Computers and Mathematics with Applications 60 (2010) 326–337

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

Chaos-based secure satellite imagery cryptosystem

Muhammad Usama [a], Muhammad Khurram Khan [a,*], Khaled Alghathbar [a,c], Changhoon Lee [b]

[a] Center of Excellence in Information Assurance (COEIA), King Saud University, Saudi Arabia

[b] School of Computer Engineering, Hanshin University, South Korea

[c] Information Systems Department, College of Computer and Information Sciences, King Saud University, Saudi Arabia

ARTICLE INFO

Keywords:
Satellite image
Chaos
Cryptosystem
Security
Network

ABSTRACT

With the large-scale development in satellite and network communication technologies, there is a great demand for preserving the secure storage and transmission of satellite imagery over internet and shared network environment. This brings new challenges to protect sensitive and critical satellite images from unauthorized access and illegal usage. In this paper, we address the aforementioned issues and develop techniques to eliminate the associated problems. To achieve this, we propose a new chaos-based symmetric-key encryption technique for satellite imagery. This scheme utilizes multiple chaotic maps e.g. Logistic, Henon, Tent, Cubic, Sine and Chebyshev for enhancing the key space, robustness and security of satellite imagery. We perform key sensitivity, statistical and performance analysis experiments to determine the security, reliability, and speed of our algorithm for satellite imagery. The proposed algorithm presents several interesting features, such as a high level of security, large enough key space, pixel distributing uniformity and an acceptable encryption speed as compared to AES, 3-DES, and DES.

© 2010 Elsevier Ltd. All rights reserved.

1. Introduction

With the rapid growth of internet and proliferation of space sciences and technologies, applications of satellite images and maps have become common and is still continuously attracting the attention of commercial, academic and government communities. Satellite-based communication and remote sensing technologies have shown their capabilities in providing services related to education, healthcare, weather forecast, land and water resources management, etc., [1]. However, the satellite imagery distribution and deployment process is usually based on CD/DVD-ROM or on shared network environment (Internet, LAN, WAN etc.). In the same way as for multimedia images, the digital format of satellite imagery implies an inherent risk of an unauthorized copy or use of the product. Thus, it is important to enforce security to ensure authorized access to sensitive data.

To fulfill such security and privacy needs, image encryption algorithms are important for satellite imagery protection. There are a number of encryption algorithms available such as DES, AES, International Data Encryption Algorithm (IDEA) and RSA (developed by Rivest, Shamir and Adleman) [2–4]. These traditional encryption algorithms have shortcomings and they are not considered as ideal for image applications, mainly because of low level of efficiency when dealing with large and redundant blocks of image data. Moreover, these algorithms require more than the usual expected computation time and power while performing image encryption.

In order to develop efficient and reliable image encryption algorithm, it is important to study and understand the characteristics and special features of existing text and multimedia images encryption techniques. In the following subsection, some basic concepts in cryptography with respect to image encryption are introduced first.

* Corresponding author.
E-mail addresses: khurram.khan@scientist.com, mkhurram@ksu.edu.sa (M.K. Khan).

0898-1221/$ – see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2009.12.033

2. Chaos-based encryption schemes

To overcome previously mentioned problems and drawbacks in classical image encryption techniques like Simple-DES, Triple-DES and AES, the chaos-based encryption is suggested by many researchers to deal with multimedia data especially images [5–8]. Chaos-based encryption techniques are one of the most efficient ways for dealing with bulky, difficult, intractable problem of fast and highly secure multimedia image encryption. A chaotic system is rich in significance or implication because it has high sensitivity to its initial condition, parameter value, ergodicity (a system that tends in probability to a limiting form that is independent of the initial conditions), random behaviour and unstable periodic orbits with long periods. The properties diffusion, dispersion, disorder, and confusion required in conventional cryptography algorithms are achieved through iterative processing. The important difference between chaos-based and conventional cryptography algorithms is that encryption transformations are defined on finite sets, while chaos has meaning only on real numbers [9].

Initial work on chaos-based cryptosystems was based on chaotic dynamical system. The concepts of chaotic dynamical system were associated with synchronization of two chaotic systems and controls [10]. Several methods and techniques have been proposed in this domain to synchronize chaotic systems. Some typical forms have been brought up, which include chaotic masking, shift keying and modulation using inverse systems [11–14].

To overcome the security, privacy and reliability issues of satellite imagery, in this paper, a new chaos-based symmetric key cryptosystem has been proposed using external secret key, which has been extended by including multiple chaotic maps named as Logistic, Henon, Tent, Cubic, Sine and Chebyshev. The proposed system is a chaotic cryptosystem to secure satellite imagery over shared network environments and secure storage on CDs, DVDs and/or hard disks. The simple logical XOR and one time multiple key generation processes have been carried out for satellite image encryption and decryption. A series of experiments have been performed to evaluate the security analysis of the presented system and according to the comparative, theoretical and experimental results; we conclude that the proposed chaos-based satellite image cryptosystem is much useful for real-time satellite image encryption and decryption, in order to keep the storage and transmission process secure and reliable.

The rest of the paper is organized as follows: In Section 3, we propose our algorithm based on multiple chaotic maps. In Section 4, we perform experiments to evaluate the robustness of the presented system. At the end, Section 5 concludes the findings of this paper.

3. The proposed algorithm based on multiple chaotic maps

Since 1990s, there have been a number of symmetric-key chaos-based image and plaintext encryption algorithms proposed to achieve the high diffusion and confusion for securing sensitive data [15–24]. These algorithms are based on single chaotic map for generating secret key, used in encryption and decryption process.

This study mainly concern with the idea of using multiple chaotic maps for generating secret key. The concept of multiple chaotic maps improve the security level of the algorithm by enhancing confusion and diffusion in encryption. The proposed chaos-based image encryption algorithm is a block cipher which uses multiple chaotic maps for generating secret key of variable length e.g. 128, 256, 512 bits. The detailed description of each step of the proposed encryption and decryption process is given below:

For the encryption/decryption, we divide original and output cipher images into variable length blocks e.g. 128, 256 and 512 bits (named as block size BS in bits). Original image and Cipher image of n blocks can be represented as:

$$O = O_1O_2O_3O_4O_5 \ldots O_m \tag{1}$$

$$C = C_1C_2C_3C_4C_5 \ldots C_m \tag{2}$$

where $i = 1, 2, 3, \ldots, m$ and $m \geq 1$.

The proposed algorithm uses variable length secret key of 128, 256 and 512 bits (must be same sized as defined for input block) which is converted into bytes format before encryption and decryption operations. Secret key can be represented as:

$$K = bK_1bK_2bK_3bK_4bK_5 \ldots bK_n \tag{3}$$

where $n = BS/8$ and BS is block size in bits, so the size of the secret key K depends on size of the block. For example, if block size BS is 256 bits then size of the secret key K in bytes is 32 (that is equal to 256/8). As defined earlier that the proposed symmetric-key block cipher is designed to utilize multiple chaotic maps for secret key, which will be further helpful in increasing the strength of encryption. The proposed algorithm uses six different chaotic maps named as Logistic map, Tent map, Henon map, Sine map, Cubic map and Chebyshev map. The reason for choosing these maps is that their security and efficiency have been proven by many researchers [6,11,14]. Table 1 shows the experimented chaotic maps, governing equations, and their parameter values. The generated chaotic sequences of each chaotic map are in real number, so first we transform these real numbers into sequence of bits (0 s and 1 s) and then in bytes format for getting secret key in bytes.

The hexadecimal mode is used to define the secret key for initial condition IC (for each chaotic map) which generates different bit sequences from real numbers. The secret keys are generated by the following equations:

Table 1
Governing equations and system parameter values in chaotic range for the chaotic maps used in the proposed algorithm.

Chaotic map	Governing equation	Parameter value
Chebyshev	$x_{n-1} = \cos(\lambda \cos^{-1}(x_n))$	$\lambda = 4$
Logistic	$x_{n+1} = \lambda x_n(1 - x_n)$	$\lambda = 4$
Cubic	$x_{n+1} = \lambda x_n(1 - x_n^2)$	$\lambda = 2.59$
Sine	$x_{n+1} = \lambda \sin(\pi x_n)$	$\lambda = 0.99$
Henon	$x_n = 1 + \lambda(x_{n-2} - x_{n-3}) + ax_{n-2}^2$	$\lambda = 0.3 \quad 1.07 \leq a \leq 1.09$
Tent	$x_{n+1} = \begin{cases} x_n/\mu & \text{if } x_n \leq \mu \\ 1 - x_n/1 - \mu & \text{if } x_n \geq \mu \end{cases}$	$\mu = 0.4$

$$
\begin{array}{rrllll}
Chebyshev = & BK = & bk_1, & bk_2, & \ldots\ldots\ldots\ldots\ldots, & bk_n \\
 & & \oplus, & \oplus, & \ldots\ldots\ldots\ldots\ldots, & \oplus \\
Logistic = & LK = & lk_1, & lk_2, & \ldots\ldots\ldots\ldots\ldots, & lk_n \\
 & & \oplus, & \oplus, & \ldots\ldots\ldots\ldots\ldots, & \oplus \\
Cubic = & CK = & ck_1, & ck_2, & \ldots\ldots\ldots\ldots\ldots, & ck_n \\
 & & \oplus, & \oplus, & \ldots\ldots\ldots\ldots\ldots, & \oplus \\
Sine = & SK = & sk_1, & sk_2, & \ldots\ldots\ldots\ldots\ldots, & sk_n \\
 & & \oplus, & \oplus, & \ldots\ldots\ldots\ldots\ldots, & \oplus \\
Henon = & HK = & hk_1, & hk_2, & \ldots\ldots\ldots\ldots\ldots, & hk_n \\
 & & \oplus, & \oplus, & \ldots\ldots\ldots\ldots\ldots, & \oplus \\
Tent = & TK = & tk_1, & tk_2, & \ldots\ldots\ldots\ldots\ldots, & tk_n \\
 & & =, & =, & \ldots\ldots\ldots\ldots\ldots, & = \\
Keys = & K = & k_1, & k_2, & \ldots\ldots\ldots\ldots\ldots, & k_n
\end{array}
$$

Fig. 1. Equation form of key generation process.

$$N = \sum_{i=1}^{BS/8} (bK_i/256) \tag{4}$$

$$IC = N - \lfloor N \rfloor \tag{5}$$

where bK_i is the ith key value in decimal equivalent of the secret key, $\lfloor N \rfloor$ is the floor of the value N, BS is the block size of the variable length e.g. 128, 256 and 512 bits and IC is the initial condition value, which is transformed back into real numbers.

If we use same secret key and block size for every chaotic map in the proposed algorithm then it generates same initial condition for all chaotic maps as shown in Table 1. The proposed algorithm uses six different combinations of chaotic maps for key generation as shown in Eq. (6), Figs. 1 and 2, where each map may have n numbers of keys and n must be greater than or equal to one.

Suppose the number of different distinct keys is n which is $n = 10$ then proposed algorithm creates ten keys from each map using single input secret key and parameters values (as given in Table 1). To combine these keys, we perform XOR operation as:

$$K_i = BK_i \oplus LK_i \oplus CK_i \oplus SK_i \oplus HK_i \oplus TK_i \tag{6}$$

where $i = 1, 2, 3, \ldots, n$ and $n \geq 1$.

For the encryption/decryption, we divide original/cipher image into m number of blocks of variable length (e.g. 128, 256 and 512 bits) as defined in Eqs. (1) and (2). For encryption, each block b_i of original image is XOR-ed with key k_i for producing cipher block c_i where $i = 1, 2, 3, \ldots, m$ as shown in Fig. 3 (in equation form).

The encryption/decryption process continues till the original/cipher image is completely encrypted/decrypted. The block diagram of encryption/decryption process is given in Fig. 4. The proposed algorithm is secret key dependent, so the encryption time will always depend on time taken to generate n number of keys during encryption or decryption. In this study, we have calculated the encryption time using ten keys (parameters defined in Table 2, for Boston satellite image).

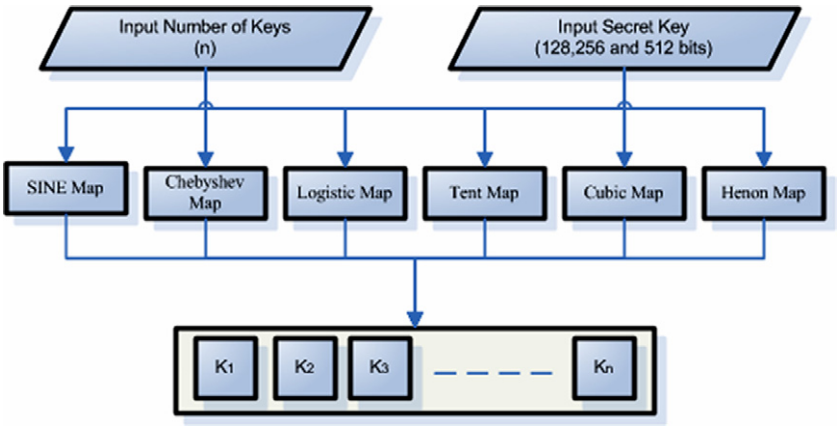


Fig. 2. Block diagram of key generation process.

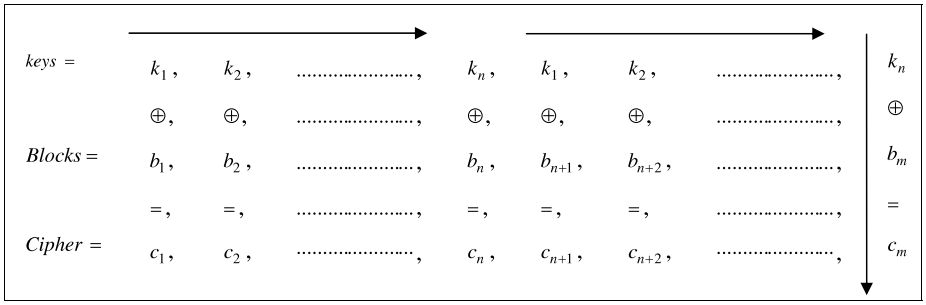


Fig. 3. Equation form of encryption/decryption process.

We observe from the experimental results that the encrypted image produced by the proposed algorithm has same size, resolution and geo-information as found in original Boston satellite image. Therefore, both encrypted and original images are identical in term of size, image resolution and geographical information (which is found in almost every satellite image). The performance level of the algorithm may vary with the size of the key. The supported key sizes are 128, 192, 256 and 512 bits. The length of the secret key does not increase the processing time of the algorithm drastically however, a minor change in processing time is negligible.

The total number of operations required for encryption will remain same (if we are using same number of key as in this case $n = 10$). Therefore, increase in the length of the secret key does affect the overall processing and computation time while performing encryption and decryption operations on the particular satellite image. However, time required to compute secret keys (once in the algorithm) for encryption may vary. This computation is required only once during encryption/decryption process. So this minor change in time can be negligible and does not have any great impact on algorithm's overall performance.

4. Measurements, simulation and evaluation experiments

We have implemented the proposed algorithm using Microsoft C#. Net programming language and MATLAB, and observed the results on a Pentium-IV 1.8 GHz PC with 1.46 GB RAM. The results of some experiments are given to prove efficiency and security of the proposed cryptosystem for satellite images. We use gray-scale Boston satellite image of size 1000×1000 as the original image in Fig. 5. The secret key "123456GHIJKLMNOPQRSTUVWXYZ[\]^_" (in ASCII) is used for encryption and decryption whose size is 256 bits (32 Bytes).

Tables 1 and 2 show initial parameters and governing equations of chaotic maps used in the presented algorithm. Some of the encrypted images using proposed application are shown in Figs. 6(a), 7(a) and 8(a). The encrypted images are totally scrambled from original images. The decrypted images are shown in Figs. 6(b), 7(b) and 8(b).

The visual inspection of Figs. 6–8 shows the possibility of applying the proposed algorithm successfully and it reveals the algorithm's effectiveness in hiding the information contained in them. Although it's not enough to completely rely on visual inspection because there is good possibility of human error. Therefore, quantitative measurement techniques are needed to be considered for inspection and evaluation of satellite image encryption quality.

To evaluate and compare the experimental results of the proposed algorithm, some security analysis techniques are considered as quality measurement factors such as Maximum deviation, Information entropy, Histograms analysis and Key

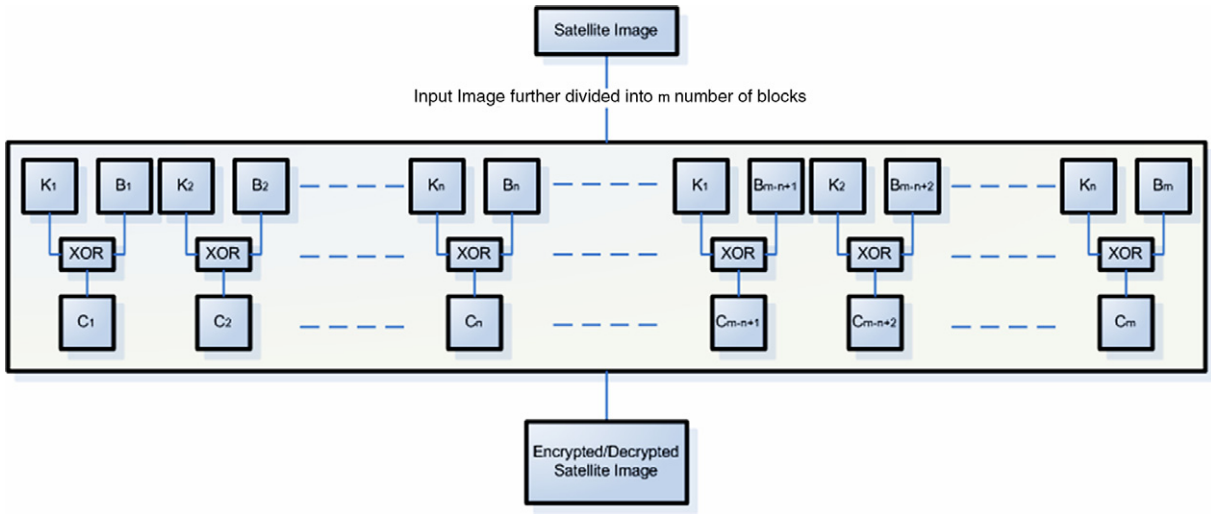


Fig. 4. Block diagram of encryption/decryption process.

Table 2
Initial parameters used in experiments.

Secret key	123456GHIJKLMNOPQRSTUVWXYZ[\]^_‘		
Hex secret key	63676B8F93979B9FA3A7ABAFB3B7BBBF		
Block size	256 bits (32 Bytes)		
Map	Lambda	Counter	Initial condition
Cubic	2.5900005	0.0000001	0.734375
Henon	0.3000005	0.0000001	0.734375
Logistic	4.0000005	0.0000001	0.734375
SINE	0.9000005	0.0000001	0.734375
Tent	0.4000005	0.0000001	0.734375
Chebyshev	2.6000005	0.0000001	0.734375

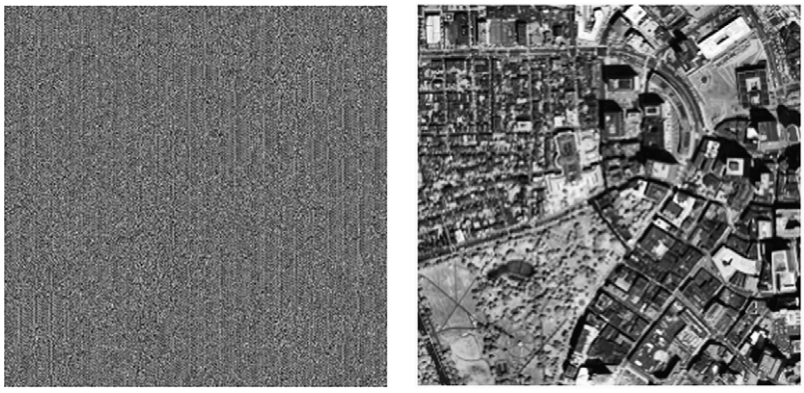
Fig. 5. The Boston original image.

space analysis. In the following subsections, we perform these experiments to evaluate the robustness and security of our proposed algorithm:

4.1. Maximum deviation

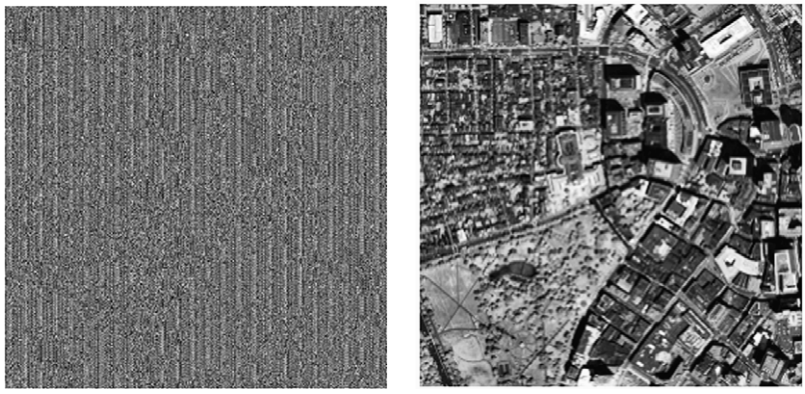
Maximum deviation measures the inaccuracy of encryption or decryption process in terms of how algorithm maximizes the deviation between the resultant encrypted/decrypted and original images [25], positive or negative value to indicate an acceptance or satisfaction level. The steps of this measure are as follows [26]:

Firstly, generate the histogram chart that shows the distribution for both encrypted/decrypted and original gray-scale images and then count the number of pixels of each gray-scale value between the range of 0 to 255.



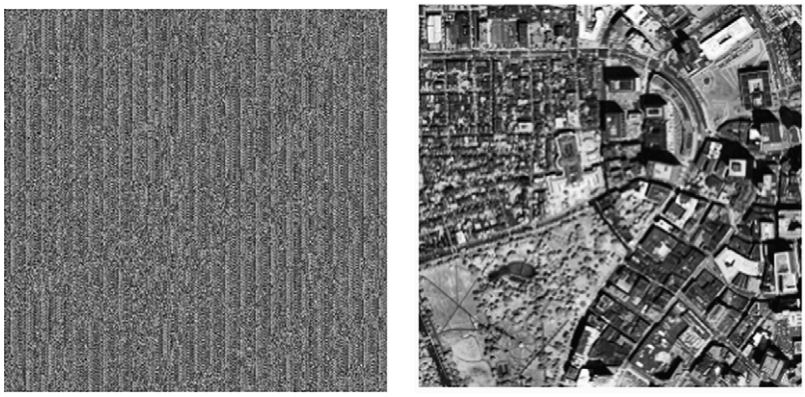
(a) Encrypted image. (b) Decrypted image.

Fig. 6. Application of the proposed algorithm to the Boston image using Chebyshev, Cubic, Henon, Logistic, Tent, and Sin chaotic maps.



(a) Encrypted image. (b) Decrypted image.

Fig. 7. Application of the proposed algorithm to the Boston image using Chebyshev, Logistic, and Tent chaotic maps.



(a) Encrypted image. (b) Decrypted image.

Fig. 8. Application of the proposed algorithm to the Boston image using Logistic, Tent, and Sin chaotic maps.

Secondly, calculate the difference between these two computed values. At last, measure the area under the curve by simply adding these values and it is the sum of deviation D.

$$D = \frac{h_0 + h_{255}}{2} + \sum_{i=1}^{254} h_i \tag{7}$$

here h_i is the maximum displacement between two curves at value i. The higher (positive or negative) the value of deviation D, the more the encrypted image is deviated from the original image.

Table 3
Maximum deviation of the proposed algorithm with different combinations of chaotic maps.

Different combinations of chaotic maps	Maximum deviation
Chebyshev, Cubic, Henon, Logistic	21 814 139
Chebyshev, Cubic, Henon, Logistic, SINE	21 970 372
Chebyshev, Cubic, Henon, Logistic, Tent	21 759 044
Chebyshev, Cubic, Henon, Logistic, Tent, SINE	22 376 774
Henon, Logistic	21 519 040
Henon, Logistic, SINE	21 814 569
Henon, Logistic, Tent	22 106 332
Logistic, SINE	21 252 492
Logistic, Tent	21 608 475
Logistic, Tent, SINE	22 260 921

Table 4
Entropy of the proposed algorithm with different combinations of chaotic maps.

Different combinations of chaotic maps	Entropy
Chebyshev, Cubic, Henon, Logistic	7.9994
Chebyshev, Cubic, Henon, Logistic, Sine	7.9994
Chebyshev, Cubic, Henon, Logistic, Tent	7.9991
Chebyshev, Cubic, Henon, Logistic, Tent, Sine	7.9992
Henon, Logistic	7.9992
Henon, Logistic, Sine	7.9994
Henon, Logistic, Tent	7.9991
Logistic, Sine	7.9963
Logistic, Tent	7.9957
Logistic, Tent, Sine	7.9993

The calculated maximum deviation results of the gray-scale Boston encrypted/decrypted images from the original image using the proposed algorithm are given in Table 3. The experiment results suggest that proposed algorithm gives a greater maximum deviation results which are desirable for an efficient and secure cryptosystem.

4.2. Information entropy

Information theory is the mathematical theory of data communication and storage, which was introduced by Claude E. Shannon in his classic paper [27]. The Shannon entropy or information entropy is a measure of the uncertainty associated with a random variable. It quantifies the information contained in data, usually in bits or bits/symbol. It is the minimum message length necessary to communicate information. For example, a long string of repeating characters has entropy of 0, since every character is predictable. The entropy of English text is between 1.0 and 1.5 bits per letter, [28] or as low as 0.6 to 1.3 bits per letter, according to estimates by Shannon based on human experiments [29].

To calculate the entropy $H(m)$ of a source m, we have:

$$H(m) = \sum_{i=0}^{2N-1} P(m_i) \log_2 \frac{1}{P(m_i)} \tag{8}$$

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. $m = \{m_1, m_2, \ldots, m_{2^8}\}$. After evaluating the above equation, we obtain its entropy $H(m) = 8$, corresponding to a truly random source.

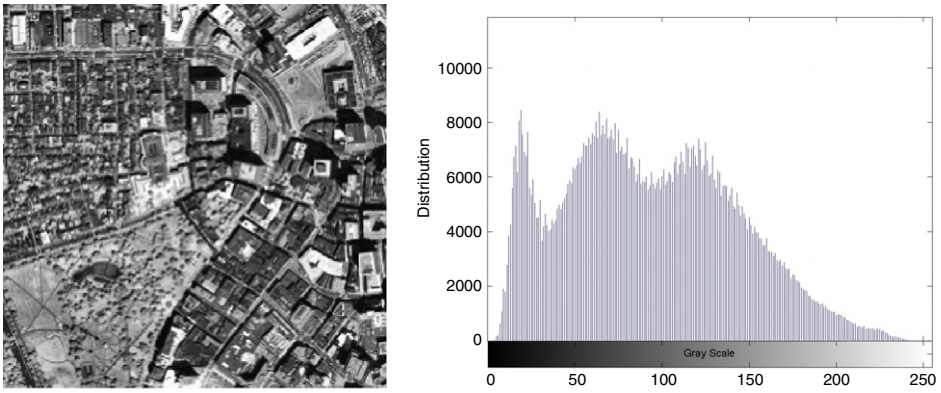
Let us consider the cipher image using the proposed algorithm, the number of occurrence of each cipher image block is recorded and the probability of occurrence is computed. The entropy is calculated by the following equation:

$$H(m) = \sum_{i=0}^{255} P(m_i) \log_2 \frac{1}{P(m_i)}. \tag{9}$$

The calculated information entropy values with different combinations of chaotic maps are given in Table 4. The different combinations of chaotic maps give greater results of information entropy. The values obtained by the experiments are very close to the theoretical value of 8. Here, it is pertinent to say that the information leakage in the image enciphering process is negligible and the proposed cryptosystem is robust upon the entropy attack.

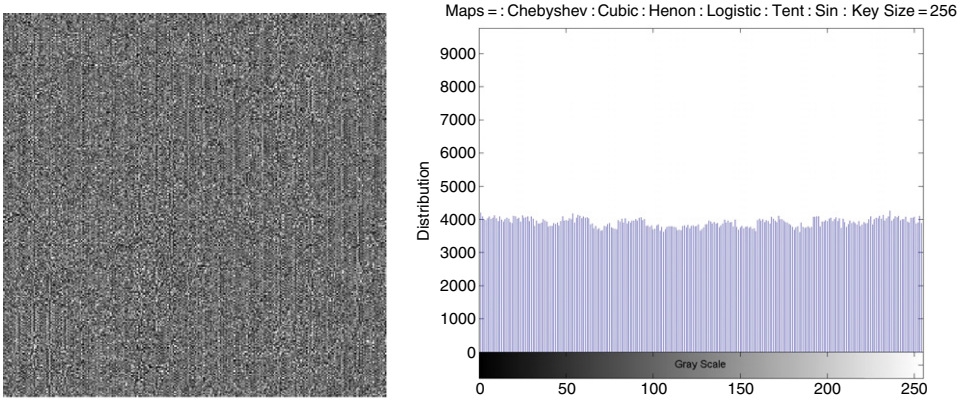
4.3. Histogram analysis

To prevent the leakage of information to attackers, it is important to ensure that the encrypted and original images do not have any statistical similarities. The histogram of image clarifies that how pixel elements in an image are distributed



(a) Original image. (b) Histogram of the original image.

Fig. 9. The Boston satellite image and its corresponding histogram.



(a) Encrypted image. (b) Histogram of the image.

Fig. 10. Histogram analysis of the Boston image using Chebyshev, Cubic, Henon, Logistic, Tent, and Sin chaotic maps.

using graphical display of the pixel elements, by measuring color intensity level of each pixel element. To examine the statistical distribution, we perform analysis on several test results of the proposed algorithm by computing and analyzing the histograms of these images.

Mathematically, a histogram is a mapping m_i that calculates the number of observation that has no common categories. Let N be the total number of facts learned by observation and n be the total number of categories, the histogram can be computed by the following equation:

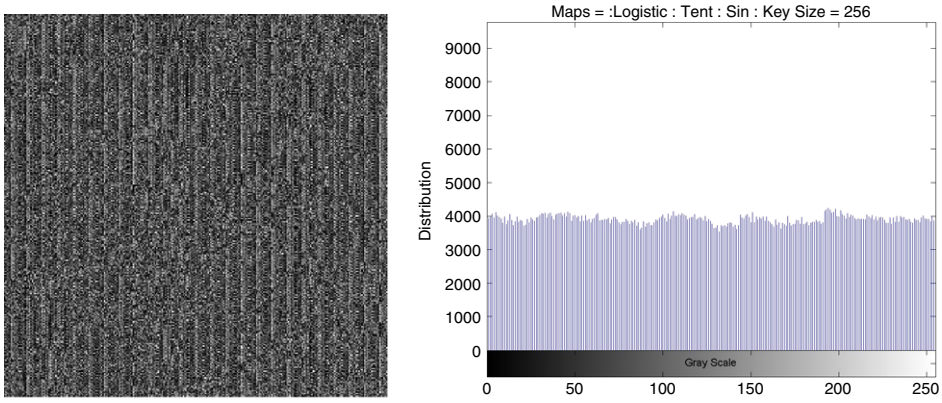
$$N = \sum_{i=0}^{n} m_i. \tag{10}$$

The histogram of a Boston satellite image contains large sharp rises followed by sharp declines as shown in Fig. 9. These sharp rises and declines correspond to color values that appear more often in the Boston satellite image.

The histogram of the encrypted images using proposed algorithm are shown in Figs. 10 and 11, these histograms have uniform distribution which are significantly different from Boston image histogram with no statistical similarity in appearance. Therefore, the proposed chaos-based satellite image encryption algorithm does not provide any clue for statistical attack.

4.4. Key space analysis

For the secure image encryption and decryption algorithm, the key space should be large enough to make the brute force and other similar attacks infeasible and unworkable. The proposed algorithm has 2^{128}, 2^{192}, 2^{256} and 2^{512} different combinations of the secret keys. An image encryption with such a long key space is sufficient for reliable practical use. Furthermore, in the proposed algorithm, multiple chaotic maps are employed and they all are very sensitive to their initial conditions and parameter values.



(a) Encrypted image. (b) Histogram of the image.

Fig. 11. Histogram analysis of the Boston image using Logistic, Tent, and Sin chaotic maps.

4.5. Key sensitivity analysis

An ideal image encryption algorithm should be sensitive with respect to both the input secret key and original image. The change of a single bit in either the secret key or original image should produce completely different output results. To prove the robustness of the proposed algorithm, we perform sensitivity analysis with respect to key and image. High key sensitivity is required by secure satellite image cryptosystem, which means that the encrypted image cannot be decrypted correctly although there is only a slight difference between secret key. This guarantees the security of the proposed algorithm against brute-force attacks to some extent. For testing the key sensitivity of the proposed algorithm, we have performed the following steps:

An original image in Fig. 12(a) is encrypted by using the secret key "12345678901234567890123456789012" (in ASCII) and the resultant image is referred as encrypted image A as shown in Fig. 12(b).

The same original image is encrypted by making the slight modification in the secret key i.e. "223456789012345678 90123456789012" (in ASCII) (the most significant bit is changed in the secret key) and the resultant image is referred as encrypted image B as shown in Fig. 12(c).

Again, the same original image is encrypted by making the slight modification in the secret key i.e. secret key "12345678901234567890123456789033" (in ASCII) (the least significant bit is changed in the secret key) and the resultant image is referred as encrypted image C as shown in Fig. 12(d).

Finally, We performed comparison of three satellite enciphered images A, B and C.

As it can be seen from Fig. 12 that the difference of encrypted images cannot be observed by a naked eye thus for the accurate comparison, we computed the correlation coefficient measure of the original image and the three enciphered satellite images. For this purpose, we used the following correlation coefficients formula:

$$r_{xy} = \frac{\mathrm{Con}(x, y)}{\sqrt{D(x)}\sqrt{D(y)}} \tag{11}$$

$$\mathrm{Con}(x, y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(y)) \tag{12}$$

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{13}$$

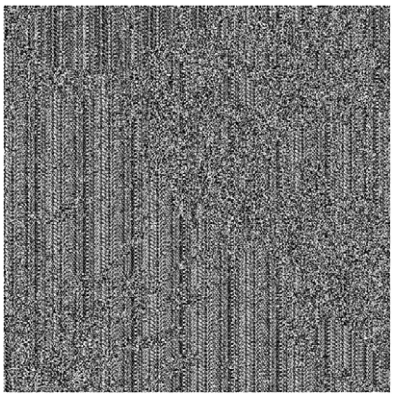
$$E(y) = \frac{1}{N}\sum_{i=1}^{N} y_i \tag{14}$$

$$D(x) = \frac{1}{N}\sum_{j=1}^{N}(x_i - E(x))^2 \tag{15}$$

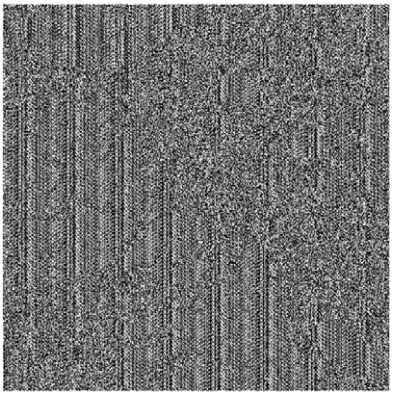
$$D(y) = \frac{1}{N}\sum_{j=1}^{N}(y_i - E(y))^2 \tag{16}$$

where x and y are the values of corresponding pixels in the two enciphered satellite images to be compared. In Table 5, the results of the correlation coefficients between the corresponding pixels of the three enciphered satellite images A, B

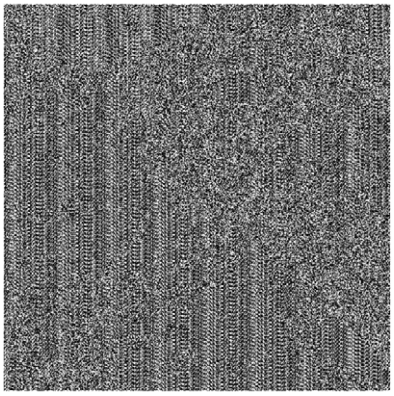
(a) Original image.



(b) Encrypted image with key "12345678901234567890123456789012".



(c) Encrypted image with key "22345678901234567890123456789012".



(d) Encrypted image with key "12345678901234567890123456789033".

Fig. 12. Key sensitivity test result with the proposed algorithm.

Table 5
Correlation coefficients between the corresponding pixels of the three different encrypted images obtained by using a slightly different secret key of an image shown in Fig. 12.

Image 1	Image 2	Correlation coefficient
Encrypted image A Fig. 12(b)	Encrypted image B Fig. 12(c)	0.0076
Encrypted image B Fig. 12(c)	Encrypted image C Fig. 12(d)	0.0798
Encrypted image C Fig. 12(d)	Encrypted image A Fig. 12(b)	0.0201

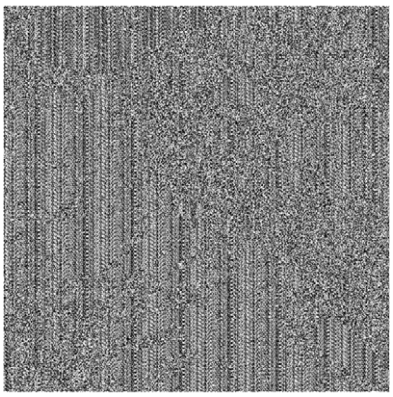
and C have been delineated. It is very prominent from the experimental results that there is no correlation among three enciphered satellite images even though these have been produced by slightly different secret keys. In our experimentation, the key sensitivity analysis shows that change of one bit in secret key will result a completely different encrypted image by more than 99% in terms of pixel gray-scale values.

Moreover, in Fig. 13, we have shown some more attempts to decrypt an enciphered satellite image with slightly different secret keys than the one, which was used in the encryption process of the original satellite image. Particularly, in Fig. 13(a) and (b) respectively, the original satellite image and the enciphered satellite image produced using the secret key "12345678901234567890123456789012" are shown. Whereas the Fig. 13(c) and (d) show the images after the decryption of the enciphered satellite image (as shown in Fig. 13(b)) with the secret keys "12345678901234567890123456789012" and "111456789 01234567890123456789012". It is obvious from the experimental results that the decryption with a slightly different key fails completely therefore, the presented satellite image cryptosystem is highly key sensitive to its secret keys.

4.6. Performance evaluation

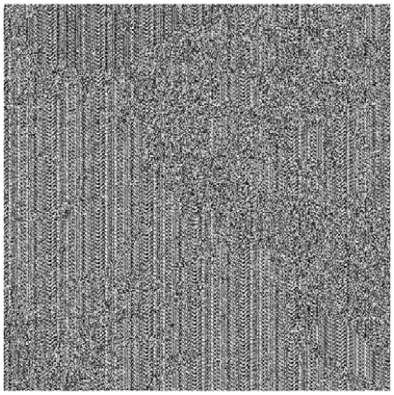
Apart from the security analysis by evaluating statistical analysis and measurements, some other very important issues on satellite image encryption and decryption need to be considered. These issues include the performance and efficiency for real-time applications while doing satellite image encryption/decryption. The results of some experiments are given

(a) Original image.



(b) Decrypted image with key "12345678901234567890123456789012".

(c) Decrypted image with key "12345678901234567890123456789012".



(d) Decrypted image with key "11145678901234567890123456789012".

Fig. 13. Key sensitivity test result with the proposed algorithm.

Table 6
Comparative speed test experiment results of the proposed chaos-based algorithm and tradition satellite image encryption/decryption algorithms on Boston image.

Satellite image encryption/decryption algorithm	Avg. time taken for encryption (s)	Avg. time taken for decryption (s)	Peak time taken for encryption (s)	Peak time taken for decryption (s)	Minimum time taken for encryption (s)	Minimum time taken for decryption (s)
Simple-DES	5.42	5.47	5.53	5.61	5.31	5.39
Triple-DES	5.76	5.73	6.14	6.10	5.52	5.50
AES	0.44	0.46	0.56	0.56	0.38	0.41
Proposed algorithm	0.31	0.28	0.39	0.34	0.29	0.26

to prove the proposed algorithm's encouraging performance and efficiency. The proposed algorithm is faster in speed as compare to other traditional image encryption algorithms like AES, Simple-DES and Triple-DES. In addition, each set of the timing tests shown in Table 6 was executed 5 times, and we report the average of the times thereby obtained.

The experimental results of simulation show that the average encryption and decryption processing time of the proposed algorithm is 0.31 s and 0.28 s respectively, and the peak speed can reach up 0.39 s for encryption and for decryption 0.34 s using SINE chaotic map while all other combination of chaotic maps takes less time for encryption and decryption for Boston image of size 1000×1000.

Table 6 summarizes the encryption/decryption speeds for the proposed chaos-based satellite image encryption algorithm and other traditional satellite image encryption algorithm like AES, Simple-DES and Triple-DES on Boston satellite images as shown in Fig. 5. The simulation results proves that proposed chaos-based satellite image encryption/decryption algorithm have edge in performance and it takes less time to perform encryption and decryption operations for same Boston satellite as compare to other traditional satellite image encryption/decryption algorithms.

AES satellite image encryption algorithm takes far less time to perform same operations as compare to Simple-DES and Triple-DES. However, the proposed algorithm is far better in speed than the AES algorithm. Hence, we can easily conclude that our chaos-based satellite image algorithm outperforms compare to AES, Simple-DES and Triple-DES.

5. Conclusion

To overcome security, performance, privacy and reliability issues of satellite imagery, in this paper, a new chaos-based symmetric key cryptosystem has been proposed. In the presented system, the idea of chaotic cryptography using external secret key has been extended by including multiple chaotic maps named as Logistic, Henon, Tent, Cubic, Sine and Chebyshev for enhancing the key space and security for satellite imagery. The simple logical XOR and one time multiple key generation processes have been carried out for satellite image encryption and decryption for high level of security and performance. We have performed key space analysis, key sensitivity analysis and statistical analysis to demonstrate the security, performance and reliability of the proposed satellite image encryption system. According to the comparative, theoretical and experimental results, we conclude that the proposed chaos-based satellite image cryptosystem is useful for real-time satellite image encryption and decryption, in order to keep the storage and transmission process secure and reliable. The proposed system is not just limited to this area, but can also be widely applied in the secure storage and transmission of confidential multimedia images over the Internet and/or any shared network environment.

References

[1] G.M. Nair, Role of communications satellites in national development, IETE Technical Review 25 (2008) 3–8.
[2] A.J. Menezes, P.C. Oorschot, S.A. Vanstone, Handbook of Applied Cryptography, CRC Press, Boca Raton, FL, USA, 1996.
[3] W. Stallings, Cryptography and Network Security: Principles and Practice, Prentice-Hall, Upper Saddle River, NJ, USA, 1999.
[4] D.R. Stinson, Cryptography: Theory and Practice, 2nd ed., CRC, Boca Raton, USA, 2002.
[5] J.C. Yen, J.I. Guo, A new chaotic image encryption algorithm, in: Proceeding of National Symposium on Telecommunications, 1998, pp. 358–362.
[6] J.C. Yen, J.I. Guo, A new chaotic mirror-like image encryption algorithm and its VLSI architecture, Pattern Recognition and Image Analysis 10 (2) (2000) 236–247.
[7] J.C. Yen, J.I. Guo, Efficient hierarchical chaotic image encryption algorithm and its VLSI realization, IEE Proceedings. Vision, Image & Signal Processing 147 (2) (2000) 167–175.
[8] S. Li, X. Mou, Y. Cai, Improving security of a chaotic encryption approach, Physics Letters A 290 (3) (2001) 127–133.
[9] Z.P. Jiang, A note on chaotic secure communication systems, IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications 49 (1) (2002) 92–96.
[10] M.Z.H. Sarker, M.S. Parvez, A cost effective symmetric key crypto-graphic algorithm for small amount of data, in: Proceedings of the 9th IEEE International Multi topic Conference, December 2005, pp. 1–6.
[11] M.K. Khan, J. Zhang, L. Tian, Chaotic secure content-based hidden transmission of biometrics templates, Chaos, Solitons and Fractals 32 (2007) 1749–1759.
[12] J. Fridrich, Symmetric ciphers based on two-dimensional chaotic maps, International Journal of Bifurcation and Chaos 8 (6) (1998) 1259–1284.
[13] S. Li, X. Mou, Y. Cai, Improving security of a chaotic encryption approach, Physics Letters A 290 (2001) 127–133.
[14] M.K. Khan, J. Zhang, Chaotic hash-based fingerprint biometric remote user authentication scheme on mobile devices, Chaos, Solitons and Fractals 35 (2008) 519–524.
[15] M.S. Baptista, Cryptography with chaos, Physics Letters A 240 (1998) 50–54.
[16] Z. Kotulski, J. Szczepanski, Discrete chaotic cryptography (DCC): New method for secure communication, in: Proceedings of NEEDS'97, 1997.
[17] R. Matthews, The derivation of a chaotic encryption algorithm, Cryptologia XII (1) (1989) 29–42.
[18] T. Xiang, K.W. Wong, X.F. Liao, An improved chaotic cryptosystem with external key, Communications in Nonlinear Science and Numerical Simulation 13 (9) (2008) 1879–1887.
[19] Z. Kotulski, J. Szczepanski, K. Gorski, Z. Paszkiewicz, A. Zugaj, Application of discrete chaotic dynamical systems in cryptography–DCC method, International Journal of Bifurcation and Chaos 9 (1999) 1121–1135.
[20] E. Alvarez, A. Fernandez, G.P. Jimenez, A. Marcano, New approach to chaotic encryption, Physics Letters A 263 (1999) 373–375.
[21] W.K. Wong, L.P. Lee, K.W. Wong, A modified chaotic cryptographic method, Computer Physics Communications 138 (2000) 234–236.
[22] K.W. Wong, A fast chaotic cryptography scheme with dynamic look-up table, Physics Letters A 298 (2002) 238–242.
[23] M.K. Khan, J. Zhang, Multimodal face and fingerprint biometrics authentication on space-limited tokens, Neurocomputing 71 (13–15) (2008) 3026–3031.
[24] N.K. Pareek, V. Patidar, K.K. Sud, Discrete chaotic cryptography using external key, Physics Letters A 309 (2003) 75–82.
[25] I. Ziedan, M. Fouad, D.H. Salem, Application of data encryption standard to bitmap and JPEG images, in: Proceedings Twentieth National Radio Science Conference, NRSC, March 2003, pp. C16.
[26] N. Fishawy, O.M.A. Zaid, Quality of encryption measurement of bitmap images with RC6, MRC6, and Rijndael block cipher algorithms, International Journal of Network Security 5 (3) (2007) 241–251.
[27] C.E. Shannon, A mathematical theory of communication, The Bell System Technical Journal 27 (3) (1948) 379–423. 623–656.
[28] B. Schneier, Applied Cryptography: Protocols, Algorithms and Source Code in C, 2nd ed., John Wiley and Sons, 1996.
[29] C.E. Shannon, Prediction and entropy of printed English, The Bell System Technical Journal 30 (1950) 50–64.