Image Encryption based on Floating-Point Representation

Ali Hussein Fadel

University of Diyala- Diyala- Iraq

Received 19 April 2016 Accepted 8 June 2016

Abstract

In this paper we have presented a new design random numbers generator based on single precision floating point(RNG-SFP). Randomness of RNG-SFR is used for encryption the images. The new technique has advantage of bigger key space, smaller iteration times and high security analysis such as key space analysis. The experimented result show that the proposed technique is efficient and has high security feature

Keywords : Entropy ,Histogram, Correlation Coffiencal Horizontal , Correlation Coffiencal Vertical requency test, Serial test،,Poker test,runs test, chi-square

تشفير الصورة بالاعتماد على دقه النقطة العائمة الأحادية

علي حسين فاضل

جامعة ديالى - العراق

الخلاصة

في هذا البحث تم تقديم تصميم جديد لمولد الأرقام العشوائية بالاعتماد على دقة النقطة العائمة الأحادية. عشوائية المولد استخدمت لتشفير الصورة. هذه التقنية لديها فترة توليد فضاء مفاتيح كبيرة بأقل وقت تكرار وامنية عالية في تحليل فضاء المفاتيح. نتائج تجربة التقنية المقترحة فعالة وذات ميزات عالية الأمنية.

الكلمات المفتاحية: الانتروبي، الرسم البياني، الارتباط الرسمي أفقي، عمودي الارتباط الرسمي اختبار تردد، اختبار المسلسل ، اختبار بوكر، بتشغيل إختبار، تشي مربع

Introduction

Because the Internet has become a very big, Digital photos and videos security, it has become a necessary issue to whole Internet users. Therefore, the cryptography styles can be used to conserve the information before transmission. Transform the important information into garbage data so that no hackers can read the data called the encryption; the researchers suggested a lot of algorithms to cryptography the information such as DES, IDES and RSA.On the other hand, specific styles and specific rules need to be considered to secure the images and multimedia application. Image cryptography systems random distribution rhymester uses. Chaos is one of the most important notions that are utilized to generate a random chain because of rising suspicion of the cryptography process, which first used in the computer in 1963 by Edward Lorenz. . It was used in the chaos cryptography system due to its advantages, such as sensitivity to prime stipulations and the inability predict the sequence of chaos. Many roads in the attempt to design algorithms to encrypt the image using the chaos, such as [1] uses multiple chaotic maps to encrypt images by splitting the system in the first place in two stages. In the first stage by using a Arnold Cat map pixels are permuted and then in the second stage the permuted pixels are encrypt using multi-chaotic maps. In [2] where used one-dimensional detached Chebyshev chaotic series for column and row jostle for every pixel on the main image. [3] Used Rossler chaotic system to augmentation the suspicion in the cipher images by performing changes in the pixels value and their postures. [4] To cryptography the image and increment the size of the encrypted keys in cipher the one time pads are used together with the logistic map (as a chaotic function).In[5] to cryptography the image without using any chaotic functions; it was used a knight's tour with slips cryptography filter. However, analyzed security results, hurdles and the power of the chaotic systems [6, 7, 8]. In this sheet, we used a double precision floating point format with three different initial stipulations to establishment three different double precision floating point format series with two pixels mapping tables to increment the suspicion in the encrypted image without shuffling the original image and change the pixels value. This way reduce the implementation time of the algorithm and raise the worthiness and performance of the system.

Floating-Point Representation

A non-negative real number can be represented in decimal form with an integer fraction and denary point and it is the standard way such as in example, 33.20829, 0.000457 1128 and 70 00519.44059. We can use another standard way to represent this number by shifting the denary point and supplying appropriate powers of 10 and this method known as normalized scientific notation. So, the previous numbers have Substitute representations as

$$12.26837827 = 0.1226837827 \times 10^{2}$$

$$0.00227\,1828 = 0.22718\,28 \times 10^{-1}$$

$$30\,00527.11059 = 0.30005\,27110\,59 \times 10^{7}$$

The number is represented in normalized scientific notation by a fraction multiplied and the pioneer digit in the portion is not zero "except when the number involved is zero" so we write 79325 as 0.79325×10^{5}, not as $0.07932\,5 \times 10^{6}$ or 7.9325×10^{4} or some other way.

The word length in numerous binary computers is 32 bits (binary digits) we shall characterize contrivance of this kind whose imitative numerous work stations and personal computers in widespread use. This collection is a limited subset of the real numbers. It includes $\pm 0; \pm\infty$ the normal and sub normal single-accuracy floating-point numbers, but not the values of the number. It is noteworthy that because of the real numbers have infinite decimal or binary extensions they cannot be represented precisely as floating-point numbers for example $\pi; e; \frac{1}{3}\ ;\ 0.1\ and\ son\ on.$

The standard single-precision floating-point representation

$$(-1)^{s} \times 2^{c-127} \times (1.f)_{2}$$

for sign of mantissa we use the most significant bit for this purpose where s=1 coincide with –s, s=0 coincide with + and the number c in the exponent represented by using the next eight bits.

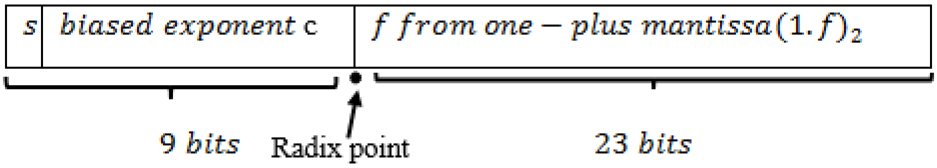


Figure 1 Partitioned floating-point single-precision computer word

of 2^{c-127}, It is interpret as a surplus-127 code. At the end, the final 23 bits from the fractional section of the mantissa in the 1-plus form represent $(1.f)_2$: Each floating-point single- accuracy word is partitioned as in Figure 1.1.

In the example below of how we can find the single-precision machine representation of the denary number 2654.42045133441 , Converting the integer portion to binary, we have$(2654.)_{10} = (5136.)_8 = (101\ 001\ 011\ 110\ .)_2$. Next, converting the fractional portion, we have $(.42045133441)_{10} = (.471205351201)_8 = (.100\ 111\ 001\ 010\ 000\ 101\ 011\ 101\ 001\ 010\ 000\ 001\ )_2$.Now

$$(2654.42045133441)_{10}$$
$$= (101001011110.100111001010000101011101001010000001)_2$$
$$= (1.01001011110100111001010000101011101001010000001)_2 \times 2^{11}$$

is the corresponding one-plus form in base 2, and $(.101\ 000\ 011\ 110)_2$ is the stored mantissa . Next the exponent is $(11)_{10}$, and since c− 127 = 11, we immediately see that $(138)_{10} = (212)_8 = (10\ 001\ 010\ )_2$ is the stored exponent. Thus, the single-precision representation of 2654.42045133441 is

$$[1\ 10\ 001\ 010\ 01001011110100111001010000101011101001010000001]_2 =$$

$$[1100\ 0101\ 0010\ 0101\ 1110\ 1001\ 1100\ 1010\ 0001\ 0101\ 1101\ 0010\ 1000\ 0001\ ]_2$$

In the table below shows the Floating-Point Representation group of random numbers

Table 1 Floating-Point Representation group of random numbers

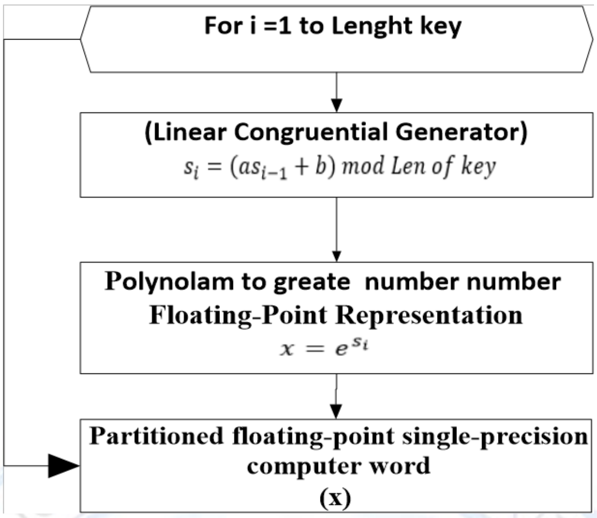
	decimal number	Floating-Point Representation
1	175.154710422 755	0100 0011 1101 0111 1010 0100 0000 0101 0111 0101 1011 0100 1110 0011
2	380.615938663 52	1100 0011 1011 1110 0111 0010 1011 1010 0011 1110 1000 0011 1000 00
3	15432.0133058 071	0100 0110 1111 1000 1001 0001 1111 1011 1001 0011 1000 0101 1100
4	36.2646969939 414	1100 0010 0001 0010 0110 1000 0100 1011 1011 0011 1111 0101 1101 0110
5	435.672143074 468	1100 0011 1101 1001 1001 0011 1000 1111 1101 1011 0010 1100 1001 0100 1000 00
6	9559.49964120 618	0100 0110 1100 1010 1010 1111 0111 0100 0100 0010 1111 1111 0100 0101 0100 00
7	141.801802738 374	0100 0011 1100 0110 1001 0111 0101 0101 1110 0101 0101 1011 0101 1000 1100 00
8	19597.3349556 453	1100 0110 1001 1001 0001 1010 1100 0111 1010 0110 0010 1100 1110 0101
9	32.3453586651 904	1100 0010 0000 0011 0010 0100 0001 1001 1100 1110 0011 1011 0000 0000
10	289.842975097 763	1100 0011 1001 0000 1001 1000 1000 1000 1010 0111 0110 0110 0111 0100 0110 00
. . .	. . .	. . .

Proposed system mode

The propose image encryption algorithm consists of two stage iteration (multilevel) block permeation and nonlinear key stream cipher.

1. **proposed key generation**

   In this section of the pseudo-random number generation and the structure is based on floating point linear congruential generator and representation, and such a demand generator natural source of randomness (non- deterministic)



Algorithm (1,1)	
Goal :	the generation of the key Depending on height and Wight image
Input :	Wid-Image ,Hgt-Image
Output :	Key Generation
Step 1	*Set parameter from key generation* $Len\ of\ key$ ← Wid-Image*Hgt-Image *24 Get t a,b s_i where $1 \le a, b \le len\ of\ key - 1$ and $0 \le s_0 \le len\ of\ key - 1$ $(Linear\ Congruential\ Generator)$
Step 2	Generation bit key Key bit =null For all i Do { where 0 To $Len\ of\ key$ } $s_i = (as_{i-1} + b)\ mod\ Len\ of\ key$ $x = e^{s_i}$ Stram_bit= call Function $Floating - Point\ Representation(x)$ Key bit= Key bit + Stram_bit Exit For

Figure 1 algorithms key generation

2. Encryption and Decryption process

then proposed image encryption and decryption algorithm can be summarized in the following algorithm:

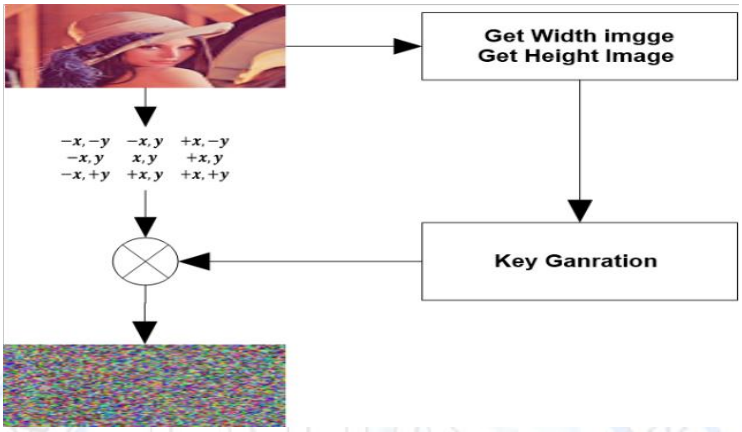


Figure 2 Encryption / Decryption image

Algorithm (1,1) Encryption / Decryption	
Goal :	the generation of the key Depending on
Input :	Wid-Image ,Hgt-Image
Output :	Key Generation

Encryption of image

```
Offset ←1
Countk ←0
For all  X , Y Do {where Offset to Wid – Offset  , Offset To Hgt- Offset }
     For all I Do { Where Offset To Offset -1}
          For all J Do { Where Offset To Offset -1}
               ReGrBl ←Convert To Bin(GetPixel(j + fi, i + fj) ))
               xbin ←""
               For all K Do { Where 1  To 24}
                    If Countk = Length Bits Key THEN
                         Countk ← 0
                    End If
                    Countk+ ← + 1
                    xbin += Key[Countk]
               End For
               xReGrBl←ReGrBl Xor Bin2Dec(xbin)
               Put ReGrBl (x,y)
          End For
     End For
Exit For
```

Figure 3 Algorithm Encryption / Decryption

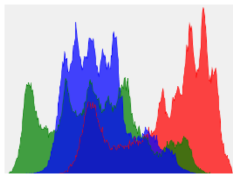
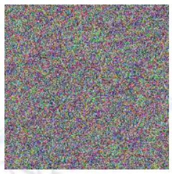
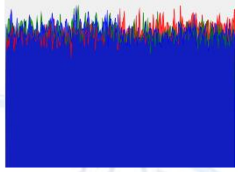
Result and Analysis

In this paragraph will be tested on Statistics generated by the algorithm mentioned above, where the key was conducted four tests which tests, 10000 key length and the results were as shown in the table below (frequency test، serial test، poker test، runs test)

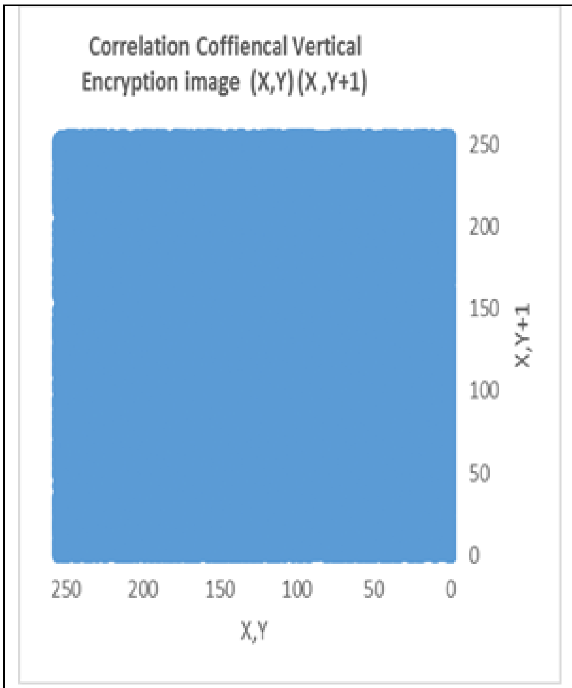
Table 2 Statistical tests

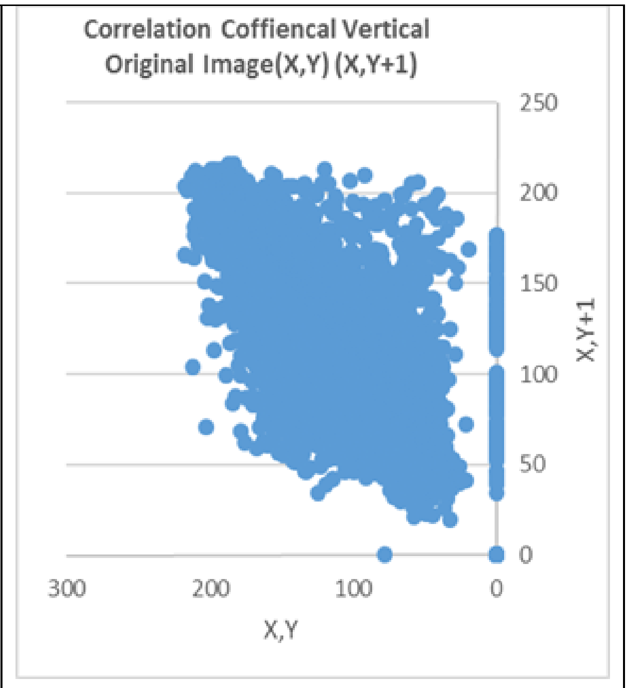
Key size	Statistical tests		X2 (chi-square) distribution			Result P(X>x)
	Name Test	Value(x)	α	Degrees of freedom	Value(X)	
500	Frequency test	3.752	0.05	1	3.8415	Pass
	Serial test	5.665		2	5.9915	Pass
	Poker test	10.552		15	26.2962	Pass
	Runs test	12.055		6	26.2962	Pass
1000	Frequency test	3.830		1	3.8415	Pass
	Serial test	5.027		2	5.9915	Pass
	Poker test	38.247		31	82.5287	Pass
	Runs test	9.318		8	82.5287	Pass

Where the key that is configured of the proposed algorithm has been applied in the encrypted color image has been holding a series of measurements or tests (Entropy, Histogram, Correlation Coffiencal Horizontal, Correlation Coffiencal Vertical) As shown in the chart below

Histogram		
Original image		
Encryption image		

	Test	Original image	Encryption image
1	Correlation Coffiencal Vertical (X,Y) (X ,Y+1)	0.985662091165786	0.61052688914019
2	Correlation Coffiencal Vertical (X,Y) (X+1,Y)	0.989217276188664	0.595872112765076
3	Entropy	7.27129320337589	7.99651685627914





Conclusions

In this research was to provide a new random number generator depends on (Floating-Point Representation) Where it was generating an initial value through a function numbers $x = e^x$ (Floating-Point). The result was characterized by sequential access to statistical characteristics of a good where succeeded in statistical tests as in the Table FIGURE 7.After that has been adopted on a row in the encrypted image of colorful Bmp type where the results were as shown in Table FIGURE 9, It was chosen as the resulting image in the encryption key based on the proposed test methods (Correlation Coffiencal Vertical , Correlation Coffiencal Vertical and Entropy) The results were so good that hold up against the statistical analysis and differential analysis.

References

1. Zhu, Weihua, and Ying Shen. "Encryption algorithms using chaos and cat methodology." Anti-Counterfeiting Security and Identification in Communication (ASID), pp.20-23, 2010.
2. Dinghui, Zhang, et al. "Discrete chaotic encryption and decryption of digital images." International Conference on Computer Science and Software Engineering.pp. 849-852, 2008.
3. Cao, Ying-yu, and Chong Fu. "An image encryption scheme based on high dimension chaos system." International Conference on Intelligent Computation Technology and Automation (ICICTA).pp. 104-108, 2008.
4. Jeyamala, C., S. GopiGanesh, and G. S. Raman. "An image encryption scheme based on one time pads—a chaotic approach." International Conference on Computing Communication and Networking Technologies (ICCCNT).pp.1-6, 2010.
5. Delei, Jiang, Bai Sen, and Dong Wenming. "An image encryption algorithm based on knight's tour and slip encryption-filter." International Conference on Computer Science and Software Engineering.pp.251-255, 2008.

6. Xiao, Di, Xiaofeng Liao, and Pengcheng Wei. "Analysis and improvement of a chaos-based image encryption algorithm." Chaos, Solitons & Fractals 40.5.pp.2191-2199, 2009.

7. Rhouma, Rhouma, and Safya Belghith. "Cryptanalysis of a new image encryption algorithm based on hyper-chaos." Physics Letters A 372.38.pp.5973-5978, 2008.

8. Kincaid, David Ronald, and Elliott Ward Cheney. Numerical analysis: mathematics of scientific computing. Vol. 2. American Mathematical Soc., 2002.