Survey: Image Encryption Using Salsa20

Alireza Jolfaei[1] and Abdolrasoul Mirghadri[2]

**[1] Faculty and Research Center of Communication and Information Technology, IHU
Tehran, Iran**

**[2] Faculty and Research Center of Communication and Information Technology, IHU
Tehran, Iran**

Abstract

In present times, multimedia protection is becoming increasingly jeopardized. Therefore numerous ways of protecting information are being utilized by individuals, businesses, and governments. In this paper, we survey Salsa20 as a method for protecting the distribution of digital images in an efficient and secure way. So, we performed a series of tests and some comparisons to justify salsa20 efficiency for image encryption. These tests included visual testing, key space analysis, histogram analysis, information entropy, encryption quality, correlation analysis, differential analysis, sensitivity analysis and performance analysis. Simulation experiment has validated the effectiveness of the Salsa20 scheme for image encryption.

Keywords: *Salsa20, image encryption, test, comparison.*

1. Introduction

Along with the fast progression of data exchange in electronic way, it is important to protect the confidentiality of image data from unauthorized access. Security breaches may affect user's privacy and reputation. So, data encryption is widely used to confirm security in open networks such as the internet. Due to the substantial increase in digital data transmission via public channels, the security of digital images has become more prominent and attracted much attention in the digital world today. The extension of multimedia technology in our society has promoted digital images to play a more significant role than the traditional texts, which demand serious protection of users' privacy for all applications. Each type of data has its own features; therefore, different techniques should be used to protect confidential image data from unauthorized access. Most of the available encryption algorithms are used for text data. However, due to large data size and real time requirement, it is not reasonable to use traditional encryption methods.

Thus, a major recent trend is to minimize the computational requirements for secure multimedia distribution. Many researchers proposed different image encryption schemes to overcome image encryption problems [1, 2, 3, 4]. A classification of the proposed schemes from the open literature is given in [5]. In this paper, we survey the application of Salsa20 for image encryption. Salsa20 has an interesting blocky structure that seems to be a good choice for image encryption. A series of tests were applied to justify Salsa20's efficiency for the visual encryption applications.

The rest of the paper is organized as follows: In Section 2, first we describe symmetric ciphers then we briefly overview Salsa20 as a symmetric scheme for image encryption. In Section 3, we analyze the security of the surveyed cryptosystem and evaluate its performance through various statistical analysis, key sensitivity analysis, differential analysis, key space analysis, speed analysis, etc and compare the results. Finally, some conclusions are given in section 4.

2. Symmetric Cryptography

Symmetric encryption is the oldest branch in the field of cryptology, and is still one of the most important ones today. Symmetric cryptosystems rely on a shared secret between communicating parties. This secret is used both as an encryption key and as a decryption key. Generally, symmetric encryption systems with secure key are divided into two classes: stream ciphers and block ciphers. Stream ciphers encrypt individual characters of a plaintext message, using a time variant encryption function, as opposed to block ciphers that encrypt groups of characters of a plaintext message using a fixed encryption function [6]. Nowadays, the boundaries between block ciphers and stream ciphers are becoming blurred. So, it is difficult to tell whether a symmetric cipher is a stream or block cipher. Stream ciphers are beyond the most important encryption systems which have major applications in military,

strategic sectors and etc. They are generally faster than block ciphers in hardware, and have less complex hardware circuitry. They are also more appropriate when buffering is limited or when characters must be individually processed as they are received. Stream ciphers may also be advantageous in situations where transmission errors are highly probable, because they have limited or no error propagation.

2.1 Salsa20

As a response to the lack of efficient and secure stream ciphers, ECRYPT manages and coordinates a multiyear effort called eSTREAM to identify new stream ciphers suitable for widespread adoption. Salsa20, one of the eSTREAM candidates, is a new synchronous stream cipher proposed by Bernstein [7]. 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. Salsa20 passed to Phase 3 without major known attacks. The Core of Salsa20 is a hash function with 64-byte input and 64-byte output. The Hash function is used in counter mode as a stream cipher: Salsa20 encrypts a 64-byte block of plaintext by hashing the key, nonce, and block number and XOR-ing the result with the plaintext [8]. Salsa20/r algorithm is defined as follows [9]:

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

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

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

$$Salsa20(x) = x + (rowround\,(columnround\,(x)))^R, \quad (4)$$

where R is the number of double rounds and r is number of ciphering rounds ($r = 2R$). Bernstein proposed three variants of the Salsa20 stream cipher: Salsa20/20, which has 20 ciphering rounds; Salsa20/12, which is Salsa20 reduced from 20 rounds to 12 rounds; and Salsa20/8, which is Salsa20 reduced from 20 rounds to 8 rounds.

Image matrix is a binary sequence of $8\times H\times W$ length, where H is number of rows and W is number of columns. Salsa20 algorithm generates a set of pseudo-random 64-byte stream known as keystream, equal length to image matrix size. Then each 64-byte block is XOR-ed with its corresponding 64-byte block in the plain-image as follows:

$$Plain\_image(i) \oplus Keystream(i) = Cipher\_image(i) \quad (1)$$

where, $i \in \{0,1,2,...,2^{64}-1\}$. This procedure is shown in Fig. 1. 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 messages.

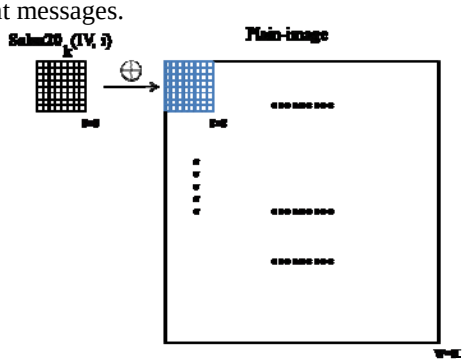

Fig. 1. Salsa20 image encryption scheme procedure is demonstrated. Each 8×8 block (64-byte) of keystream is XOR-ed with its corresponding 8×8 block of plain-image to produce cipher-image.

3. Security and Performance Analysis

A fundamental issue of all kinds of ciphers is the security. A strong cipher is capable of resisting any kind of cryptanalytic attacks including brute-force attack, statistical attack, known plain text attack and chosen-plaintext attack. Thus, a cipher of high key and plaintext sensitivity with a large key space is desirable. Besides, computational speed, size and quality of encrypted images are other important issues as well since they always include the feasibility of encryption schemes. In this section we performed a series of test to justify and compare the efficiency of the cryptosystem under study.

3.1 Visual Testing

The algorithm is applied with a 256-level gray scale TIF image of Lena that has the size of 256×256 and visual test is performed. Fig. 2 demonstrates encryption result. The original image is encrypted by using '0123456789abcdef0123456789abcdef' as secret key and '0000000000000000' as IV.

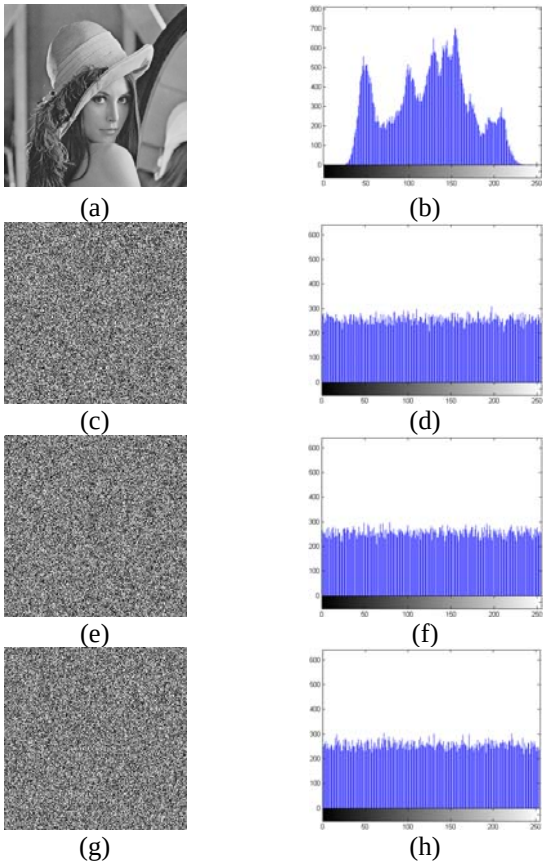

Fig. 2. Visual test result: Figures (a) and (b) depicts plain-image and plain-image histogram, respectively. Figures (c), (e) and (g) show cipher-images of Salsa20/8, Salsa20/12 and Salsa20/20, respectively. Figures (d), (f) and (h) show cipher-image histograms of Salsa20/8, Salsa20/12 and Salsa20/20, respectively.

3.2 Key Space Analysis

It is well known that a large key space is very important for an encryption algorithm to repel the brute-force attack. Salsa20 uses the hash function in a counter mode. It has 512-bit state which is initialized by copying into it 128 or 256-bit key, 64-bit nonce and counter and 128-bit constant. Then the key space can be $2^{(128+64)} = 2^{192} \approx 6.3\times10^{57}$ or $2^{(256+64)} = 2^{320} \approx 2.1\times10^{96}$. Apparently, the key space is large enough to resist all kinds of brute-force attacks.

3.3 Histogram Analysis

To prevent the leakage of information to attackers, it is important to ensure that encrypted and original images do not have any statistical similarities. The histogram analysis clarifies that, how the pixel values of image are distributed. Fig. 2 shows histogram analysis on test image using salsa20 algorithm. The histogram of original image contains large sharp rises followed by sharp declines as shown in Fig. 2(b). And the histograms of the encrypted images as shown in Figs. 2(d), 2(f) and 2(h) have uniform distribution which are significantly different from original image and have no statistical similarity in appearance. Therefore, the surveyed algorithm does not provide any clue for statistical attack. The encrypted image histogram, approximated by a uniform distribution, is quite different from plain-image histogram. Relatively uniform

$$\chi^2 = \sum_{k=1}^{256} \frac{(v_k - 256)^2}{256}, \qquad (5)$$

distribution in cipher-image histogram points out good quality of method. Uniformity caused by salsa20/r hash function is justified by the chi-square test [10] as follows: where k is the number of gray levels (256), v_k is the observed occurrence frequencies of each gray level (0–255), and the expected occurrence frequency of each gray level is 256. With a significance level of 0.01, it is found that for three variants of salsa20 $\chi^2_{test} < \chi^2(255, 0.01)$, implying that the null hypothesis is not rejected and the distribution of the encrypted histogram is uniform.

3.4 Information Entropy

Information theory is the mathematical theory of data communication and storage founded in 1949 by Shannon [11]. Information entropy is defined to express the degree of uncertainties in the system. It is well known that the entropy $H(m)$ of a message source m can be calculated as:

$$H(m) = \sum_{i=0}^{2N-1} P(m_i) \log_2 \frac{1}{P(m_i)}, \qquad (6)$$

where $P(m_i)$ represents the probability of symbol m_i and the entropy is expressed in bits. Let us suppose that the source emits 2^8 symbols with equal probability, i.e., $m = \{m_1, m_2, ..., m_{2^8}\}$. Truly random source entropy is equal to 8. 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-images in Fig. 2, the number of occurrence of each gray level is recorded and the probability of occurrence is computed. The entropy is listed in table 1. The value obtained is very close to the theoretical value of 8. This means that information leakage in the encryption process is negligible and the encryption system is secure upon the entropy attack.

Table 1: Entropy value of the cipher-images.

		Entropy value
Salsa20/r	**r = 8**	7.9969
	r = 12	7.9970
	r = 20	7.9971

3.5 Measurement of Encryption Quality

Plain-image pixels values change after image encryption as compared to their original values before encryption. Such change may be irregular. This means that the higher the change in pixels values, the more effective will be the image encryption and hence the encryption quality. So the encryption quality may be expressed in terms of the total changes in pixels values between the plain-image and the cipher-image. A measure for encryption quality may be expressed as the deviation between the original and encrypted image [12, 13]. The quality of image encryption may be determined as follows:

Let P and C denote the original image (plain-image) and the encrypted image (cipher-image) respectively, each of size $H \times W$ pixels with L grey levels. $P(x,y), C(x,y) \in \{0,...,L-1\}$ are the grey levels of the images P and C at position (x, y), $0 < x < H-1$, $0 < y < W-1$. We will define $H_L(P)$ as the number of occurrence for each grey level L in the original image (plain-image), and $H_L(C)$ as the number of occurrence for each grey level L in the encrypted image (cipher-image). The encryption quality represents the average number of changes to each grey level L and it can be expressed mathematically as:

We computed the encryption quality of three variants of Salsa20 and depicted the results in Fig. 3. Fig. 3 shows

$$EQ = \frac{\sum_{L=0}^{255} |H_L(C) - H_L(P)|}{256}. \quad (7)$$

that Salsa20 achieves a better encryption quality in the 8th ciphering round compared to the other variants.

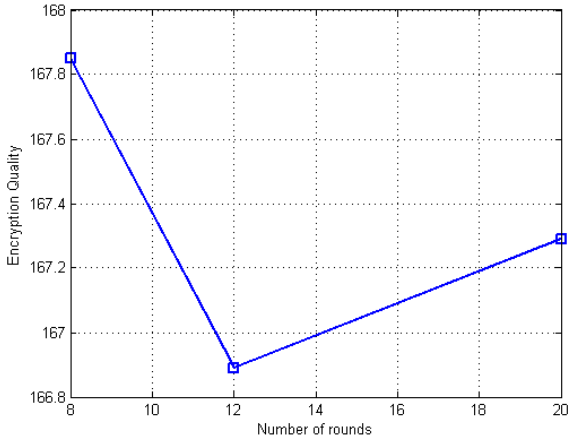

Fig. 3. Encryption quality of three variants of Salsa20. The test image is Lena.

3.6 Correlation Analysis

There is a very good correlation among adjacent pixels in the digital image [14]. Following Equations are used to study the correlation between two adjacent pixels in horizontal, vertical and diagonal orientations:

x and y are intensity values of two neighboring pixels in the image and N is the number of adjacent pixels selected

$$r_{xy} = \frac{Cov(x,y)}{\sqrt{D(x)}\sqrt{D(y)}}, \quad (8)$$

$$D(x) = \frac{1}{N}\sum_{j=1}^{N}(x_j - \frac{1}{N}\sum_{j=1}^{N}x_j)^2, \quad (9)$$

$$Cov(x,y) = \frac{1}{N}\sum_{j=1}^{N}(x_j - \frac{1}{N}\sum_{j=1}^{N}x_j)(y_j - \frac{1}{N}\sum_{j=1}^{N}y_j), \quad (10)$$

from the image to calculate the correlation. 1000 pairs of two adjacent pixels are selected randomly from image to test correlation. Correlation test image is depicted in Fig. 2(a). Fig. 4 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 image. Results for correlation coefficients of three variants of Salsa20 are shown in table 2. It is not easy to compare the results by simply observing them in the table. So, for a better comparison, we computed the average of vertical, horizontal and diagonal correlation coefficients in each ciphering round and depicted the results in figure 5. With

respect to correlation analysis versus ciphering rounds, the experimental results in Figs. 4 and 5 and table 2 show that with the increment of ciphering round, the amount of correlation between adjacent pixels changes. Also, results show that Salsa20/12 dissipates the correlation between pixels better than the other two variants of Salsa20.

Table 2: Correlation coefficients of two adjacent pixels in plain-image and cipher-image using 3 variants of salsa20.

Correlation Coefficient Analysis			
Image	**Adjacent Pixels Orientation**		
	Vertical	**Horizontal**	**Diagonal**
Plain-image	0.9986	1.0000	0.9988
Salsa20/8 Cipher-image	0.0383	0.0430	0.0117
Salsa20/12 Cipher-image	0.0021	0.0348	0.0195
Salsa20/20 Cipher-image	0.0030	0.0204	0.0653

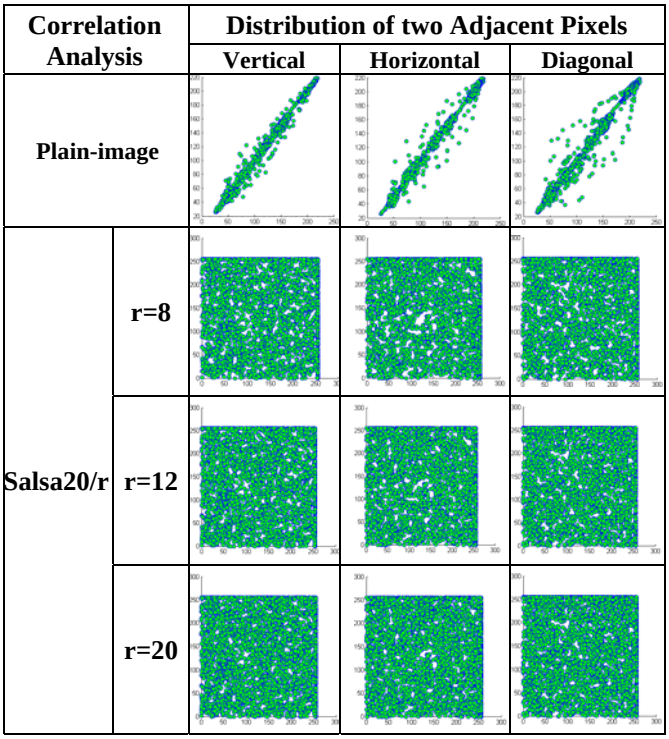

Fig. 4. Correlation analysis and distribution of two adjacent pixels in the cipher-image.

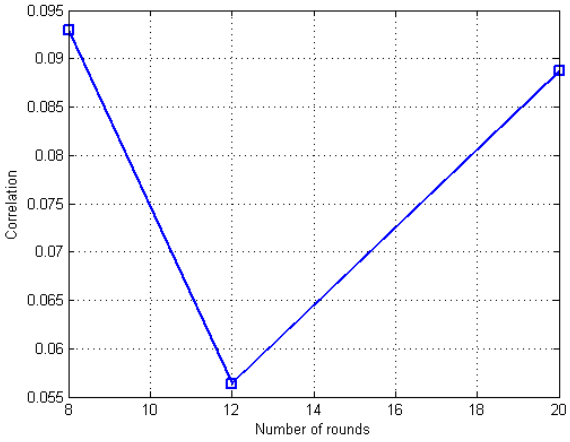

Fig. 5. Average of vertical, horizontal and diagonal correlation coefficients in each ciphering round.

3.7 Differential Analysis

In general, a desirable property for an encrypted image is being sensitive to the small changes in plain-image (e.g., modifying only one pixel). Opponent can create a small change in the input image to observe changes in the result. By this method, the meaningful relationship between original image and encrypted image can be found. If one small change in the plain-image can cause a significant change in the cipher-image, with respect to diffusion and confusion, then the differential attack actually loses its efficiency and becomes practically useless. Three common measures were used for differential analysis: MAE, NPCR and UACI [15, 16]. MAE is mean absolute error. NPCR means the number of pixels change rate of ciphered image while one pixel of plain-image is changed. UACI which is the unified average changing intensity, measures the average intensity of the differences between the plain-image and ciphered image.

Let $C(i, j)$ and $P(i, j)$ be the gray level of the pixels at the ith row and jth column of an $H \times W$ cipher and plain-image, respectively. The MAE between these two images is defined in

$$MAE = \frac{1}{W \times H} \sum_{j=1}^{W} \sum_{i=1}^{H} |C(i,j) - P(i,j)|. \qquad (11)$$

Consider two cipher-images, C_1 and C_2, whose corresponding plain-images have only one pixel difference. The NPCR of these two images is defined in

$$NPCR = \frac{\sum_{i,j} D(i,j)}{W \times H} \times 100\%, \qquad (12)$$

where $D(i,j)$ is defined as:

$$D(i,j)=\begin{cases}0, & if \quad C_1(i,j)=C_2(i,j),\\ 1, & if \quad C_1(i,j)\neq C_2(i,j).\end{cases} \quad (13)$$

Another measure, UACI, is defined by the following formula:

$$UACI=\frac{1}{W\times H}\times\sum_{i,j}[\frac{|C_1(i,j)-C_2(i,j)|}{255}]\times 100\% \quad (14)$$

Tests have been performed on the Salsa20 encryption scheme on a 256-level gray scale image of size 256×256 shown in Fig. 2(a). The MAE experiment result is shown in table 3. It is illustrated that there is a slight fluctuation between MAE of 3 variants of Salsa20. There is a slight decrease in MAE as the number of rounds rises up. The larger the MAE value, the better the encryption security. The NPCR and UACI test results are shown in table 4. Results obtained from NPCR show that the encryption scheme's sensitivity to small changes in the input image is under 0.01%. The UACI estimation result shows that the rate influence due to one pixel change is very low. The results demonstrate that a swiftly change in the original image will result in a negligible change in the ciphered image. Unfortunately, Salsa20 encryption scheme fails to satisfy plaintext Sensitivity requirement. The reason lies in Salsa20 mode of operation. Salsa20 hash function operates in counter mode and does not mangle plaintext in a complicated way like other ciphers.

Table 3: A comparison between MAE of 3 variants of Salsa20.

Method	MAE
Salsa20/8	73.2289
Salsa20/12	73.0922
Salsa20/20	72.6794

Table 4: NPCR and UACI comparison of Salsa20.

NPCR	UACI
0.0015%	0.0006%

3.8 Sensitivity Analysis

An ideal image encryption procedure should be sensitive with the secret key. It means that the change of a single bit in the secret key should produce a completely different cipher-image. For testing the key sensitivity of Salsa20 encryption scheme, the standard test image Lena (256×256) is encrypted using the secret key "A = 0123456789abcdef0123456789abcdef" (in hexadecimal) and a slightly modified secret key i.e. "B = 1123456789abcdef0123456789abcdef". Fig. 6 shows key sensitivity test result. It is not easy to compare the encrypted images by simply observing these images. So for comparison, the cipher-images histograms are depicted in Fig. 6. It can be observed that two encrypted images with a slightly different key are quite different. To ease the comparison, the percentage of different pixels between the cipher-images under these two different key is listed in table 5. Therefore, the image encryption scheme under study is highly key sensitive.

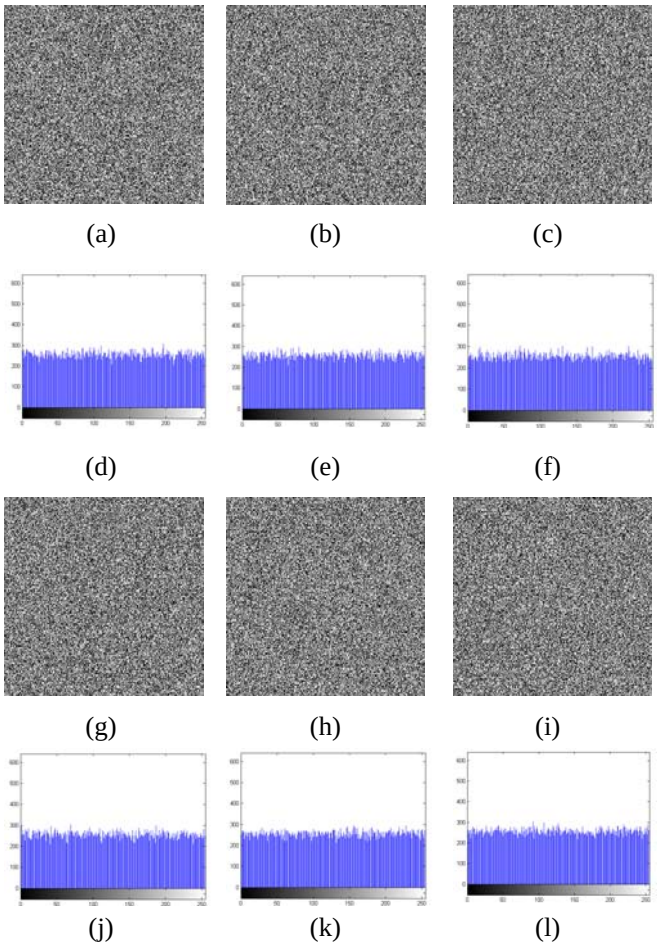

Fig. 6. Key sensitivity test: cipher-images of Salsa 20/8, Salsa20/12 and Salsa 20/20 using key A are depicted in (a), (b) and (c), respectively. The histograms of Salsa20/8, Salsa20/12 and Salsa20/20 with key A are depicted in (d), (e) and (f), respectively. cipher-images of Salsa 20/8, Salsa 20/12 and Salsa20/20 using key B are depicted in (g), (h) and (i), respectively. The histograms of Salsa20/8, Salsa20/12 and Salsa20/20 with key B are depicted in (j),(k) and (l), respectively.

Table 5: The percentage of different pixels between the cipher-images under Key A and B.

Percentage of Different Pixels		
Salsa20/8	**Salsa20/12**	**Salsa20/20**
99.59%	99.56%	99.59%

3.9 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, in order to show the effectiveness of the proposed image encryption scheme over existing algorithms, we have undertaken an analysis for the explicit comparison between the encryption speeds of three variants of Salsa20. We evaluated the performance of encryption schemes with an un-optimized MATLAB code. Performance was measured on a machine with Intel core 2 Duo 2.00 GHz CPU with 2 Gbytes of RAM running on Windows XP. The average time used for encryption/decryption on 256 gray-scale images of size 256×256 for Salsa20/8, Salsa20/12 and Salsa20/20 is respectively about 1.3, 1.7 and 2.6 s (decryption and encryption speed are the same).

4. Conclusion

In this paper, a successfully efficient implementation of Salsa20 scheme is introduced for digital image encryption. The encryption system has different variants according to number of ciphering rounds. Salsa20 has a large key space that is resistant to all kinds of brute-force attacks. Theoretical and experimental Research results showed that the scheme has resistance to statistical attacks. The uniformity was justified by the chi-square test. It is shown that Salsa20 hash function generates uniform cipher-images. Information entropy test results indicate that the cipher-image histogram distribution of the encryption scheme is so even that the entropy measured is almost equal to the ideal value. So, the surveyed encryption system is secure upon the entropy attack. The measured encryption quality showed that Salsa20/8 has a better encryption quality than the other two variants. Correlation analysis showed that correlation coefficients between adjacent pixels in the plain-image are significantly decreased after applying encryption function. Comparison between correlation coefficients of different cipher rounds showed that the least correlation occurs at the 12th round of cipher. To quantify the difference between encrypted image and corresponding plain-image, three measures were used: MAE, NPCR and UACI. The MAE experiment result showed that Salsa20/8 has the biggest MAE value among variants of Salsa20. Moreover, the MAE value decreases as the number of rounds increases. Unfortunately, differential analysis showed that Salsa20 encryption scheme fails to satisfy plaintext Sensitivity requirement. The results demonstrate that a swiftly change in the original image will result in a negligible change in the ciphered image. According to sensitivity analysis, the proposed encryption scheme is highly sensitive to the key; a small change of the key will generate a complete different decryption result and cannot get the correct plain-image. According to Performance analysis, Salsa20/8 is faster than the other two variants. All parts of the encryption system were simulated using MATLAB code. According to latter discussions, it seems that Salsa20 can be a good candidate for image encryption.

Acknowledgments

This research was supported by the Iran Telecommunication Research Center (ITRC) under Grant no. 18885/500.

References

[1] A. Akhshani, S. Behnia, A. Akhavan, H. Abu Hassan, and Z. Hassan, “A Novel Scheme for Image Encryption Based on 2D Piecewise Chaotic Maps,” Optics Communications 283, pp. 3259–3266, 2010.
[2] A. Jolfaei and A. Mirghadri, “An Applied Imagery Encryption Algorithm Based on Shuffling and Baker's Map,” Proceedings of the 2010 International Conference on Artificial Intelligence and Pattern Recognition (AIPR-10), Florida, USA, pp. 279–285, 2010.
[3] A. Jolfaei and A. Mirghadri, “A Novel Image Encryption Scheme Using Pixel Shuffler and A5/1,” Proceedings of The 2010 International Conference on Artificial Intelligence and Computational Intelligence (AICI10), Sanya, China, 2010.
[4] A. Jolfaei and A. Mirghadri, “An Image Encryption Approach Using Chaos and Stream Cipher,” Journal of Theoretical and Applied Information Technology, vol. 19, no. 2, 2010.
[5] M. Sharma and M. K. Kowar, “Image Encryption Techniques Using Chaotic Schemes: a Review,” International Journal of Engineering Science and Technology, vol. 2, no. 6, pp. 2359–2363, 2010.
[6] R. A. Rueppel, “Stream Ciphers,” Contemporary Cryptology: the Science of Information Integrity, vol. 2, pp. 65–134, 1992.
[7] D. J. Bernstein, “the Salsa20 Stream Cipher,” Proceedings of Symmetric Key Encryption Workshop (SKEW 2005), Workshop Record, 2005.
[8] D. J. Bernstein, “Salsa20 Design,” 2005, http://cr.yp.to/ snuffle/design.pdf.
[9] D. J. Bernstein, “Salsa20 Specification,” 2005, http:// cr.yp.to/ snuffle/design.pdf.
[10] 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.

[11] C. E. Shannon, Communication theory of secrecy systems. Bell Syst Tech J; 28, pp. 656–715, 1949.
[12] H. H. Ahmed, H. M. Kalash, and O. S. Farag Allah, "Encryption Quality Analysis of RC5 Block Cipher Algorithm for Digital Images," Journal of Optical Engineering, vol. 45, 2006.
[13] A. Jolfaei and A. Mirghadri, "A New Approach to Measure Quality of Image Encryption," International Journal of Computer and Network Security, vol. 2, no. 8, pp. 38–44, 2010.
[14] A. N. Pisarchik and M. Zanin, "Image Encryption with Chaotically Coupled Chaotic Maps," Physica D, 237, 20, pp. 2638–2648, 2008.
[15] G. Chen, Y. Mao, C. Chui, "A symmetric image encryption scheme based on 3d chaotic cat maps", Chaos, Solitons & Fractals, vol. 12, pp. 749–761, 2004.
[16] G. Alvarez and S. Li, "Some basic cryptographic requirements for chaos-based cryptosystems", International Journal of Bifurcation and Chaos, vol. 16, no. 8, pp. 2129–2151, 2006.

Alireza Jolfaei received the Bachelor's degree in Biomedical Engineering in the field of Bio-electric with the honor degree from Islamic Azad University, Science and Research branch, Tehran, Iran in 2007 and Master's degree in Telecommunication in the field of Cryptography with the honor degree from IHU, Tehran, Iran in 2010. He was a chosen student in the first meeting of honor students of Islamic Azad University, Science and Research Branch in 2005. Currently, he is a teacher assistant (TA) at the faculty and research center of communication and information technology, IHU, Tehran, Iran. His research interest includes: Cryptography, Information Systems Security, Network Security, Image Processing and Electrophysiology.

Abdolrasoul Mirghadri received the B.Sc., M.Sc. and PHD degrees in Mathematical Statistics, from the faculty of Science, Shiraz University in 1986, 1989 and 2001, respectively. He is an assistant professor at the faculty and research center of communication and information technology, IHU, Tehran, Iran since 1989. His research interest includes: Cryptography, Statistics and Stochastic Processes. He is a member of ISC, ISS and IMS.