology (ICT), Griffith University, Gold Coast campus, Queensland, Australia. His main research interests include multimedia security, cryptology, steganology and image processing. He is a member of IACSIT.

A. Mirghadri is an associate professor at the Faculty and Research Center of Communication and Information Technology, IHU, Tehran, Iran. His research interests include cryptography, statistics and stochastic processes. He is a member of IACSIT, ISC, ISS and IMS.

A. Vizandan is an assistant professor at the Faculty and Research Center of Communication and Information Technology, IHU, Tehran, Iran. His research interests are cryptanalysis, steganography, network security and coding theory.