# Encryption/Decryption of Audio and Image Files Using Chaotic Maps and FPGA Implementation

Shahnawaz Mustafir, Mohan Yadav, and Yash Yadav
*Department of Physics, IIT Kanpur*
(Dated: November 11, 2024)

In this paper, we present a novel approach to encrypting and decrypting audio and image files using chaotic maps, initially implemented in Python and subsequently simulated on FPGA using Verilog. Chaotic maps, with their high sensitivity to initial conditions, are particularly suited to secure encryption, as even minor variations in the encryption key produce unpredictable changes in the output. Our approach integrates the Logistic map, Henon map, and Arnold Cat map for encrypting multimedia data, each chosen for its unique properties in enhancing security. The simulation results demonstrate the feasibility of chaotic encryption, showcasing the potential of chaotic maps for robust, real-time applications in secure multimedia communication.

## I. INTRODUCTION

With the growth of digital communication, safeguarding multimedia data like audio and image files is essential for preventing unauthorized access and ensuring data integrity. Traditional encryption methods, while effective, often face high computational demands and vulnerabilities. This project tackles these challenges using chaotic maps—deterministic models with unpredictable behavior—for secure, lightweight encryption. Starting with Python-based chaotic encryption and decryption algorithms, the project moved to an FPGA simulation in Verilog to evaluate hardware performance. FPGAs, with their parallel processing, high-speed, and reconfigurable capabilities, handle complex algorithms efficiently, demonstrating the potential of chaotic maps for real-time, secure multimedia encryption as a robust alternative for data security.

## II. INTRODUCTION TO CHAOTIC MAPS

Chaotic maps are mathematical functions that display chaotic behavior, where even a slight change in initial conditions leads to significantly different outcomes. In our approach, we explore three different chaotic maps: the Logistic map, Henon map, and Arnold Cat map, each selected for its unique contribution to secure multimedia encryption.

### A. Logistic Map

The Logistic map is defined as:

$$x_{n+1} = r x_n (1 - x_n), \tag{1}$$

where $r$ is a constant parameter and $x_n$ is the value at iteration $n$. For encryption, $3.57 < r < 4$ ensures chaotic behavior, making it useful for scrambling data. It is computationally efficient and easy to implement in software and hardware, making it ideal for real-time encryption of sensitive data.

### B. Henon Map

The Henon map is a two-dimensional chaotic map defined as:

$$x_{n+1} = 1 - a x_n^2 + y_n, \quad y_{n+1} = b x_n, \tag{2}$$

where $a = 1.4$ and $b = 0.3$ ensure chaotic behavior. The Henon map generates two-dimensional pseudorandom sequences, which make it particularly effective for image encryption, as the chaotic sequence makes images unintelligible without the correct decryption key.

### C. Arnold Cat Map

The Arnold Cat map applies a transformation to pixel coordinates, defined as:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \mod N, \tag{3}$$

where $N$ is the image size. Due to its periodic behavior, the Arnold Cat map allows the original image to reappear after a certain number of iterations, making it highly suitable for encryption and decryption, with the number of iterations serving as the key.

## III. WHY IS CHAOTIC ENCRYPTION PREFERABLE

Chaotic encryption offers several advantages over traditional encryption techniques. First, the inherent sensitivity of chaotic maps to initial conditions provides a high degree of security. Even a minor change in the initial conditions produces a vastly different output, making it extremely challenging for an attacker to retrieve the original data without the exact key values.

Additionally, chaotic encryption is computationally lightweight, requiring fewer resources than traditional methods like RSA and AES. While RSA relies on large prime numbers and AES on complex transformations, chaotic maps generate pseudorandom sequences with simpler calculations, making them more suitable for real-time applications where low-latency encryption and decryption are essential.

Furthermore, chaotic maps are particularly effective for multimedia data due to their structure-preserving properties. Maps like the Henon and Arnold Cat can scramble pixel or audio data into visually or audibly unrecognizable forms, while maintaining the ability to recover the original data through decryption. This characteristic makes chaotic maps an optimal choice for applications involving images and audio, where data integrity and speed are crucial.

## IV. PYTHON-BASED IMPLEMENTATION

We initially implemented the chaotic map algorithms in Python to evaluate the effectiveness of chaotic encryption. Audio and image files were digitized and segmented into arrays of numerical values. For audio encryption, we applied the Arnold Cat and Henon maps, while for images, we utilized all three maps—Logistic, Henon, and Arnold Cat—each enhancing security in unique ways.

The encryption process involved generating pseudorandom sequences with the Arnold Cat and Henon maps for XORing with audio data segments, making any minor deviation in initial conditions prevent successful decryption. For images, pixel values were scrambled using all three maps, resulting in visually unrecognizable encrypted images. This setup ensures the data is highly sensitive to initial values, providing a strong layer of security.

Decryption involved regenerating the pseudorandom sequences using the correct initial conditions, enabling us to reverse the encryption and recover the original data. Testing confirmed that the chaotic maps were highly sensitive to key values, demonstrating their effectiveness in securing multimedia data.

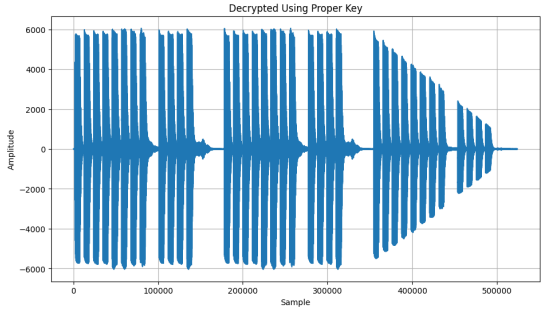
## V. FPGA SIMULATION USING VERILOG

In the FPGA simulation phase, we developed Verilog modules to implement chaotic encryption algorithms for audio and image files. The primary modules include the encryption modules, each generating pseudorandom sequences based on initial conditions. These sequences are essential for scrambling data, providing a high level of security. Each chaotic map was implemented as an independent module, enabling us to test their functionality separately before integrating them into a cohesive encryption system.

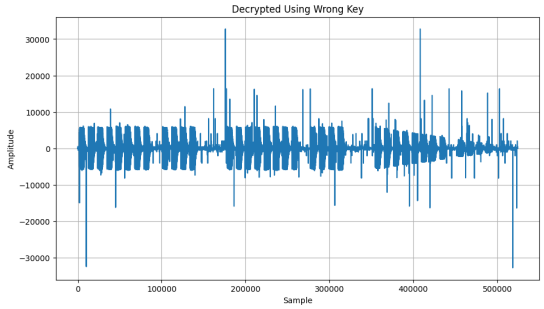
Additionally, a control unit was designed as a finite state machine (FSM) to manage data flow and synchronization across the chaotic map modules. This control unit segments the input data and ensures proper alignment during the encryption and decryption processes. Due to hardware limitations—specifically, a malfunctioning cable—we were unable to complete real-time FPGA testing, so the encryption and decryption were simulated on a PC. Future work will address these limitations to achieve real-time encryption performance on FPGA hardware.

## VI. KEY SENSITIVITY ANALYSIS

Chaotic encryption's primary strength is its sensitivity to initial conditions, which acts as the encryption key. Through experiments, we observed that even a slight change in initial conditions resulted in a completely different encrypted output. This property ensures that only the exact key can decrypt the data, offering strong protection against brute-force attacks.




(a) Decrypted with correct key



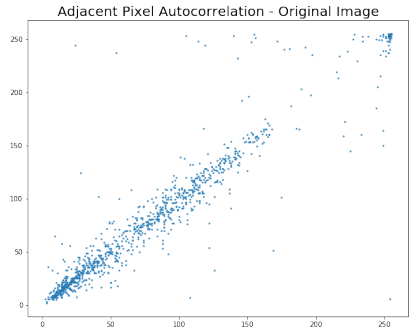

(b) Decrypted with incorrect key

FIG. 1. Effect of key sensitivity on decryption.

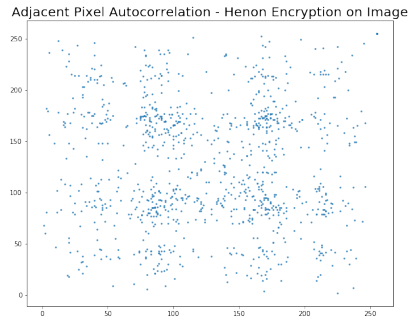
## VII. AUTO-CORRELATION ANALYSIS

Auto-correlation analysis is essential in assessing chaotic map-based systems' randomness and encryption strength. By measuring the similarity between data points at different shifts, auto-correlation helps identify any patterns within the encrypted data.

In the case of images, we perform auto-correlation on pixel values across both horizontal and vertical directions. For an effectively encrypted image, autocorrelation should approach zero for any non-zero shift, reflecting the loss of structure and predictability in pixel patterns. This disruption in image continuity confirms that the chaotic maps effectively obscure the visual information, making it resistant to statistical and brute-force attacks.



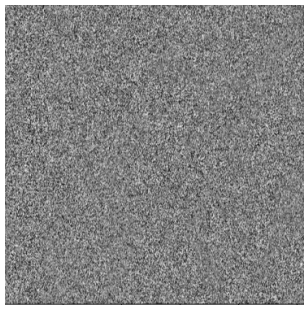

(a) Original Image



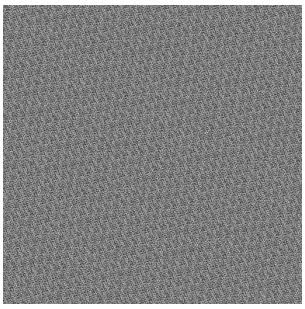

(b) Encrypted Image

FIG. 2. Auto-correlation plot for encrypted image, illustrating high randomness

(a) Original image



(b) Encrypted image using Henon map

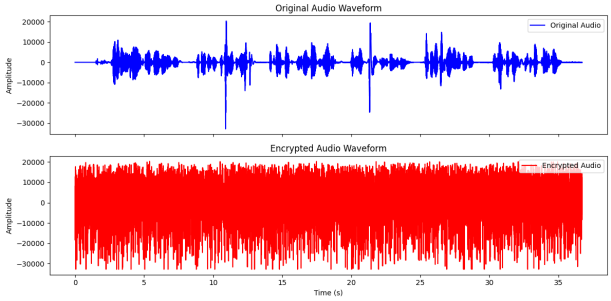


(c) Encrypted image using Arnold Cat map

FIG. 4. Comparison of the image before and after encryption using Henon and Arnold Cat maps.

## VIII. RESULTS AND DISCUSSION

The Python simulation demonstrated effective scrambling of both audio and image files, making them unintelligible without the correct decryption key. Figure 3 shows an example of an audio waveform before and after encryption. Once implemented in FPGA, this system is expected to achieve real-time encryption, ideal for practical use in secure communication systems. Chaotic maps provide a high level of security, as even a slight change in initial conditions results in drastically different outputs, making unauthorized decryption highly improbable.




(a) Original and encrypted audio waveform

FIG. 3. Comparison of audio before and after encryption.

## IX. LIMITATIONS AND FUTURE WORK

While the chaotic encryption approach proved effective in simulations, we encountered hardware limitations that prevented full FPGA testing. Specifically, a malfunctioning cable restricted our ability to verify the real-time performance of hardware. Future work will resolve these hardware issues and optimize the Verilog design for faster and more resource-efficient performance. Exploring other chaotic maps and hybrid encryption techniques may enhance security and adaptability across a broader range of multimedia applications.

## X. CONCLUSION

This paper presents a chaotic map-based approach for the encryption and decryption of audio and image files, implemented in Python and simulated on FPGA. We explored three different chaotic maps: the Logistic map, the Henon map, and the Arnold Cat map, each with unique properties suitable for encryption. Our simulations highlight the feasibility of using chaotic maps for secure communication. The FPGA design demonstrates the potential for real-time performance, making it suitable for practical applications in secure multimedia

systems. Future work will focus on full FPGA testing and exploring additional optimizations for better hardware resource management.

## XI. ACKNOWLEDGMENTS

We would like to express our gratitude to our course instructor, Prof. Rajeev Gupta, Upendra Sir, and the assigned TAs for their invaluable support and guidance in the completion of this project. We would also like to thank the Electronics Club, IIT Kanpur for providing us with the FPGA instrument used in this experiment.

## XII. REFERENCES


- S. H. Strogatz, *Nonlinear Dynamics and Chaos*, Westview Press, 2000.
- H. Yassin, A. Mohamed, A. Abdel-Gawad, M. Tolba, H. Saleh, A. Madian, and A. Radwan, "Speech Encryption on FPGA Using a Chaotic Generator and S-Box Table," in *Proceedings of the 2019 IEEE African Conference on Transformation in Engineering and Technology (ACTEA)*, 2019, pp. 1-4. doi: 10.1109/ACTEA.2019.8851086.
- E. Yavuz, R. Yazıcı, M. C. Kasapbaşı, and E. Yamaç, "A chaos-based image encryption algorithm with simple logical functions," *Journal of Computer Science*, Istanbul Commerce University, Istanbul, Turkey, 2016.
- A. S and M. Krishnan, "Enhanced Audio Encryption using 2-D Zaslavsky Chaotic Map," *2022 International Conference on Computer Communication and Informatics (ICCCI)*, Coimbatore, India, 2022, pp. 1-4. doi: 10.1109/ICCCI54379.2022.9740761.