Final Project

Nick Wade, Christian Ortiz

Image Encryption Algorithm Using a 3D Lorenz System

Introduction:

Information encryption has been used to establish secure communication for over a thousand years [8]. The desire for message encryption has created a heavy demand for robust encryption methods, especially in the information age. Image encryption describes the methods to covertly represent imagery information using an encryption algorithm and a shareable key. There are various methods of scrambling the information of an image. One such method is the use of chaotic systems of equations which have broad applications in computer and physical sciences due to their pseudorandomness and high sensitivity to insignificant changes in system parameters and initial conditions. These facts boost applicability in encryption as natural defenses against brute force attacks [6]. In this project, we present a basic method of image encryption using the well-studied three-dimensional Lorenz system of equations, a system that was originally developed to measure atmospheric circulation [11].

A method of encryption is presented that alters the stored information in the pixels that compose a digital image. The algorithm will utilize a 3D Lorenz system by using the chaotic solutions of the Lorenz equations to transform the information contained in each pixel in a pseudorandom manner. For the sake of this project, an image encryption algorithm is considered to be any method of transforming the unencrypted pixel values comprising an image in a way that produces an altogether unintelligible image. And by unintelligible, it is assumed the encrypted image cannot be discerned without decryption.

There is a lot of information that can be stored in a pixel [9][10]. Each pixel contains a value that indicates its brightness and color. Each colored pixel is described as a combination of a trio of colors: red, green, or blue (typically in that order as well) with varying degrees of each. Kaur and Kumar [2] show that a change of just one pixel in the original image can produce drastic changes in the ciphered image. This feature is why an algorithm that displays chaotic behavior is a desirable choice for transforming the information of an image.

The proposed chaotic system-based encryption scheme generates a set of cryptographic values based on the integration composite of the assembled Lorenz system. In other words, it produces an "encrypted" (or randomized) version of each pixel RGB based on factors such as system parameters, initial conditions, and integration spans. These are labeled as *encr_R*, *encr_G*, and *encr_B* in the figure below. Each encrypted pixel RGB should effectively be a random pixel color, regardless of the original pixel RGB. The key set, shown in the figure below, can be created with just 6-digits and, when following the algorithm laid out in the methods section, will correctly obfuscate (and de-obfuscate) the imagery data without fail.

key - DataFrame

Index	X	Y	Z	encr_R	encr_G	encr_B
0	5	-7	12	160	244	66
1	4.97338	-6.98736	11.9851	237	218	147
2	4.94684	-6.97476	11.9704	242	61	41
3	4.92039	-6.9622	11.9556	158	245	58
4	4.89403	-6.9497	11.941	176	249	79
5	4.86775	-6.93724	11.9264	216	243	79
6	4.84156	-6.92482	11.9118	205	201	49
7	4.81546	-6.91245	11.8973	45	21	165
8	4.78944	-6.90013	11.8829	2	54	171
9	4.7635	-6.88785	11.8686	145	116	51
10	4.73765	-6.87562	11.8543	218	15	35
11	4.71189	-6.86343	11.84	28	8	84
12	4.68621	-6.85129	11.8259	48	254	84
13	4.66061	-6.83919	11.8117	51	122	124
14	4.63509	-6.82714	11.7977	118	35	51
15	4.60966	-6.81514	11.7837	63	122	161
16	4.58432	-6.80318	11.7698	47	255	6
17	4.55905	-6.79127	11.7559	60	9	32

Once the encryption/decryption procedure in the methods section below is hard-coded and precisely verified, a simple handoff of digital image data and the 6-digit key sequence can yield secure endless data sharing between users. The usage of the single 6-digit key sequence between encrypting and decrypting users is an example of symmetric encryption.

Lorenz Systems:

Marsden, J. E., & McCracken, M. inform us that “the Lorenz equations are an idealization of the equations of motion of the fluid in a layer of uniform depth when the temperature difference between the top and the bottom is maintained at a constant value...” [15] This derivation summary is from Lorenz's 1962 paper on Deterministic Nonperiodic Flow [11]:

Consider a system governed by:

$$\frac{dX_i}{dt} = F_i(X_1, \dots X_M),\ i = 1, \dots, M \dots\dots\dots\dots\dots\dots (1)$$

All points possess a unique solution:

$$X_i = f_i\left(X_{1_0}, \dots X_{M_0}, t\right),\ i = 1, \dots, M \dots\dots\dots\dots\dots\dots (2)$$

And satisfy the condition:

$$f_i\left(X_{1_0}, \dots X_{M_0}, t_0\right) = X_{i0},\ i = 1, \dots, M \dots\dots\dots\dots\dots (3)$$

Which suggests a unique trajectory through each point of the system.

This forced dissipative system can be generalized by

$$\frac{dX_i}{dt} = \Sigma_{j,k} a_{ijk} X_j X_k - \Sigma_j b_{ij} X_j + c_i \dots\dots\dots\dots\dots\dots (4)$$

Lorenz allows several terms be constants, mandated by:

$Q = \frac{1}{2}\Sigma_i X_{i^2}, \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots . (5)$

When $e_1, \dots, e_M$ are roots of the equation, then:

$\Sigma_j\left(b_{ij} + b_{ji}\right)e_j = c_i \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots (6)$

Combining (6), (5), and (4) gives:

$\frac{dQ}{dt} = \Sigma_{i,j} b_{ij} e_i e_j - \Sigma_{i,j} b_{ij}(X_i - e_i)\left(X_j - e_j\right) \dots \dots (7)$

To solve (1) numerically,

Lorenz chooses initial time and time increment:

$X_{i,n} = X_i(t_0 + n\Delta t) \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots . . (8)$

And he uses auxiliary approximations:

$X_{i(n+1)} = X_{i,n} + F_i(P_n)\Delta t \dots \dots \dots \dots \dots \dots \dots \dots \dots (9)$

$X_{i(n+2)} = X_{i,(n+1)} + F_i\left(P_{(n+1)}\right)\Delta t \dots \dots \dots \dots \dots \dots (10)$

And uses the double-approximation procedure:

$X_{i,n+1} = X_{i,n} + \frac{1}{2}\left[F_i(P_n) + F_i(P_{(n+1)})\right]\Delta t \dots \dots \dots . (13)$

And rewrites it using (9) and (10):

$X_{i,n+1} = \frac{1}{2}(X_{i,n} + X_{i(n+2)}) \dots \dots \dots \dots \dots \dots \dots \dots \dots (14)$

This equation is successively evaluated to accomplish Lorenz's computations.

In the special case where all motion is parallel to the x-z plane, and there is no variation in the y-axis direction, Lorenz rewrites his governing equations as:

$\frac{\partial}{\partial t}\nabla^2\Psi = -\frac{\partial(\Psi,\nabla^2\psi)}{\partial(x,z)} + \nu\nabla^4\Psi + g\alpha\frac{\partial\theta}{\partial x} \dots \dots \dots \dots \dots (17)$

$\frac{\partial}{\partial t}\theta = -\frac{\partial(\Psi,\theta)}{\partial(x,z)} + \frac{\Delta T}{H}\frac{\partial\Psi}{\partial x} + \kappa\nabla^2\theta \dots \dots \dots \dots \dots \dots \dots (18)$

Where Ψ is a "stream function" for two dimensional Motion, θ is a temperature difference, and the constants $g, \alpha, \nu, and\ \kappa$ denote gravity, thermal expansion, viscosity, And thermal conductivity.

Lorenz then used Rayleigh's equation for the fields of motion:

$\Psi = \Psi_0 \sin(\pi a H^{-1}x)\sin(\pi H^{-1}z) \dots \dots \dots \dots \dots \dots \dots (19)$

$\theta = \theta_0 \cos(\pi a H^{-1}x)\sin(\pi H^{-1}z) \dots \dots \dots \dots \dots \dots \dots (20)$

and he notes that these equations require the Rayleigh number:

$R_a = g\alpha H^3 \Delta T \nu^{-1}\kappa^{-1} \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots (21)$

To exceed a critical value:

$R_c = \pi^4 a^{-2}(1 + a^2)^3 \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots (22)$

Lorenz builds on the work of Saltzman, who expanded $\Psi\ and\ \theta$ In a double Fourier series in x and z, substituted them into (17) And (18), swapped products of trigonometric functions for sums, Reduced the infinite system for a finite system by omitting sets And only used a specified finite set of functions of t.

Lorenz then truncated the series to include three terms:

$$a(1+a^2)^{-1}\kappa^{-1}\Psi = X\sqrt{2}\sin(\pi a H^{-1}x)\, ain(\pi H^{-1}z) \dots\dots\dots (23)$$

$$\pi R_c^{-1}R_a\Delta T^{-1}\theta = Y\sqrt{2}\cos(\pi a H^{-1}x)\sin(\pi H^{-1}z) - Zsin\,(2\pi H^{-1}z) \dots\dots\dots\dots (24)$$

Finally, substitutes (23) and (24) into (17) and (18) and omits the trig terms to get the Lorenz equations:

$$\dot{X} = -\sigma X + \sigma Y$$
$$\dot{Y} = -XZ + rX - Y$$
$$\dot{Z} = XY - bZ$$

Where dimensionless time is used: $\tau = \pi^2 H^{-2}(1+a^2)\kappa t$

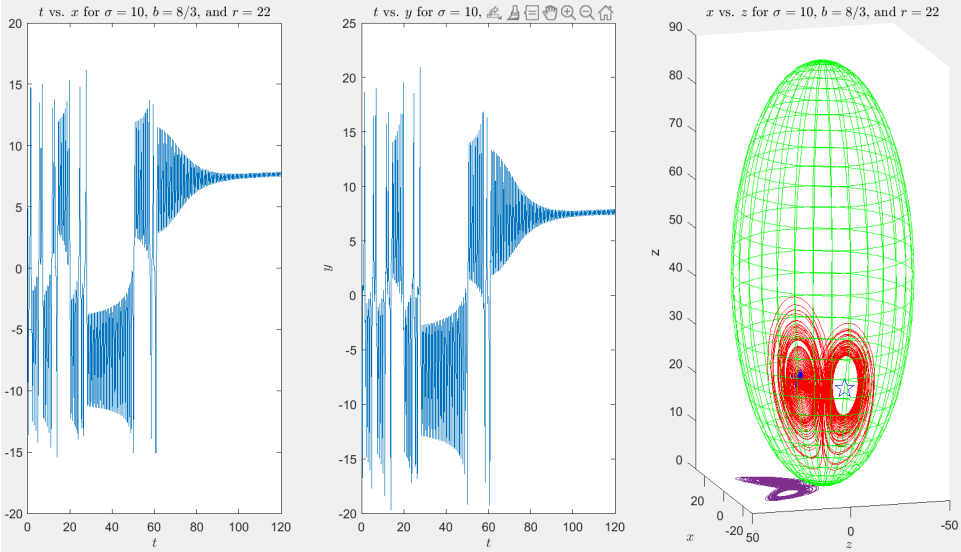
The Prandtl number is: $r = R_c^{-1}R_a$

And $b = 4(1+a^2)^{-1}$

In these equations, x is proportional to the intensity of the convective motion, y is proportional to the temperature difference between the ascending and descending currents, z is proportional to the distortion of the vertical temperature profile from linearity... Similar signs of x and y denoting that warm fluid is rising, and cold fluid is descending. A positive z value indicates that the strongest gradients occur near the boundaries.[11][15] Though these equations were designed to demonstrate the unpredictable nature of weather patterns[7], they have been applied to a wide range of applications including physics, biology, complex networks, economics[1], generators, chemical reactions, brushless DC motors, forward osmosis, lasers, electrical circuits, thermosiphon[4], cryptography and encryption[1-8].

The quality we desire from these equations for our application is the system's ability to generate pseudorandom numbers due to its chaotic behavior. The system does not always exhibit chaotic behavior [7]. Therefore, it is important to clarify the range in which we will be utilizing these equations. The Lorenz equations exhibit chaotic behavior after a Hopf bifurcation has occurred; where the systems limit cycles become unstable and the motion of trajectories are bounded and under the influence of the unstable limit cycles and some distant attractor [7].

For our project, we select r to be 22, σ to be 10, and b to be 8/3. From the lorenzmap.m and lorenz1.m code provided by Dr. Tasos Liakos in this course, we can plot the behavior of a Lorenz system with these parameters and initial condition [5, -7, 12]:



The graph of the behavior of the system seems to show a strange attractor being traced and aperiodic behavior for the first portion of its movement. As the movement of the trajectory continues, it settles down to move towards a stable fixed point. When referencing Strogatz figure 9.5.1:

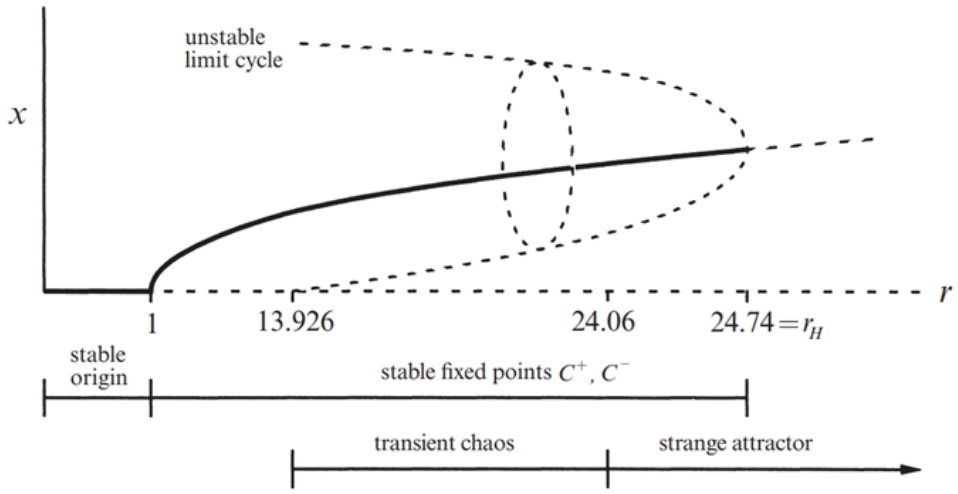


Figure 9.5.1

One may notice that our choice or parameters brings us to the "transient chaos" range of behavior for the Lorenz system. In this project we exemplify Strogatz words "Transient chaos shows that a deterministic system can be unpredictable, even if its final states are very simple. In particular, you don't need strange attractors to generate effectively random behavior."(Strogatz, 340) With this we can be confident that the numbers our algorithm generates will still be sufficiently random. For completeness, we also use our algorithm in the completely random range of parameters where we select r to be 30, σ to be 10, and b to be 8/3.

Methods:

This project demonstrates that a simple image encryption algorithm can be accomplished using the Lorenz equation to generate pseudorandom numbers using a 6-digit input sequence (described as the "(cryptographic) key" from now on) and associate these values with the original pixel RGB values. As mentioned in the introduction, the dataframe of pseudorandom ciphered values can be generated using the equations and the key. The first step to encrypting (or decrypting) an NxM image is to generate at least N*M pseudorandom values. The 3 dimensions will cover each color RGB within each pixel. That is, X=R, Y=G, Z=B.

The method for generating a pseudorandom number (henceforth PRN) follows the work of V. Lynnyk in [12]. The results of x, y, and z value at each timestep are multiplied by 10^13 and converted to an IEEE754 floating point number. The leftmost 8 bits of this value are truncated, and the rightmost 8 bits contain the PRN. Conveniently, 8-bit numbers create values ranging from 0-255 which is also the range for each R, G, and B value.

Finally, the trick to associate these ciphered PRNs to the original pixel RGB is to apply the Boolean logic operation XOR. This operation is particularly useful because the inverse operation of XOR is also XOR, so encrypting and decrypting images is simply XOR-ing the unencrypted/decrypted pixel RGB to the generated cipher dataframe.

The following subsection describes the method succinctly. It should be noted that the "t-space" is the interval in which solving the Lorentz system is accomplished. In testing, this was 0 to 20 in 300*300=90,000 steps. However, further simplification can be utilized by setting the end time to the absolute value of the product of the Lorenz parameters (|sigma*r*b|) and the number of steps to the number of pixels (N*M). In this way, integration is dependent on the key and does not necessarily need to be specified in the key itself.

Encryption algorithm steps: Given initial conditions x0, y0, z0, Lorenz parameters sigma, r, b, and an NxM image to encrypt.

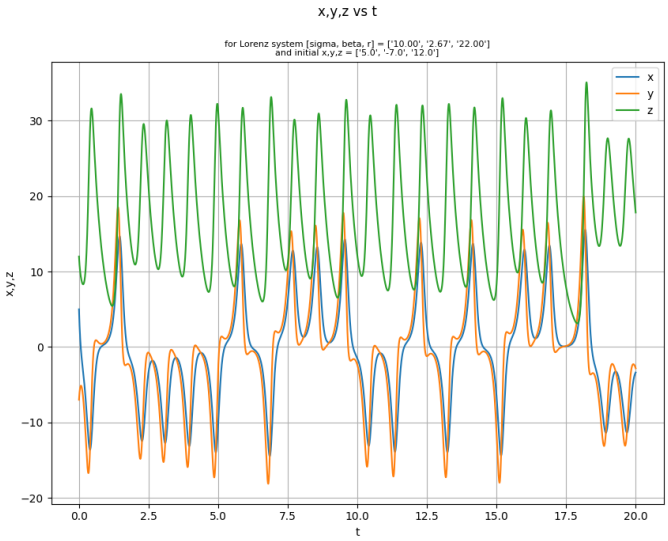
1. Produce PRNs of at least N*M iterations over the desired t-space [12]
    a. Create x, y, z values from Lorenz equations w/ initial conditions and parameters over desired t-space.
    b. Multiply each x, y, and z value by 10^13 and convert to IEEE754 floating point.
    c. The rightmost 8 bits to step 1b is the PRN. This is achieved using step1b mod 256. Result is stored as [Xc, Yc, Zc]
2. Pixel [R, G, B] XOR Step 1 result [Xc, Yc, Zc]
3. Resplice image pixels together. The order in which this is achieved is user preference.

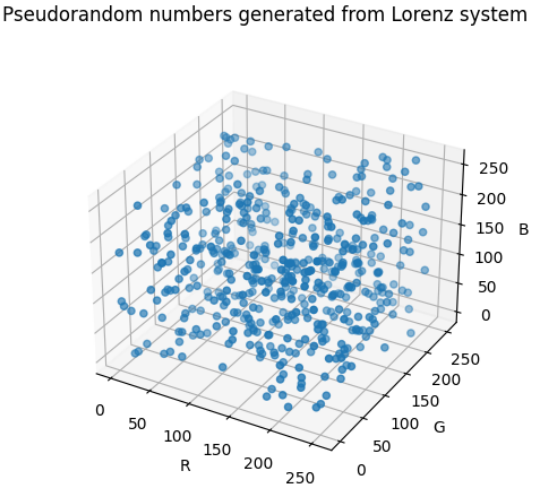
Decryption algorithm steps: Decryption is essentially the same as the encryption algorithm. Given Lorenz params and initial conditions and N x M encrypted image:

1. Reproduce PRNs, identical to step 1 from encryption method.
2. XOR step1 results with the encrypted pixel mage [R,G,B] values.
3. Resplice image pixels together in the same order as step 3 of encryption method

Due to the relative simplicity of the encryption/decryption algorithms, the required code is concise, and yet, from the nature of chaotic systems, the encryption is robust. Additionally, the algorithm is easy to understand and does not require much behind-the-scenes work to implement. The

below figures show additional viewpoints to the encryption keys. It is clear that the Lorenz system chosen can produce PRNs which allow for effective encryption to the original image pixels.

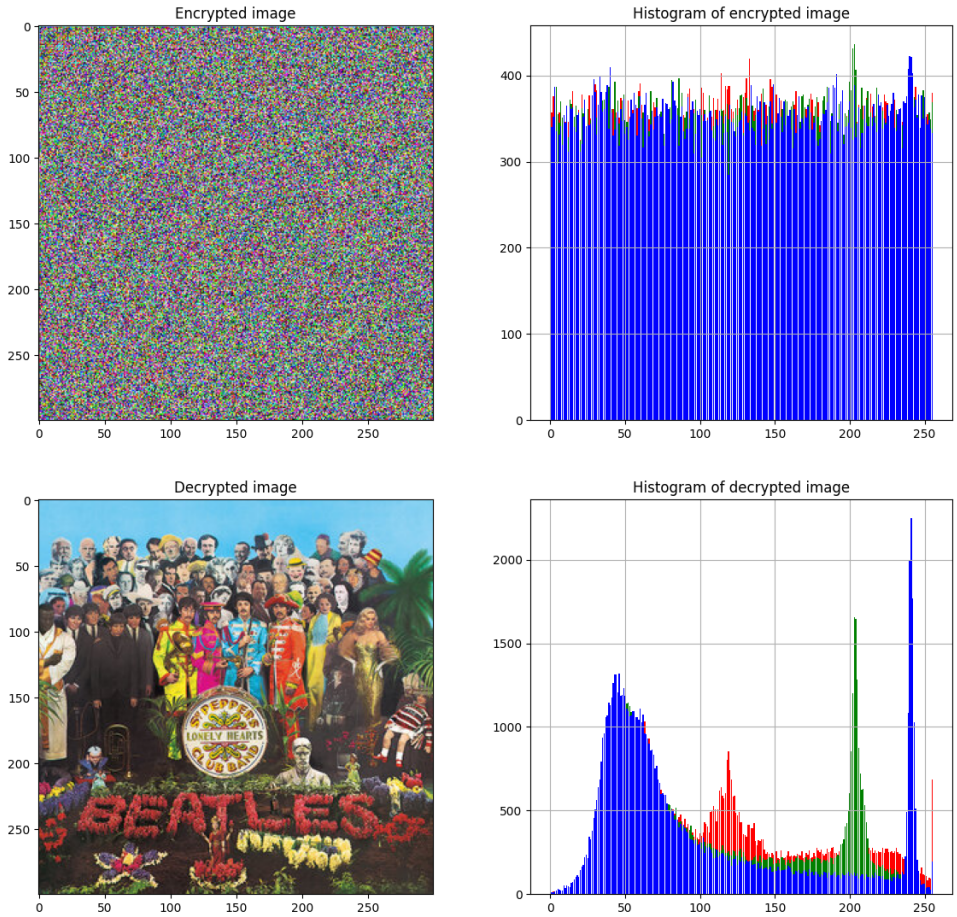




Results:

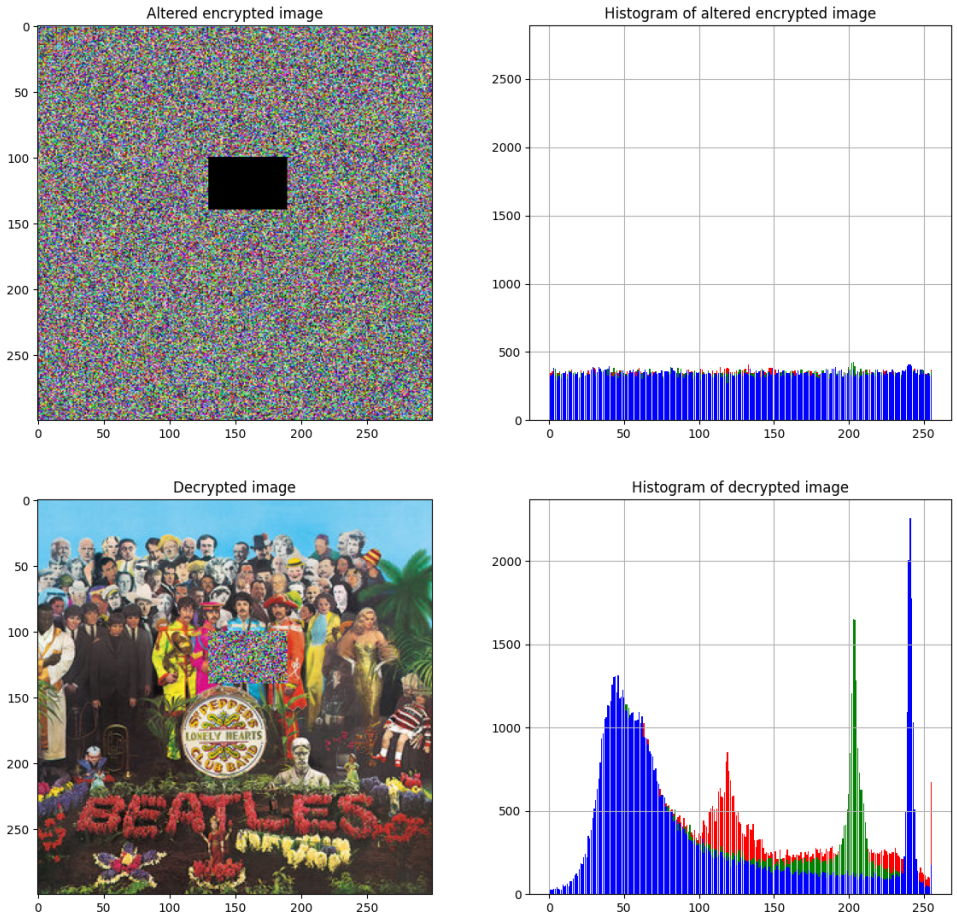
The image we decided to encrypt is The Beatles – Sgt. Pepper's Lonely Hearts Club Band (1967, designed by Peter Blake and Jann Haworth; cover art photograph by Michael Cooper) [13]. We decided to use this image because it contains a high level of color and shape differentiation. We assumed that this variance would be simple to encrypt because small changes could easily blur this image but difficult to decrypt because the image would need to be reconstructed exactly, with minimal error, to account for the complexity of the image.

The below figures show an example encryption and decryption for a 300x300 pixel image using the above method. Initial tests show that this algorithm effectively masks image data and sufficiently decrypts in a 1-to-1 manner, as expected. The decrypted image is an exact copy of the original, which is not shown here. It is clear that the encrypted image does not show signs of uniformity across all pixels, something that would be a by-product of non-randomness within the generated PRNs. Additionally, the table below shows benchmarking for this test case on a 2.3 GHz i5 processor Macbook Pro with 8 GB of memory. As this example is just proof of concept, no significant efficiency improvements have been made to the code or algorithm therefore, this is likely a worst-case scenario. Speed considerations are an area of future improvement.

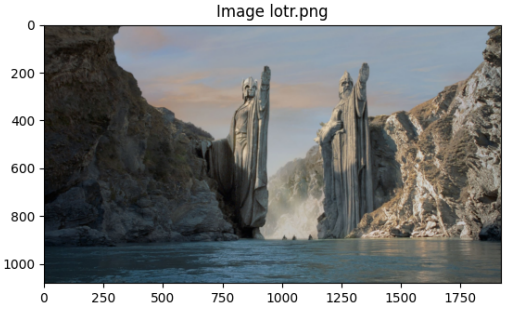


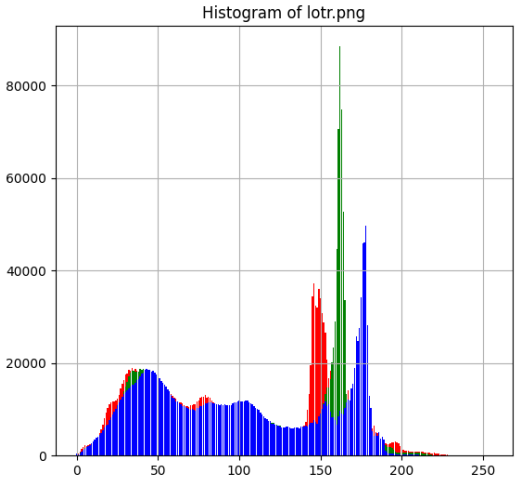
Function	Description	Elapsed Time (sec)
Load Lorenz Trajectories (N=300*300)	Solve Lorenz system w/ specific initial conditions and parameters and calculate cipher values (PRNs)	15.678
Read Image	Read saved .png file into numpy arrays (encrypted & decrypted)	0.713
Encrypt Image	Encrypt image based on PRNs and pixel RGB	13.845
Decrypt Image	Decrypt image based on PRNs and pixel RGB	14.021
Total		**44.256**

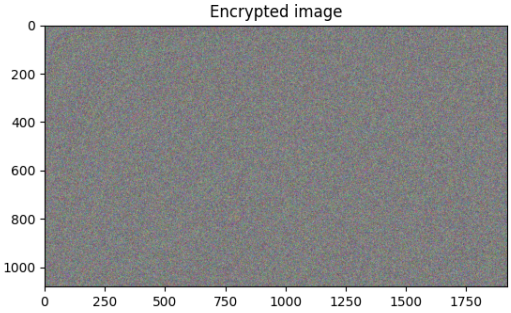
This obfuscation method produces PRNs on a per-color per-pixel basis, i.e. 1-to-1, and therefore would not preserve the integrity of the encrypted image when additional noise is involved, as shown in the figures below. This would be a desirable improvement to the algorithm: associating the PRN values generated from the Lorenz system to the actual pixel RGB values with the goal of creating a more cohesive image reconstruction algorithm. Hence, even if the noisy areas result in off-color areas in the decrypted image when obscured, the image itself may still be understandable.

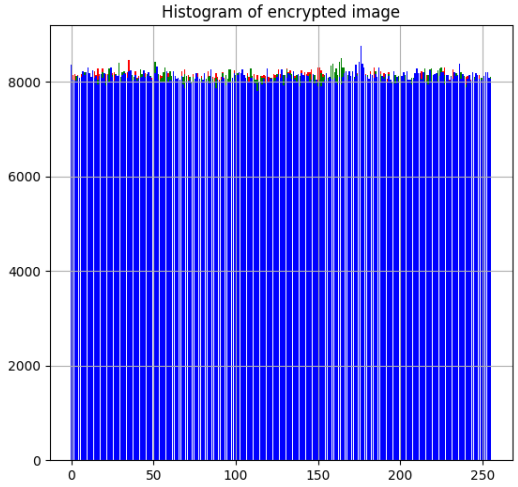


A 300x300 image does not have viable resolution in most cases so a 1920x1080 standard desktop background image was tested and benchmarked using the same workstation. The results below show expected behavior: perfect encryption/decryption abilities at the cost of significant increase in runtime as the number of pixels increases. Note that while processing time is upwards of 15 mins long, the image has approximately x20 more pixels to process. Regardless, 15 mins for standard image size encryption is too long for most applications. Some easy improvements could be to store cipher PRN data frames as a csv file, although this quickly breaks security protocols.









Function	Elapsed Time (sec)	Increase from 300x300 image processing time*
Load Lorenz Trajectories (N=1920*1080)	310.218	+1978.7%
Read Image	2.681	+376.0%
Encrypt Image	284.824	+2057.2%
Decrypt Image	291.122	+2076.3%
Total	**888.845**	**+2008.4%**

**Note 1920x1080 image has approx. x20 more pixels than 300x300 image.*

Conclusion:

We find the Lorenz equations to be an effective tool for accomplishing image encryption due to its ability to produce pseudorandom numbers. These pseudorandom numbers can be injected into a simple encryption algorithm to encrypt image data sufficiently albeit slowly. The strong chaotic behavior acts as a scrambler which adequately distorts the input image. The concise equations afford lower code complexity while the disorganized algorithm is likely to blame for most efficiency bottlenecks.

To increase security of our encryption methods, one could increase the complexity of several facets of the algorithm. For starters, several more layers of security would need to be added, such as encrypting the key itself or changing it every run. Additionally, utilizing asymmetric encryption into our model would boost encryption. Finally, employing end-to-end encryption can secure both sides of the imagery data. Beyond just adding more cryptographic aspects, one could apply published "improved" Lorenz systems which seek to deploy further security hardening efforts and push the initial Lorenz system towards identified provable random sequences [1].

Bibliography:

1. Zou, C., Zhang, Q., Wei, X., & Liu, C. (2020). Image encryption based on improved Lorenz system. *IEEE Access*, *8*, 75728-75740.
    a. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9072440
2. Kaur, M., & Kumar, V. J. E. L. (2018). Efficient image encryption method based on improved Lorenz chaotic system. *Electronics Letters*, *54*(9), 562-564.
    a. https://ietresearch.onlinelibrary.wiley.com/doi/epdf/10.1049/el.2017.4426
3. Huang, L., Shi, D., & Gao, J. (2016). The design and its application in secure communication and image encryption of a new Lorenz-like system with varying parameter. *Mathematical Problems in Engineering*, *2016*.
    a. https://downloads.hindawi.com/journals/mpe/2016/8973583.pdf
4. Arshad, U., Batool, S. I., & Amin, M. (2019). A novel image encryption scheme based on Walsh compressed quantum spinning chaotic Lorenz system. *International Journal of Theoretical Physics*, *58*(10), 3565-3588.
    a. https://www.researchgate.net/publication/335010304_A_Novel_Image_Encryption_Scheme_Based_on_Walsh_Compressed_Quantum_Spinning_Chaotic_Lorenz_System
5. Peng, X., & Zeng, Y. (2020). Image encryption application in a system for compounding self-excited and hidden attractors. *Chaos, Solitons & Fractals*, *139*, 110044.
    a. https://www.sciencedirect.com/science/article/pii/S0960077920304410
6. Narasimhan Aarthie and Rengarajan Amirtharajan, 2014. Image Encryption: An Information Security Perceptive. *Journal of Artificial Intelligence, 7: 123-135.*
    a. https://scialert.net/fulltext/?doi=jai.2014.123.135
7. Strogatz, S. H. (2018). *Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering*. CRC press.
8. Pandya, D., Ram, K., Thakkar, S., Madhekar, T., & Thakare, B. (2015). Brief History of Encryption. *International Journal of Computer Applications*, *131*(9), 28-31.
    a. https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.741.6752&rep=rep1&type=pdf
9. Lyon, R. F. (2006, February). A brief history of 'pixel'. In *Digital Photography II* (Vol. 6069, p. 606901). SPIE.
    a. http://www.dicklyon.com/tech/Photography/Pixel-SPIE06-Lyon.pdf
10. Smith, A. R. (1995). A Pixel Is Not A Little Square, A Pixel Is Not A Little Square, A Pixel Is Not A Little Square!. *Microsoft Computer Graphics, Technical Memo*, *6*.
    a. http://alvyray.com/Memos/CG/Microsoft/6_pixel.pdf
11. Lorenz, E. (1962). Deterministic Nonperiodic Flow. *Journal of Atmospheric Sciences.* Volume 20 Issue 2, pages 130-141.

12. Lynnyk, V., et al (2015). Pseudo random number generator based on the generalized Lorenz chaotic system, 4th IFAC Conference on Analysis and Control of Chaotic Systems CHAOS 2015. Volume 48, Issue 18, pages 257-261.
13. McGuinness, P. (2022, January 26). *The Beatles album covers explained*. uDiscover Music. Retrieved August 14, 2022, from https://www.udiscovermusic.com/stories/the-beatles-album-covers-explained/
14. Marsden, J. E., & McCracken, M. (2012). The Hopf bifurcation and its applications (Vol. 19). Springer Science & Business Media.