

Secure Data Compression using Chaotic Images

Sai Venkatesh Balasubramanian

*Sree Sai Vidhya Mandhir, Mallasandra, Bengaluru-560109, Karnataka, India.
saivenkateshbalasubramanian@gmail.com*

Abstract

A Chaos based embedding process for textual data offering high capacity and high security simultaneously is designed and implemented. A chaotic image, obtained using a frequency dependant driven chaotic system is used as the data carrier in which textual data is embedded. The decryption and subsequent performance analyses reveal a high fidelity with a mean square error of around 0.0009 percent and a compression ratio increasing nonlinearly with text size, with ratio values more than 150:1 obtained for significantly large texts. Moreover, a very high level of security leading to up to 60 percent of mean square error values even for 1 percent misalignment in the decryption process is observed. The extreme simplicity of implementation coupled with the twin advantages of high compression ratios and high security forms the highlight of the present work.

Keywords: Frequency Dependent Chaos, Ring Oscillator, Adaptive Bias Control, Electrical Solitons, Wavelet

1. Introduction

The rise and growth of the internet and networking in recent years has led to an information explosion [11, 10, 30]. Buzzwords such as Big Data, Cloud Computing and Internet of Things are used extensively in industry and academia alike and the demands for higher data storage capacities are increasing by the day [31, 27, 20]. The urgent necessity of innovative techniques promising large compression ratios with minimal loss cannot be over-stated [31, 26, 7, 2].

Fortunately, advances made in the fields of nonlinear signal processing, wavelet transforms, chaos and fractal theory have enabled efficient techniques that achieve compression ratios of more than 100:1, where the compression ratio is defined as the ratio of the uncompressed data size to the compressed data size. [2, 24, 5, 29].

Specifically, fractal image compression, proposed by Barnsley et al. have exploited the self-similar and fractal pattern ‘domains’ in text and images and by eliminating the redundant and repetitive structures, have achieved compression ratios in the excess of 100:1 [3, 9, 4]. On the other hand, mainstream techniques such as JPEG and MPEG achieve maximum compression ratios of 30:1 with an acceptable loss factor [8, 22].

However, in the present era of Big Data, high storage capacity is not the only pressing need. The recent surge in cybercrimes such as hacking has highlighted a critical need for data security [12, 6]. While secure embedding techniques such as steganography and watermarking exist, they compromise heavily on the embedding capacity [13, 18].

The present work purports to the design and implementation of a chaos-based embedding process for textual data using images as the storage medium. Specifically, a frequency dependant chaotic system is proposed where the control parameter, the frequency ratio of the input signals forms a secure embedding ‘key’. This chaotic system is used to generate a chaotic image. The textual data is embedded onto the image. This process thus forms a one-step process covering both compression and encryption, enabling the twin advantages of capacity and security simultaneously. The compression ratios obtained are greater than 150 and show a nonlinear dependence on the text size. The decryption of the text is observed to yield very low bit error rates of the order of 0.0009 percent. The simplicity of implementation of the present work coupled with extremely high compression ratios and secure encryption form the highlights of the present work.

The remainder of the work is structured as follows. ‘Methodology’, described in Sec. II elaborates on the two principal steps used in the embedding process - The Generation of a Chaotic ‘Carrier’ image, and the Embedding Process where the text to be compressed is converted into a pixel pattern, normalized and additively embedded in the image. The corresponding decryption process is also outlined. The ‘Performance Assessment’, described in Sec. III characterizes the performance of the embedding process pertaining to three principal aspects - Capacity, Fidelity and Security. Standard parameters such as Compression Ratio and Mean Square Error are used for ascertaining the performance of the proposed embedding process. Also, the variation of such parameters with the nature of chaos generated, as described by the bifurcation diagram, as well as with the variations in the text size are discussed.

2. Methodology

The embedding process comprises of two functional units. The first pertains to the generation of the chaotic image, using a frequency-dependant iterative map. The second unit converts the text to be compressed into a pixel pattern, and after appropriate normalization, additively embeds it into the chaotic ‘carrier’ image generated in the first unit. This embedding process forms a one-step encryption-compression process. The de-compression then comprises of the corresponding inverse operations - subtractive decoding and denormalization. The compression-decompression process is illustrated as a block diagram in Fig (1).

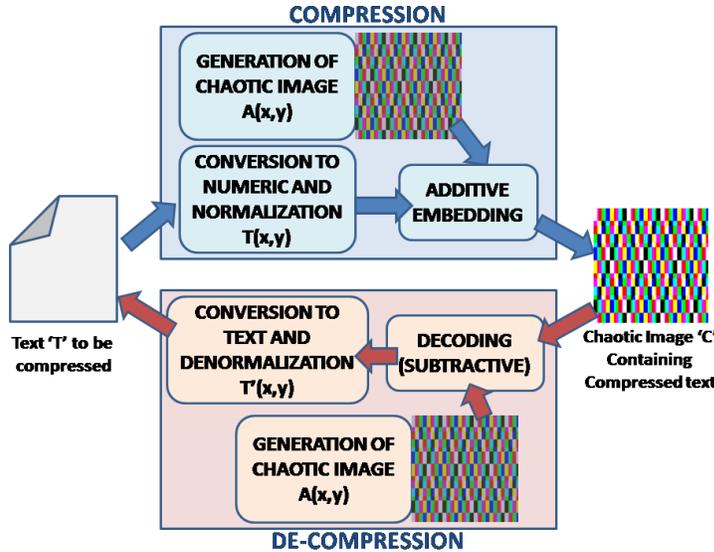


Figure 1: Block Diagram of the proposed Embedding Process

2.1. Generation of Chaotic ‘Carrier’ Image

The use of a chaotic system to facilitate a secure high capacity embedding process is the backbone of the present work.

The first step in the generation of any chaotic system is the definition of an iterative map, which defines the evolution of a system using a control parameter r [28]. The presence of nonlinearity in the iterative map ensures that for some control parameter values, the system enters chaotic regimes [28, 1]. Conventionally, iterative map nonlinearities are defined on amplitudes by employing polynomial based functions [1]. However, the present work encompasses a paradigm shift from this perspective, and defines a frequency based iterative map, more similar in tone to the phase-based chaotic iterative functions used in standard circle maps [14]. The iterative function used in the present work for a two-input system is given as follows:

$$f_o(i + 1) = \text{mod}(f_o(i) + \frac{f_2}{f_1} - V(f_o(i)), \pi) \quad (1)$$

Here the f_o terms denote the output frequencies, whereas f_1 and f_2 denote the frequencies of the input signals. $V(f_o)$ denotes the input signal waveform employed in the chaotic system. The salient features of the above mentioned iterated function are as follows:

1. The nonlinearity, provided by the modulus function represents the switching operation, physically implemented using differential amplifiers or XOR gates [23].
2. The control parameter $r=f_2/f_1$ is an additive parameter and determines when the system transits from order to chaos and vice versa.
3. The $V(f_o)$ introduces a signal dependence, thus enabling the controlling of chaos by changing the waveform used as input.

The bifurcation plot for a sinusoidal input $V(f_o(i))=\sin(f_o(i))$ is given in Fig.(2).

From the diagram, two inferences stand out:

1. The pattern repeats itself with a period of 1.

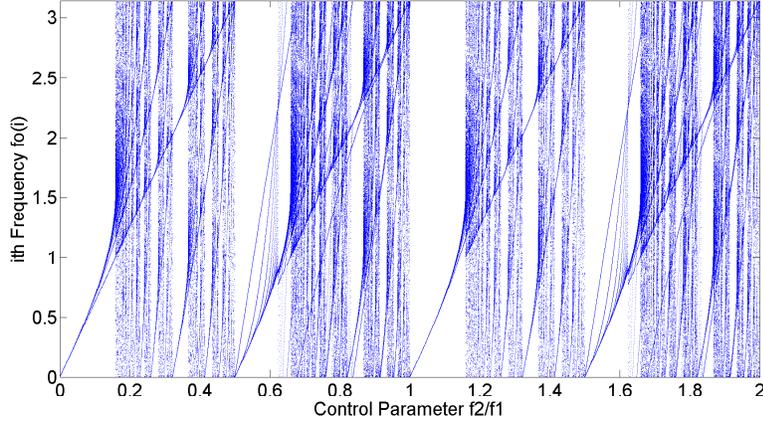


Figure 2: Bifurcation Diagram of the chaotic system with a sinusoidal input

2. Control values close to integers (1,2,...) and half-integers (0.5, 1.5,...) give rise to order whereas non integral ratios such as 0.22 or 0.41 give rise to chaos.

A similar bifurcation diagram is plotted for square wave input, and is shown in Fig.(3).

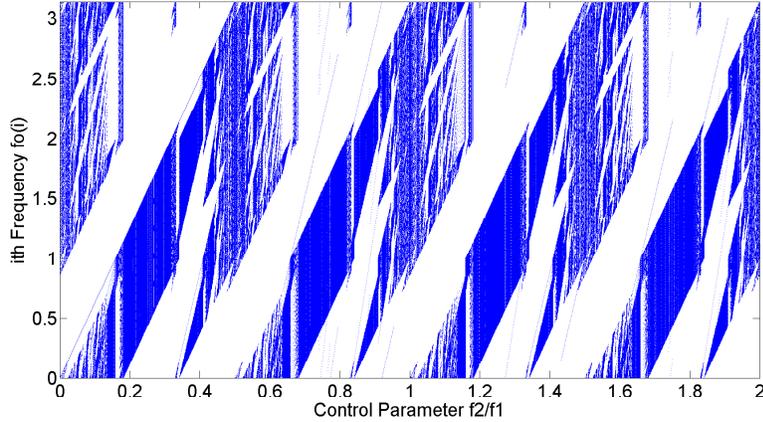


Figure 3: Bifurcation Diagram of the chaotic system with a square wave input

The square wave bifurcation diagram shows chaotic behavior for a larger set of values of control parameter r compared to the sinusoid, and owing to this reason, square wave signals are used in the present work.

To understand the nature of the proposed iterative map in chaotic regimes, the cobweb plot, a graphical visualization of the long term status of the system under repeated application of the proposed iterative map, is plotted for the frequency ratio of π in Fig. (4). [28, 1].

The features of the ‘driven’ chaotic system thus formulated is in principle extended to a 2-dimensional spatial coordinate system (an image), and this forms the basis for the generation of a ‘carrier’ image used as a storage element in the present work. The whole embedding process is implemented in MATLAB. The 2D system is formulated using two square wave signals as follows:

$$A(x, y) = \text{Square}(2\pi f_1 x) \times \text{Square}(2\pi f_2 y) \tag{2}$$

where \times represents the cross product operation. The pixel intensity pattern of $A(x, y)$ results in a grayscale image with the spatial coordinates defined by discrete vectors x and y . Three such functions defined using different frequencies for Red, Green and Blue result in a ‘chaotic’ image. The base frequencies f_1 terms normalized to the sampling frequency 1 are set as 0.01, 0.02 and 0.015 for red, green and blue, and the corresponding second frequencies f_2 terms are computed using a frequency ratio value $r=\pi$ for all three colors.

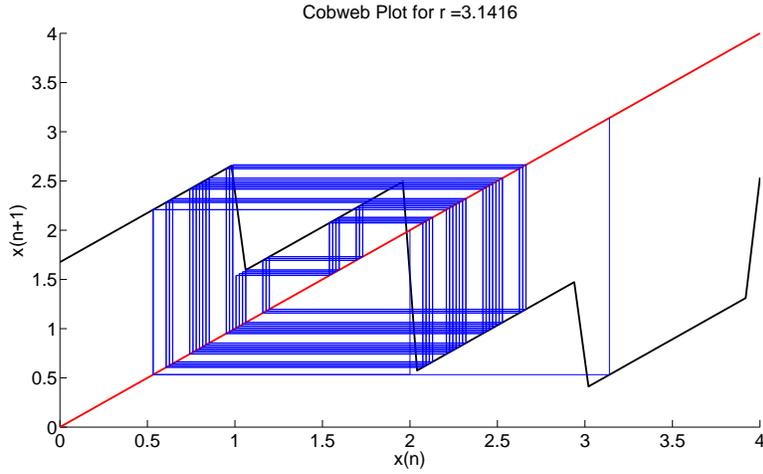


Figure 4: Cobweb Plot for a Frequency Ratio of π

The height and width of the image is determined by the volume of data to be embedded, and the pixel values are normalized to half-capacity (128).

The image thus generated is a square image shown in Fig.(5), with the dimension of 440×440 .

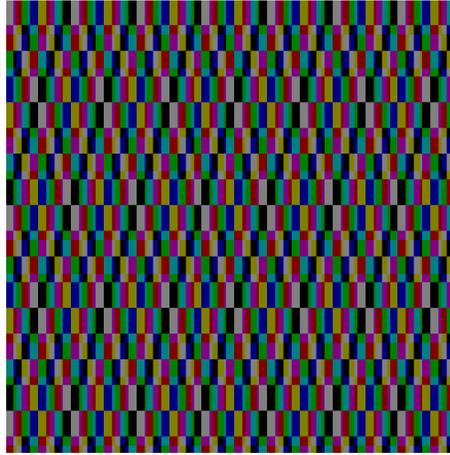


Figure 5: The chaotic ‘carrier’ image (440x440 png)

In Portable Network Graphics (png) format, the image occupies 2.27kB storage.

The chaotic/fractal nature of this image is understood by computing the fractal dimension, using the Minkowski Bouligand Box Counting Method [19]. In this method, various square ‘boxes’ of different sizes e are formed and for each size e , the number of boxes $N(e)$ required to cover the entire set is computed. The fractal dimension D is then given by

$$D = \lim_{e \rightarrow 0} \frac{\log(N(e))}{\log(e)} \quad (3)$$

For the generated image shown in Fig.(5), the fractal dimension is obtained as 1.807, indicating the presence of self-similarity in the generated image.

2.2. The Embedding Process

The next step is the embedding process. A text based file is read and the text is converted into numeric signal using ASCII for Latin based scripts and Unicode for other scripts such as Chinese or Sanskrit [25, 15]. A significant feature of the present work is that the embedding process proposed builds upon the Lempel-Ziv based Arithmetic Coding (LZ77) inherently present in the Portable Network Graphics (PNG) image format. [31, 26, 7].

The converted numeric signal is normalized to 128 and is shown as a 440×440 image in Fig.(6). This, coupled with the earlier image normalization to 128 ensures that an additive embedding will not result in an overflow. Each character

of the text is mapped to a pixel in the image file created earlier, using a simple raster embedding pattern [16]. Thus, three characters are embedded into the Red, Green and Blue values of a single pixel.

In the present work, the text-only version of the entire story of Gulliver's travels, obtained from the Gutenberg Archiving Project link is used as the text to be embedded. The size of the txt file is 566kB. The text file is read, converted into numeric using ASCII, and appropriate zero padding is done to ensure size match. The data is then additively embedded into a 440x440 image, and the embedded image is shown in Fig.(7).

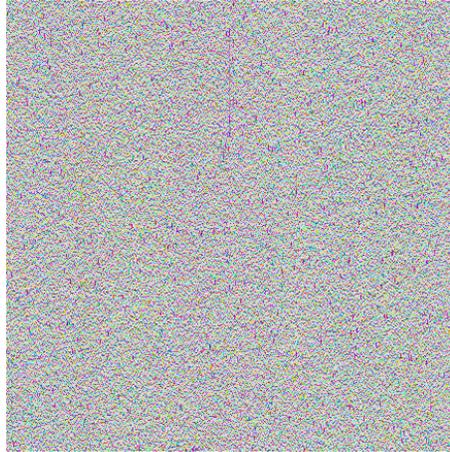


Figure 6: The text of Gulliver's Travels shown as an image

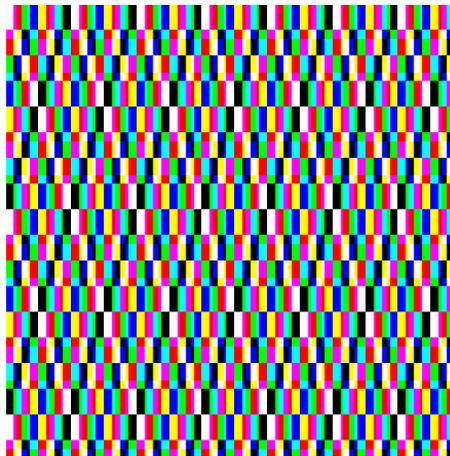


Figure 7: The chaotic image containing the text of Gulliver's Travels (440x440 png)

The image contains the text version of Gulliver's Travels embedded onto the image shown in Fig. (7). The total size of this image in png format is 2.36kB.

The decryption is performed by just subtracting the 'carrier' image shown in Fig. (5) from the embedded image of Fig.(7), followed by denormalizing and converting the signal obtained back to ASCII/Unicode text.

In the present work, the decryption performed yielded back the text only version of Gulliver's travels with a size of 566kB.

3. Performance Assessment

In order to assess and characterize the performance of the embedding process elaborated earlier, the performance is characterized with regards to capacity, fidelity and security, as follows.

3.1. Capacity

The Compression Ratio C is typically defined as follows [3, 9, 4]:

$$C = \frac{\text{Total Size of Uncompressed Data (SU)}}{\text{Total Size of Compressed Data (SC)}} \quad (4)$$

In the present work, the total size of uncompressed data SU includes the sizes of the text as well as the original image shown in Fig.(5). The total size of compressed data SC is the size of the image shown in Fig.(7).

For the text version of Gullivers travels, C is thus obtained as 240.79. This enormous capacity is attributed to the following factors:

1. The Portable Network Graphics (png) format, itself being lossless has been shown to achieve compression ratios of around 3.5 [7].
2. The chaotic ‘carrier’ image generated in Fig.(5) effectively exploits the enormous storage offered by the color space. Since each of the Red, Green and Blue components of a pixel can store a 8 bit value (maximum capacity of 255 each), the total combination of colors possible is 16.5 million. In order to validate this statement, the embedding process elaborated earlier has been carried out by replacing the ‘carrier’ image in Fig. (5) by a pure black image, with absolutely no color distribution. It is found that the compression ratio in this case is 3.2 which is close to the inherent compression ratio of the png format itself.
3. The frequency ratio of π selected for embedding plays a major role in boosting the compression ratio. The concept can be intuitively understood as follows. In a purely amplitude dependant system consisting of two signals V_1 and V_2 , there are two variables V_1 and V_2 for storage of data. However, in a frequency dependant chaotic system consisting of V_1 and V_2 , the storage variables are the frequencies f_1 and f_2 as well as the mixing products $2f_1$, $2f_2$, $3f_1$, $3f_2$, $2f_1 - f_2$, $2f_2 - f_1$, $2f_1 + f_2$, $2f_2 + f_1$ and so on, thus offering a steep improvement in storage space compared to the amplitude case [21].
4. With the image $A(x, y)$ defined as in equation (2) and giving rise to the harmonic components as enumerated above, the text data converted to numeric and reshaped forms a pixel pattern defined by a 2 dimensional function $T(x, y)$ and visualized as in Fig. (6). The additive embedding then generates the compressed image $C(x, y)$ as

$$C(x, y) = A(x, y) + T(x, y) \quad (5)$$

A typical text $T(x, y)$ has multiple frequency components, as suggested by successful application of the Zipf’s Law [17], and it is easy to visualize how certain frequency components of the text easily match with the corresponding counterparts in the image, such matches creating redundancies which will be compressed greatly due the Lempel-Ziv LZ77 coding used in the png format [7]. Thus, the more chaotic the frequency ratio f_2/f_1 , the more the harmonics produced by $A(x, y)$, the more the matches and redundancies, and the more the compression ratio C .

To observe the dependence of compression ratio on frequency ratio, the embedding process is carried out with increasing frequency ratios in steps of 0.05 from 3 to 4. For each ratio, the compression ratio is noted and the plot thus obtained is shown in Fig.(8). The error rates for each case are obtained consistently as 0.0008 percent.

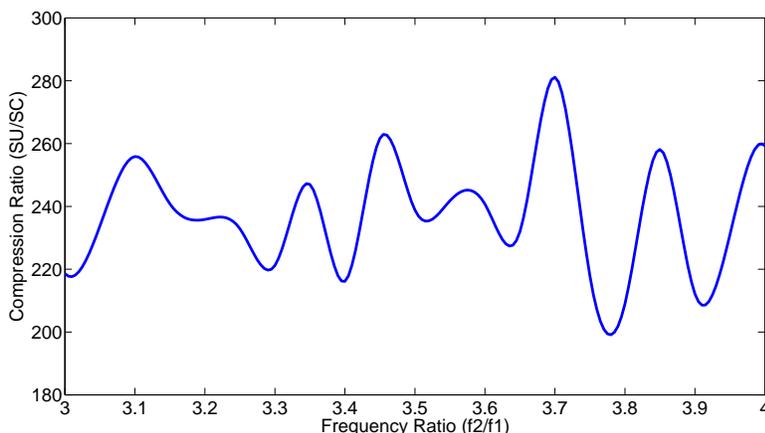


Figure 8: Dependence of Compression Ratio on Frequency Ratio

As can be seen from Fig.(8), different frequency ratios yield different compression ratios, in a wide range from around 210 to 280. This variation is due to the variation of the intensity of chaos for different frequency ratio values as

Table 1: Performance Assessment for Various Classical Texts

Text	Language	Image Dimension (Pixels)	Text Size (Bytes)	Compression Ratio	MSE (Percent)	Fractal Dimension
The Daffodils	English	15	634	6.8703	0.0006	0.8732
San Bai Qian	Chinese	29	2499	17.9765	0.0013	0.923
Odyssey Book 1	Greek	120	43110	83.6322	0.0011	0.7925
Sura Al Baqarah	Arabic	197	115712	153.3089	0.0008	0.929
Bhagavad Gita	Sanskrit	218	142336	138.5084	0.0008	0.8821
Silappathikaram	Tamil	223	148480	141.6622	0.0008	0.9438
Genesis	Hebrew	273	223232	173.9126	0.0010	0.9037
Psalms	English	294	257024	185.4852	0.0008	0.9886
Gullivers Travels	English	440	579584	240.7923	0.0008	0.9886
US Census Data	English	830	2055208	324.2077	0.0007	0.9899

demonstrated by the bifurcation diagram in Fig.(8). Patterns similar to the one shown in Fig.(8) are obtained for any group of frequency ratios from an integer n to the next integer $n + 1$.

It is noteworthy that such high compression ratios are obtained without employing any form of arithmetic coding techniques such as Huffman encoding for the text data [31, 26]. Using such coding techniques in the text in addition to the techniques proposed in the present work could further boost the compression ratio.

3.2. Fidelity

The fidelity of the embedding process is characterized by the Mean Square Error between the numeric values of the original and decrypted texts, normalized as a percentage of the maximum value (128 for ASCII). Thus, if the original Text file converted to a Numeric Signal is N , and the decrypted Numeric Signal is N_o , the MSE is given as follows.

$$MSE(Percentage) = \frac{(N - N_o)^2}{128} \quad (6)$$

The MSE obtained for Gullivers Travels is 0.00069 percent, thus indicating a remarkable fidelity of decryption.

In order to further characterize the capacity and fidelity of the proposed embedding system and the dependence of performance on the text properties, various texts of classical literature in various languages are tested. Also, English texts of various different sizes are also considered to observe any dependence of performance on text size. For each case, the capacity parameters as well as fidelity parameters are recorded and tabulated, as shown in Table 1. The fractal dimension of each text obtained using the Minkowski Bouligand Box Counting Method is also given [19].

From Table 1, two principal inferences can be obtained:

1. The fidelity of the embedding process is consistently high with a MSE in the vicinity of 0.0009 percent regardless of the language and text size.
2. The compression ratio varies with respect to text size, with larger ratios obtained for larger text sizes.

The dependence on compression ratio on the text size is fitted to a fourth order polynomial of the form

$$y = \sum_{i=0}^4 p_i x^i \quad (7)$$

where x denotes the text size in Bytes and y denotes the compression ratio. The polynomial coefficients are given as $p_0 = 14.25$; $p_1 = 0.001452$; $p_2 = -4.444 \times 10^{-9}$; $p_3 = 5.55 \times 10^{-15}$; $p_4 = -1.8 \times 10^{-21}$. The closeness of the fit is obtained as 99.94 percent, and the fitted curve alongside the original dataset is given in Fig.(9).

Though the exponentially increasing dependence of compression ratio on text size seems counterintuitive at first glance, it is completely justified by the vast number of mixed frequency components and the associated redundancies as explained earlier.

Typical compression ratios reported in literature for standard compression and encoding schemes are tabulated platform agnostically in Table 2 [7, 10, 22, 18, 25]. The state-of-art compression/encoding schemes are broadly classified under three headings:

1. Lossless Compression Techniques - very high fidelity with low to moderate amount of security.
2. Lossy Compression - moderate amount of loss and security with high compression ratios.

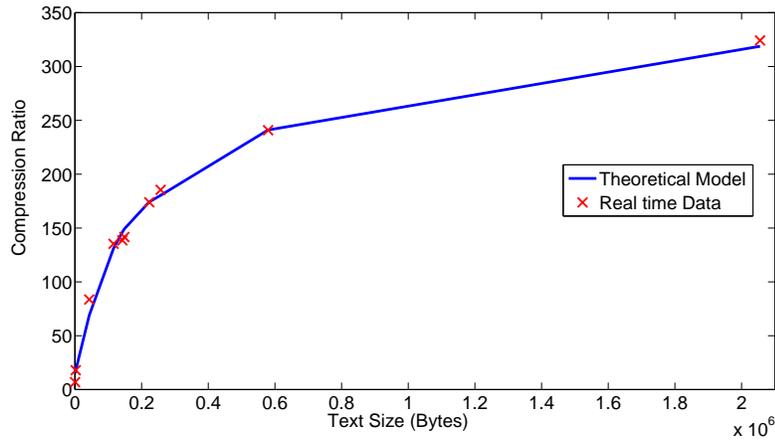


Figure 9: Compression Ratio as a function of Text Size

Table 2: Performance Assessment of State-of-the-art Compression Techniques

Algorithm Name	Category	Reported CR Value
Portable Network Graphics	Lossless	3.5
Netapp SnapMirror	Lossless	3.5
Netapp OnTap	Lossless	2
WinZip	Lossless	2-10
Joint Photographic Experts Group (JPEG)	Lossy	3.2 (5-10 percent MSE)
Moving Picture Experts Group (MPEG)	Lossy	10-20 (5-10 percent MSE)
Cyclic Redundancy Check (CRC32/RS)	Encryption	NIL (Redundancies Created)

3. Encryption - security is key, and very rarely compression is seen.

As can be seen from the table, the compression ratio of around 250 obtained in the present work is much higher than most state-of-the-art compression techniques combining the best features of encryption, lossless and lossy compression techniques. The unusually high compression ratio arises due to the color space and frequency related factors explained earlier.

3.3. Security

In order to characterize the security, a decryption of the compressed image of Gullivers travels is attempted, but with the frequency ratios not aligned to the ones used during encryption.

Specifically, the frequency ratios of the red, green and blue components were misaligned by a factor of 1 percent ($r_{new} = r(1 + 0.01)$), and the corresponding MSE values obtained are 61 percent, 63 percent and 62.5 percent. These values are in stark contrast to the MSE value of 0.0008 percent obtained during correct decryption.

In order to understand the significance of exact frequency match, the difference between the original ‘carrier’ image of Fig.(5) and the ‘erroneous’ ‘carrier’ image obtained by 1 percent misalignment of all the three frequency components is plotted as an image in Fig.(10).

As seen from Fig.(10), the difference between the original and erroneous carriers severely hampers the pixel distributions, thus guaranteeing erroneous decryption of data.

4. Conclusion

A frequency dependant chaotic system is proposed and characterized and a chaotic image generated from such a system is used as a storage ‘carrier’ to securely embed textual data. The performance assessment of the embedding process revealed the following vital points:

1. High fidelity characterized by a MSE of around 0.0009 percent is obtained regardless of data size or language.
2. The Compression ratio increases nonlinearly with the text size.



Figure 10: Difference between original ‘carrier’ and a 1 percent misaligned ‘carrier’

3. High compression ratios of around 150-300 are obtained for significantly large texts, regardless of text language, and this is attributed to the chaos generated by appropriate choice of the frequency ratio, coupled with the Lempel-Ziv coding present inherently in the png format used.
4. Extreme sensitivity to frequency ratios is seen, with a 1 percent misalignment in the frequency ratio causing an MSE as high as 63 percent, and this assertively establishes the security offered by the proposed technique.

Though a detailed and thorough investigation of the mechanism of chaos generation and the embedding process awaits, the results discussed in the present work establish the fact that this embedding process serves as a one-step compression plus encryption process, guaranteeing the twin advantages of high capacity and high security. These advantages, coupled with the extreme simplicity of implementation form the crux of the present work. The embedding process used in the present work thus opens the doors for a golden new era, the era of ‘Affordable Big Data’. The hallmarks of such an era would be the usage of extremely simple yet effective techniques for securely compressing Big Data such as DNA Genome Sequences, Automated Sensor Data, Financial Records and other Multimedia based Signals.

References

- [1] M. Ausloos, M. Dirickx, *The Logistic Map and the Route to Chaos: From the Beginnings to Modern Applications*, (Springer, US, [2006]).
- [2] K. E. Barner and G. R. Arce, *Nonlinear Signal and Image Processing: Theory, Methods, and Applications*, (CRC Press, U.S, 2003).
- [3] M. F. Barnsley, A. D. Sloan, *Chaotic Compression*, Computer Graphics World, **3** (1987).
- [4] M. F. Barnsley, *Fractals Everywhere*, (Courier, Dover, 2008).
- [5] E. Bilotta and P. Pantano, *A gallery of Chua attractors*, (World Scientific, Singapore, 2008).
- [6] R. Broadhurst, P. Grabosky, M. Alazab, S. Chon, *Organizations and Cyber crime: An Analysis of the Nature of Groups engaged in Cyber Crime*, Int. J. Cybercriminology, **8** (2014).
- [7] D.S. Cruz, C. Ebrahimi, J. Askelof, M. Laarsson, C. A. Chritsopoulos, *JPEG 2000 still image coding versus other standards*, SPIE Digital Image Processing XXIII, **4115**, 446-454 (2000).
- [8] Z. Fan, R. L. D. Queiroz, *Identification of bitmap compression history: JPEG detection and quantizer estimation*, IEEE Trans. Image Processing, **12**, 230-235 (2003).
- [9] Y. Fisher, *Fractal Image Compression*, Fractals, **2** (1994).
- [10] G. B. Giannakis, F. Bach, R. Cendrillon, M. Mahoney, J. Neville, *Signal Processing for Big Data*, IEEE Signal Processing Magazine, **31**, 15-16 (2014).
- [11] M. Hilbert, *How much of the global information and communication explosion is driven by more, and how much by better technology?*, Wiley Journal of the Association for Information Science and Technology, **65**, 856-861 (2014).
- [12] K. E. Himma, *Internet Security: Hacking, Counterhacking, and Society*, (Jones and Bartlett, UK, 2007).
- [13] N. Hopper, L. VonAhn and J. Langford, *Provably secure steganography*, IEEE Trans. on Computers **58**, 662-676 (2009).
- [14] M. H. Jensen, P. Bak, T. Bohr, *Complete Devil’s Staircase, Fractal Dimension, and Universality of Mode-Locking Structure in the Circle Map*, Phys. Rev. Lett. **50**, 1637 (1983).
- [15] J. Korpela, *Unicode Explained*, (O’Reilly, US, 2006).
- [16] T. H. Lan, M. F. Mansour, A. H. Tewfik, *Robust high capacity data embedding*, Image Processing 2000, **1**, 581-584 (2000).
- [17] W. Li, *Random texts exhibit Zipf’s-law-like word frequency distribution*, IEEE. Trans. Information Theory, **38**, 1842-1845 (1992).
- [18] P. H. Mahajan, P. B. Bhalerao, *A Review of Digital Watermarking Strategies*, International Journal of Advanced Research in Computer Science And Management Studies, **7** (2014).
- [19] P. Maragos, F. K. Sun, *Measuring the fractal dimension of signals: morphological covers and iterative optimization*, IEEE Trans. Signal Processing, **41**, 108-121 (1993).
- [20] A. McEwan and H. Cassimally, *Designing the Internet of Things*, (Wiley, 2013).
- [21] J. C. Pedro, N. B. Carvalho, *Intermodulation Distortion in Microwave and Wireless Circuits*, (Artech House, US, 2002).
- [22] C. A. Poynton, *Digital Video and HDTV: Algorithms and Interfaces*, (Morgan Kaufman, US, 2003).

- [23] B. Razavi, *RF Microelectronics*, (Prentice Hall, US, 2011).
- [24] A. M. Rufai, G. Anbarjafari, H. Demiral, *Lossy image compression using singular value decomposition and wavelet difference reduction*, Elsevier Digital Signal Processing, **24**, 117-123 (2014).
- [25] D. Salomon, *Data Compression: The Complete Reference*, (Springer, US, 2004).
- [26] D. Salomon, D. Bryant and Giovanni Motta, *Handbook of Data Compression*, (Springer, California, 2010).
- [27] E.Schouten, *IBM SmartCloud Essentials*, (Packt, 2013).
- [28] S. H. Strogatz, *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*, (Westview Press, Cambridge, 2008).
- [29] S. T. Welstead, *Fractal and Wavelet Image Compression Techniques*, (SPIE, US, 1999).
- [30] T. White, *Hadoop: The Definitive Guide*, (O'Reilly, 2010).
- [31] F. Wu, *Advances in Visual Data Compression and Communication: Meeting the Requirements of New Applications*, (CRC Press, US, 2014).