# Time-Frequency-Phase Speech Encryption using a Hybrid Logistic-Tent Chaotic System

Francisco Trianto
*Informatics Engineering*
*School of Electrical Engineering and Informatics*
*Bandung Institute of Technology*
Bandung, Indonesia
franciscotrianto@gmail.com
13522091@std.stei.itb.ac.id

***Abstract*—This paper proposes a time-frequency-domain speech encryption scheme based on a Hybrid Logistic-Tent chaotic system. The method transforms speech into a Short-Time Fourier Transform (STFT) spectrogram, then applies frequency scrambling, XOR-based diffusion on quantized magnitude and phase, and time-block permutation based on a hybrid chaotic keystream. Public metadata stores only non-secret reconstruction parameters, while security depends on a four-parameter secret key that exhibits high sensitivity inside the chaotic system. Experiments were performed on five samples from the LJ Speech Dataset using a Python implementation. The results show that the encrypted waveforms and spectrograms become noise-like with increased entropy and expanded dynamic range, whereas decrypted signals closely match the original audio both visually and in basic statistics (mean, standard deviation, RMS). These results indicate that the proposed scheme successfully provides reversible, key-sensitive, and statistically robust protection for speech, suitable for speech communication.**

***Index Terms*—Speech encryption, Hybrid chaotic map, Time-frequency domain, STFT, Audio scrambling**

## I. Introduction

Speech is one of the most natural and effective ways for humans to communicate. Modern applications like Voice over IP (VoIP), mobile telephony, and remote conferencing rely heavily on speech. As these systems transmit data over open wireless and packet-switched networks, they become vulnerable to eavesdropping and unauthorized recording by an adversary. Without adequate protection, sensitive conversations can be easily intercepted, raising serious privacy concerns [1].

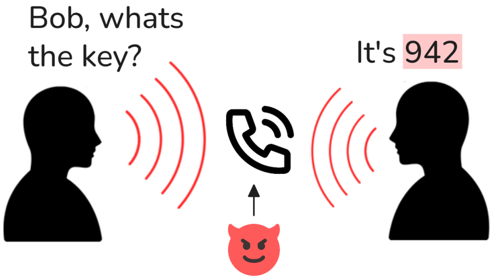


Fig. 1. Modern Speech Communication System.

Standard algorithms like the Advanced Encryption Standard (AES) are commonly used for digital data. But, while AES provides strong security bytes, it's not always optimal for speech communication. Speech has unique properties, such as high redundancy and a need for low latency. Standard block ciphers treat audio as a generic bitstream, which can introduce processing delays. Furthermore, they are highly sensitive to transmission errors as a single corrupted bit can render an entire block of audio unplayable, leading to significant signal loss [1].

In response to these challenges, researchers tried to use chaos-based cryptography. Chaotic systems are attractive because they use simple mathematical models to generate complex, pseudo-random behavior. They are highly sensitive to initial conditions, meaning a tiny change in the secret key produces a completely different output [5], [6]. This makes them efficient and effective for multimedia encryption [11].

However, many existing chaos-based schemes have limitations. Some schemes rely on simple, single-dimensional chaotic maps, which have small key spaces and vulnerable to statistical attacks. Some schemes operate only in the time domain by scrambling the raw waveform. Time-domain scrambling often fails to fully mask the intelligibility of the speech, leaving behind residual spectral patterns that attackers can exploit.

To address these issues, this paper proposes a robust speech encryption scheme that operates in the time-frequency-phase domain. We utilize a Hybrid Logistic-Tent chaotic system, which combines two distinct chaotic maps to significantly expand the key space and improve randomness [7]–[9]. By converting speech into a spectrogram (time-frequency representation) before encryption, our method effectively destroys both the waveform structure and the spectral characteristics (formants) of the audio. This approach ensures that the encrypted signal appears as wide-band noise while remaining fully reversible for the authorized receiver.

## II. Theoretical Basis

### A. Secure Speech Communication

Speech is the exchange of information through audio signals via telephone networks, Voice over Internet Protocol (VoIP), or wireless interfaces. In modern communication, human voice gets converted into digital signals and transmitted over public communication channels.However, attackers can intercept these signals easily to eavesdrop and intercept. This process creates the need for an end-to-end secure speech communication to ensure the privacy and confidentiality of information [1].

Modern communication systems assume the existence of an adversary attempting to intercept communication channels. To mitigate this, the most common defense is encryption, specifically the encryption of information to ensure confidentiality.

Standard encryption standards like AES can be applied to speech, but they are not optimal. The primary issue is efficiency since converting complex speech signals into a generic binary format for block encryption can (1) introduce significant latency, (2) increase bandwidth usage, and (3) result in complete signal loss if a single bit is corrupted during transmission (avalanche effect) [1].

Thus, specialized techniques have been developed for speech security, some of which are speech coding, steganography, and speech scrambling. This research focuses specifically on speech scrambling.

### *B. Digital Speech Representation and Intelligibility*

Audio itself is actually a form of complex sound waves. Speech is just an audio with certain characteristics that humans use to communicate. Intelligibility refers to the degree of which speech can be understood by human listeners, determined primarily by the clarity of phonetic content. Human hearing perceives speech primarily in the range of 80 Hz to 8 kHz, with most intelligible content concentrated between 300 Hz and 3400 Hz known as the telephone bandwidth [2].

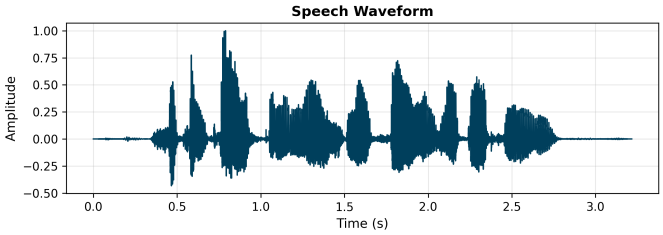


Fig. 2. Example of a speech waveform segment showing amplitude variations over time corresponding to spoken phonemes and the natural speech envelope.

Figure 2 illustrates a speech waveform with amplitude modulation, demonstrating the time-varying nature of vocal production. The frequency content of this signal evolves over time as different phonemes are produced. Figure 3 illustrates the distribution of speech energy across the frequency domain, highlighting the telephone bandwidth and the positions of the first two formants which carry critical vowel information.

The human auditory system processes speech through temporal and spectral analysis. Frequencies below 80 Hz contribute mainly to speaker recognition (voice presence) rather than explicit content comprehension. Frequencies above 4 kHz enhance the naturalness and brightness of speech but contribute less to intelligibility, making them suitable candidates for more aggressive scrambling in secure communication schemes.

Speech signals exhibit a quasi-periodic structure with a fundamental frequency (pitch) typically ranging from 85 Hz for adult males to 255 Hz for adult females [2]. Vowels show relatively stable formant structures, while consonants contain rapid transient energy concentrated in higher frequencies. Figure 4 summarizes typical ranges for pitch and the first three formants, which play a central role in the human perception of phonetic content [2].

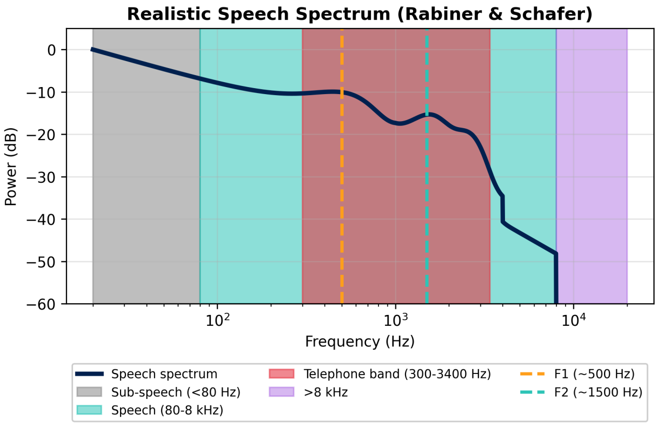


Fig. 3. Illustration of average speech power spectrum showing the full speech bandwidth (80–8000 Hz), the telephone band (300–3400 Hz), and example locations of the first two formants (F1 and F2) [2].

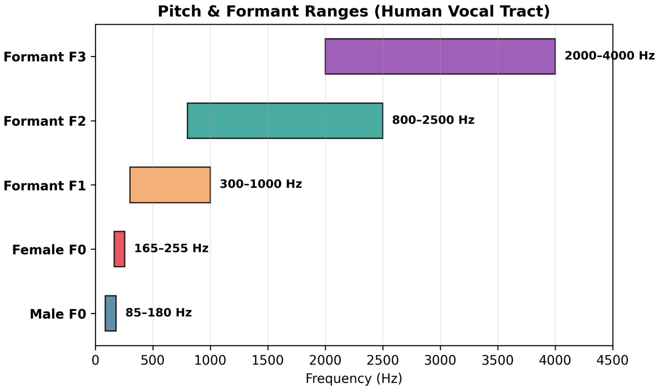


Fig. 4. Typical fundamental frequency ranges for male and female speakers together with approximate bands of the first three formants (F1–F3) [2].

### *C. Speech Scrambling*

Speech scrambling is a technique for obfuscating speech signals to prevent unauthorized access [11]. The objective is to transform the speech signal into an unintelligible noise-like format. The original speech can only be reconstructed by a receiver possessing the correct reverse algorithm and secret key. Conceptually, it combines elements of encryption with signal processing.

Scrambling can be applied across various dimensions of an audio signal, including time, frequency, amplitude, and phase [3]. Traditional methods often operate in the physical dimension (time segments) or the spectral dimension (frequency bands). This research employs multi-dimensional scrambling, manipulating the time, frequency, and phase domains simultaneously to achieve higher security.

### *D. Time-Frequency Analysis of Speech*

Speech signals are non-stationary, meaning their frequency content changes rapidly over time. The classical Fourier Transform analyzes stationary signals but loses temporal localization. To capture dynamic changes accurately, time-frequency representations are required.

The Short-Time Fourier Transform (STFT) addresses this limitation by dividing the speech signal into short overlapping segments called frames [3]. Each frame is multiplied by a window function (such as a Hamming or Hann window) to reduce spectral leakage before applying the Discrete Fourier Transform. This windowing operation localizes the spectral analysis to a specific time interval, preserving temporal information [3].

The STFT produces a spectrogram, a two-dimensional representation where one axis represents time and the other represents frequency [2]. Each cell in the spectrogram contains both magnitude (intensity) and phase (angle) information. The magnitude spectrogram reveals the energy distribution of frequencies over time, while the phase spectrogram captures the timing relationships critical for signal reconstruction [2].

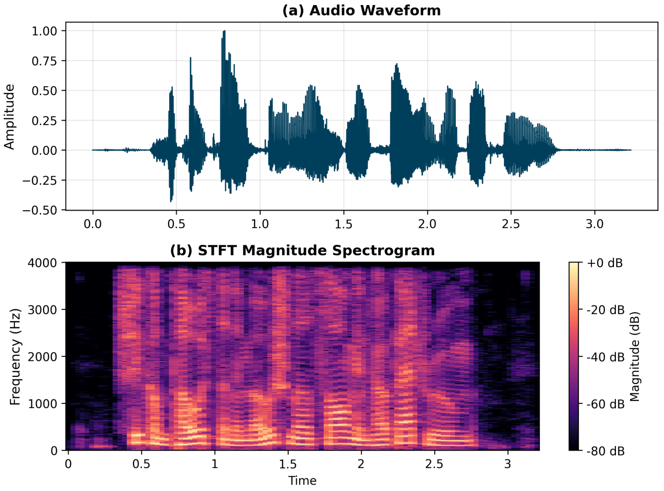


Fig. 5. Transformation of time-domain speech signal (a) into time-frequency spectrogram via STFT (b). The spectrogram reveals the harmonic structure hidden in the waveform.

The STFT parameters dictate the time-frequency resolution tradeoff [3]. Longer windows (40–50 ms) provide finer frequency resolution but coarser time localization, whereas shorter windows (20–30 ms) offer better time resolution. Overlap between consecutive frames (typically 50–75%) ensures smooth transitions and prevents information loss at frame boundaries.

Crucially, the STFT is reversible. The original signal can be recovered using the Inverse STFT (iSTFT) with overlap-add reconstruction [3]. This property is essential for encryption schemes, ensuring that the scrambling process does not permanently destroy the signal quality.

### *E. Cryptography*

Cryptography is the science of protecting information by transforming it into a format readable only through a specific decryption process. The primary objectives are to ensure the confidentiality, integrity, authenticity of data as well as non-repudiation [12].

Cryptography encompasses several fundamental concepts: encryption, decryption, hashing, and cryptographic keys. In the context of real-time speech communication, efficiency and speed are paramount; therefore, this study focuses on asymmetric-key cryptography.

*1) Symmetric-Key Encryption and Stream Ciphers:* Symmetric-key cryptography uses a single secret key for both encryption and decryption processes. This class of algorithms is generally computationally more efficient than asymmetric systems, making it suitable for processing large streams of multimedia data.

Within symmetric cryptography, algorithms are categorized into block ciphers (e.g., AES) and stream ciphers [13]. Stream ciphers encrypt plaintext digits one at a time by combining them with a corresponding digit from a keystream. This mechanism is mathematically expressed as:

$$C_i = P_i \oplus K_i$$

where $P$ is the plaintext, $K$ is the keystream, and $\oplus$ denotes the Exclusive-OR (XOR) operation. In the proposed method, a chaotic system functions as a pseudo-random number generator (PRNG) to produce this keystream $K$.

### *F. Chaotic Cryptography*

Chaotic cryptography is based on the intrinsic properties of nonlinear dynamical systems, especially their sensitivity to initial conditions. Unlike classical cryptographic algorithms that rely on number-theoretic problems, chaotic encryption is based on complex dynamics systems where simple deterministic equations produce pseudo-random trajectories that are computationally irreducible [5].

A chaotic cryptosystem typically employs two fundamental operations: confusion and diffusion to ensure security.

*1) Confusion (Permutation):* Confusion obscures the relationship between the plaintext and the ciphertext by permuting data positions. In speech encryption, this is achieved by reordering time frames or frequency bins using index sequences derived from chaotic maps [6]. If a chaotic state $x_n$ generates a permutation vector $P$, the original data $D$ is transformed such that $D'_i = D_{P[i]}$. This destroys structural patterns (like formants) while preserving total energy.

*2) Diffusion (Substitution):* Diffusion dissipates the statistical structure of the plaintext across the entire ciphertext. In chaotic systems, this is typically implemented via bitwise XOR operations or modular arithmetic between the data and the chaotic keystream [9]. A single-bit change in the plaintext or key produces an avalanche effect throughout the ciphertext. Diffusion complements confusion by ensuring that local changes propagate globally.

### G. *Chaotic Map Models*

The security of a chaotic cryptosystem depends on the quality of the underlying one-dimensional map. This work employs a hybrid logistic-tent combination to maximize the key space, randomness, and resistance to cryptanalysis.

*1) Logistic Map:* The logistic map is a classic model of population growth described by:

$$x_{n+1} = rx_n(1 - x_n), \quad x_n \in (0, 1)$$

For $3.57 < r \leq 4.0$, the system exhibits chaos with aperiodic, ergodic trajectories [7]. However, standard logistic maps suffer from non-uniform invariant density and periodic ”windows” within the chaotic regime.

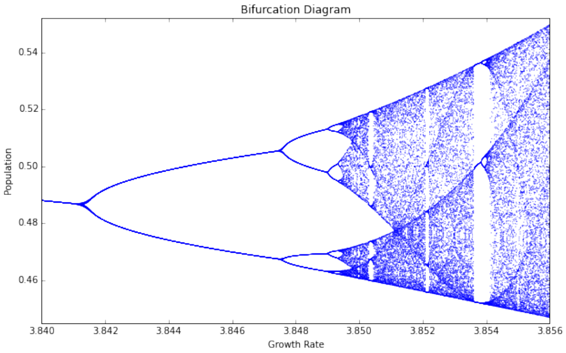


Fig. 6. Bifurcation diagram of the logistic map showing the period-doubling route to chaos.

*2) Tent Map:* The tent map provides a uniform distribution via a piecewise linear function:

$$y_{n+1} = \begin{cases} y_n/s & \text{if } y_n < s \\ (1 - y_n)/(1 - s) & \text{if } y_n \geq s \end{cases}$$

where $s \in (0, 1)$ controls the slope [8]. Unlike the logistic map, tent maps achieve a uniform invariant density and robust chaos across wide parameter ranges.

*3) Hybrid Logistic-Tent System:* Hybrid systems combine multiple maps to overcome individual limitations. The proposed hybrid map is defined as:

$$z_n = (x_n + y_n) \bmod 1.0$$

where $x_n$ follows logistic dynamics and $y_n$ follows tent dynamics [9]. This combination significantly expands the key space and improves the statistical randomness of the generated keystream.

## III. Proposed Method

This section outlines the proposed speech encryption scheme based on Time-Frequency scrambling using a Hybrid Logistic-Tent chaotic map. The overall architecture of the system is illustrated in Fig. 8. The scheme operates in the time-frequency domain by manipulating the spectrogram of the speech signal to destroy both intelligible content and statistical properties.

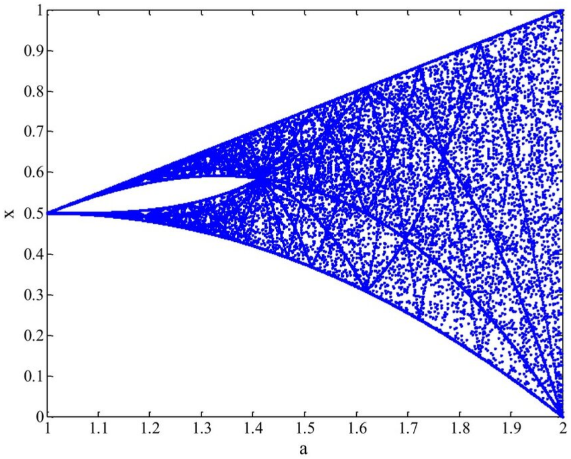


Fig. 7. Bifurcation diagram of the Tent map showing chaotic behavior.

### A. *Key Generation and Chaos Engine*

The security of the proposed method relies on a symmetric key structure consisting of four double-precision floating-point values: $K = (r, x_0, \mu, y_0)$. These parameters initialize the Hybrid Chaos Engine described in Section II.

The engine generates a pseudo-random chaotic sequence $Z = \{z_1, z_2, \ldots, z_N\}$ by combining the trajectories of the Logistic map (parameter $r$, initial state $x_0$) and the Tent map (parameter $\mu$, initial state $y_0$). This sequence $Z$ acts as the source of randomness for all subsequent scrambling and masking operations. Due to the sensitivity of chaotic maps, a minute change in any key parameter ($< 10^{-14}$) results in a completely different sequence $Z$, ensuring high key sensitivity.

### B. *Encryption Process*

The encryption process transforms the original intelligible speech into noise-like audio. The procedure consists of five main stages:

*1) STFT Transformation:* The input speech signal $S(t)$ is first converted from the time domain to the time-frequency domain using the Short-Time Fourier Transform (STFT). The signal is divided into overlapping frames, and a Fast Fourier Transform (FFT) is applied to each frame. This produces a complex spectrogram matrix containing Magnitude ($M$) and Phase ($P$) information.

$$X(f, t) = \text{STFT}\{S(t)\} \rightarrow M, P$$

*2) Frequency Scrambling (Confusion):* To destroy the spectral structure of the speech (e.g., formants and pitch), the system applies Frequency Scrambling. For each time frame $t$, the frequency bins are reordered using a permutation vector derived from the chaotic sequence $Z$.

- A segment of $Z$ is sorted to generate permutation indices.
- The frequency rows in the spectrogram are shuffled vertically according to these indices.
- This operation disperses the energy of dominant frequencies, making the audio sound unnatural.

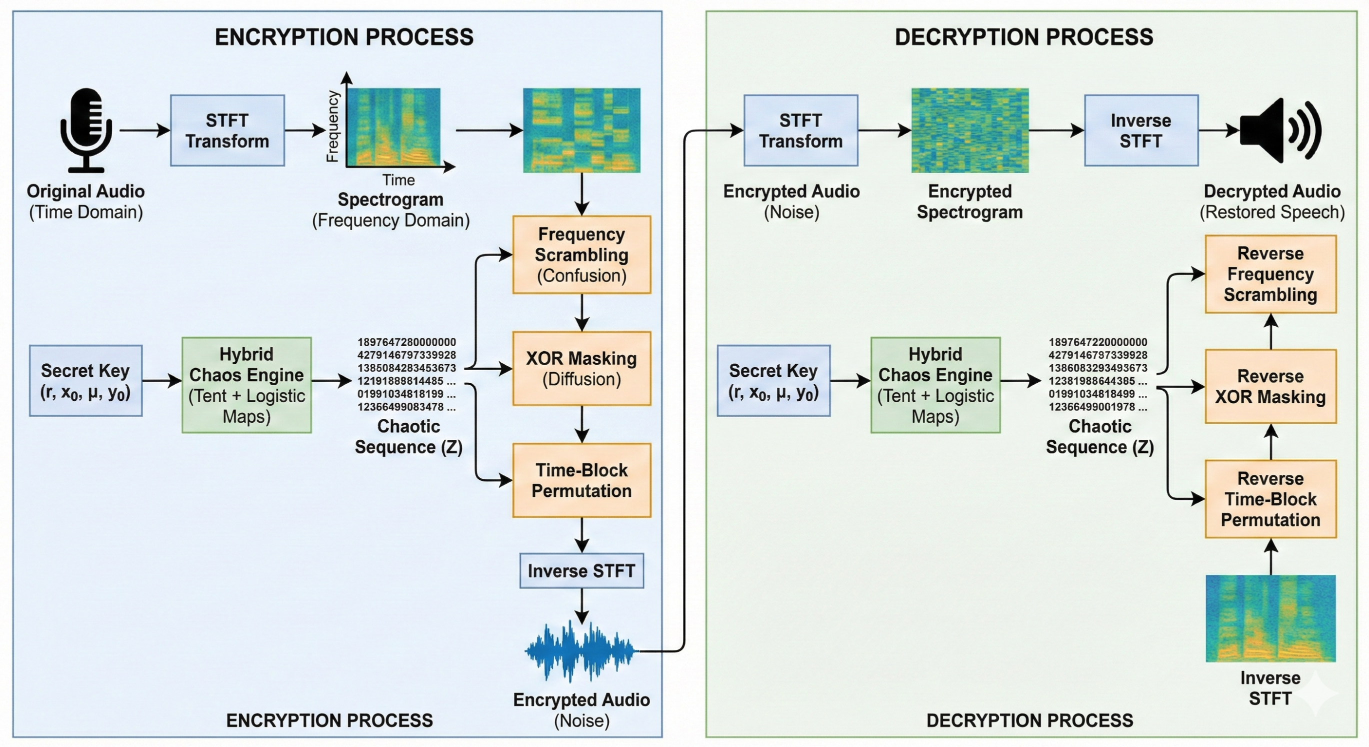


Fig. 8. The proposed speech encryption and decryption scheme using Hybrid Logistic-Tent Chaos.

*3) XOR Masking (Diffusion):* After frequency scrambling, a diffusion layer is applied to hide the statistical properties of the spectrogram. The chaotic sequence $Z$ is quantized into an integer mask matrix of the same size as the spectrogram.

- A bitwise XOR operation is performed between the scrambled spectrogram values and the chaotic mask.
- This ensures that a small change in the original plain-speech results in a widespread change in the encrypted output, providing resistance against statistical attacks.

*4) Time-Block Permutation:* To destroy the temporal characteristics (e.g., rhythm and word boundaries), the spectrogram is divided into blocks of columns (time frames). The order of columns within each block is shuffled horizontally based on the chaotic sequence $Z$. This operation disrupts the continuity of speech over time.

*5) Inverse STFT:* Finally, the fully scrambled and masked spectrogram is converted back to the time domain using the Inverse Short-Time Fourier Transform (ISTFT). The result is the encrypted audio signal $E(t)$, which is unintelligible noise.

### C. Decryption Process

The decryption process is the inverse of the encryption procedure. It requires the exact same Secret Key $K = (r, x_0, \mu, y_0)$ to successfully recover the original speech.

*1) Chaos Synchronization:* The receiver initializes the Hybrid Chaos Engine with the shared secret key. Because the system is deterministic, the engine regenerates the exact same chaotic sequence $Z$ used during encryption. This eliminates the need to transmit the chaotic sequence, reducing bandwidth overhead.

*2) Inverse Transformations:* The encrypted audio $E(t)$ is transformed back to the spectrogram domain via STFT. The decryption then proceeds in the reverse order of encryption (Last-In, First-Out):

1. **Reverse Time-Block Permutation:** Using the generated sequence $Z$, the receiver calculates the inverse permutation indices and restores the correct temporal order of the time frames.
2. **Reverse XOR Masking:** The same chaotic mask is generated from $Z$. Since $A \oplus B \oplus B = A$, applying the XOR operation again removes the mask and reveals the underlying scrambled spectrogram values.
3. **Reverse Frequency Scrambling:** The frequency bins are un-shuffled using the inverse of the frequency permutation vectors derived from $Z$, restoring the original spectral structure (formants and pitch).

*3) Signal Reconstruction:* Once the spectrogram is fully restored to its original state, the Inverse STFT is applied to reconstruct the decrypted speech signal $S'(t)$. If the key is correct, $S'(t)$ will be perceptually identical to the original speech $S(t)$.

## IV. Experimental Results

### A. Implementation and Setup

The proposed encryption scheme was implemented in Python using the `numpy`, `scipy`, `soundfile`, and `librosa` libraries for chaotic sequence generation and audio signal processing. The graphical user interface (GUI) was developed with PySide6 to allow users to load WAV files, configure the chaotic secret key, and visualize encryption and decryption outputs in real time. The complete implementation and reproducible scripts are available in an open-source repository: https://github.com/NoHaitch/Speech-Encryption-using-Hybrid-Logistic-Tent.

To evaluate the method, five speech samples were selected from the LJ Speech Dataset obtained trhough Kaggle (LJ001-0005.wav, LJ001-0016.wav, LJ001-0030.wav, LJ001-0044.wav, LJ001-0060.wav). Detailed plots of all experiments, including waveform and spectrogram comparisons, are provided in Appendix A.

### *B. Performance of Encryption and Decryption*

The experimental results show that the proposed Hybrid Logistic-Tent chaotic map can successfully transform intelligible speech into noise-like encrypted audio and then reconstruct the original signal when the correct key and metadata are provided.

In the **encryption** stage, the time-domain waveforms of all five samples become dense, noise-like signals with no visible pauses or phonetic structure. The corresponding spectrograms lose clear formant bands and harmonic trajectories, and energy is spread across frequency and time, indicating strong confusion and diffusion.

In the **decryption** stage, using the same secret key and the stored metadata (STFT parameters, quantization ranges, and block size), the inverse scrambling and masking operations restore the original spectrogram structure. The decrypted waveforms visually overlap with the original signals, confirming that the scheme is reversible and preserves intelligibility.

### *C. Audio Statistics of Original, Encrypted, and Decrypted Signals*

To quantify the transformations, several audio statistics were computed for each file and averaged across the five tests:

- **Mean amplitude** remains close to zero for all signals (original, encrypted, decrypted), indicating that no significant DC bias is introduced by the algorithm.
- **Standard deviation** approximately doubles after encryption (from about 0.09 to about 0.18), showing increased energy dispersion and higher entropy in the encrypted waveforms.
- **Dynamic range** expands from roughly $[-0.58, 0.69]$ in the original signals to around $[-0.80, 0.95]$ in the encrypted signals, which masks the original amplitude envelope.
- **Decrypted statistics** (mean, standard deviation, RMS, and duration) closely match the original values for all five tests, confirming that no significant distortion is introduced by the scrambling and inverse-STFT reconstruction.

These numerical results are consistent with the visual comparisons in Appendix A, where encrypted signals appear as wide-band noise and decrypted signals recover the original speech structure.

### *D. Strengths of the Proposed Method*

The experiments highlight several strengths of the proposed scheme:

- **Large key space and high sensitivity:** The secret key consists of four real-valued parameters $(r, x_0, \mu, y_0)$, each represented in double precision. Small changes (e.g., $10^{-14}$) in any parameter produce a completely different chaotic sequence, leading to a key space far beyond practical brute-force capabilities.
- **Strong confusion and diffusion:** Frequency scrambling, time-block permutation, and XOR masking jointly destroy both temporal and spectral patterns. Encrypted spectrograms lack visible formants or periodic structure, which makes statistical or spectral attacks difficult.
- **Exact reversibility:** When the correct key and metadata are available, the inverse process recovers the original signal with very small numerical error, making the scheme suitable for secure storage or transmission of high-quality speech.

### *E. Limitations*

The current design also has several limitations observed from the experimental results:

- **Metadata dependency:** Successful decryption requires access to the metadata file that stores non-secret parameters such as STFT configuration, quantization ranges, and block size. If this file is lost or corrupted, the encrypted audio cannot be reconstructed even with the correct key. This is a common issue in many encryption schemes and can be asummed that both side of communication already has an agreement on the metadata content.
- **Computational cost:** The algorithm involves STFT/ISTFT and multiple scrambling steps. For the short clips tested, encryption and decryption times are on the order of tens to hundreds of milliseconds, which is acceptable for offline processing but may be challenging for strict real-time or low-latency scenarios without optimization.
- **Slight data expansion:** Block-based processing and normalization can slightly change the effective length and dynamic range of the encrypted audio. While the effect is small, it should be considered when integrating the scheme into bandwidth-constrained systems.

## V. Conclusion

This paper proposed a robust speech encryption scheme that operates in the time-frequency domain using a Hybrid Logistic-Tent chaotic system combined with frequency scrambling, XOR-based diffusion, and time-block permutation. The method transforms intelligible speech into noise-like signals in both time and spectral representations while preserving perfect reversibility through inverse operations and stored public metadata.

Experimental results on five samples from the LJ Speech Dataset show that the encrypted audio exhibits high entropy, expanded dynamic range, and loss of visible formant structure, whereas the decrypted signals closely match the original waveforms and spectrograms. Audio statistics such as mean amplitude, standard deviation, and RMS confirm that the scheme effectively randomizes the ciphertext while maintaining reconstruction quality. Overall, the proposed approach provides a practical, key-sensitive, and statistically robust framework for secure speech communication in offline or semi-real-time scenarios.

## Appendix

- Github Repository: https://github.com/NoHaitch/Speech-Encryption-using-Hybrid-Logistic-Tent
- Appendix A: Detailed results of the experiment

## Acknowledgment

The writer would like to thank the IF4020 Cryptography lecturer, Dr. Ir. Rinaldi Munir, M.T., for teaching and supporting students in making contributions through innovative papers. The writer has gained a much deeper understanding of cryptography and its application in the real world through the materials and lectures from the course.

## References

[1] A. A. Pekerti, A. Sasongko and A. Indrayanto, "Secure End-to-End Voice Communication: A Comprehensive Review of Steganography, Modem-Based Cryptography, and Chaotic Cryptography Techniques," *IEEE Access*, vol. 12, pp. 75146–75168, 2024. [Online]. Available: https://doi.org/10.1109/ACCESS.2024.3405317 (accessed Dec. 10, 2025).

[2] L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1978.

[3] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, 2nd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 1999.

[4] M. Bellare and P. Rogaway, *Introduction to Modern Cryptography*. Online manuscript, p. 10, Sep. 2005. [Online]. Available: https://web.cs.ucdavis.edu/~rogaway/classes/227/spring05/book/main.pdf (accessed Dec. 11, 2025).

[5] L. M. Pecora, T. L. Carroll, G. A. Johnson, D. J. Mar, and J. F. Heagy, "Fundamentals of synchronization in chaotic systems, concepts, and applications," *Chaos*, vol. 7, no. 4, pp. 520–543, Dec. 1997. [Online]. Available: https://doi.org/10.1063/1.166278 (accessed Dec. 12, 2025).

[6] L. Kocarev and U. Parlitz, "General approach for chaotic synchronization with applications to communication," *Phys. Rev. Lett.*, vol. 74, no. 25, pp. 5028–5031, Jun. 1995. [Online]. Available: https://doi.org/10.1103/PhysRevLett.74.5028 (accessed Dec. 13, 2025).

[7] R. B. Naik and U. Singh, "A review on applications of chaotic maps in pseudo-random number generators and encryption," *Ann. Data Sci.*, vol. 11, no. 1, pp. 25–50, Jan. 2022. [Online]. Available: https://doi.org/10.1007/s40745-021-00364-7 (accessed Dec. 14, 2025).

[8] Y. Y. Hou, H. C. Chen, and J. F. Chang, "Synchronization of chaotic systems and its application in security terminal sensing node of Internet of Thing," *Micromachines*, vol. 13, no. 11, article 1993, Nov. 2022. [Online]. Available: https://doi.org/10.3390/mi13111993 (accessed Dec. 15, 2025).

[9] C. Pak and L. Huang, "A new color image encryption using combination of the 1D chaotic map," Signal Processing: Image Communication, vol. 52, pp. 28-36, 2017.

[10] R. Munir, "Kriptografi Kunci-Publik," IF4020 Kriptografi, Dec. 2025. [Online]. Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2025-2026/19-Kriptografi-Kunci-Publik-2025.pdf (accessed Dec. 24, 2025).

[11] N. A. Abbas, "Speech scrambling based on principal component analysis," *MASAUM Journal of Computing*, vol. 1, no. 3, pp. 452–456, 2009.

[12] R. Munir, "Pengantar Kriptografi," IF4020 Kriptografi, Dec. 2025. [Online]. Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2025-2026/01-Pengantar-Kriptografi-(2025).pdf (accessed Dec. 24, 2025).

[13] R. Munir, "Kriptografi Modern," IF4020 Kriptografi, Dec. 2025. [Online]. Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2025-2026/12-Kripto-modern-2025.pdf (accessed Dec. 24, 2025).

## Statement

I, the individual signing below, affirm that the content presented in this document is an original creation authored by me. It is not a derivative work, translation of another document, or a product of plagiarism.

Bandung, 25th December 2025,

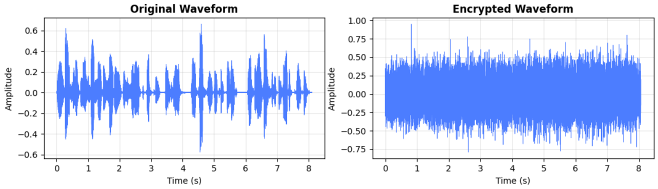
R. Francisco Trianto B., 13522091

# Appendix A
# Detailed Experimental Results

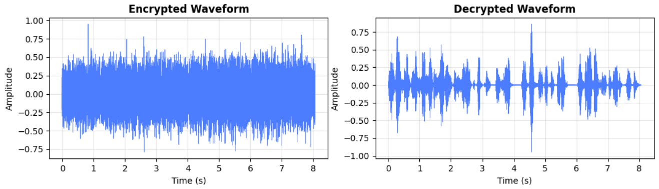
The following table presents the comprehensive visual and statistical analysis for all five experimental test cases (LJ001-0005 to LJ001-0060).

TABLE I: Combined Experimental Results: Visual and Statistical Analysis

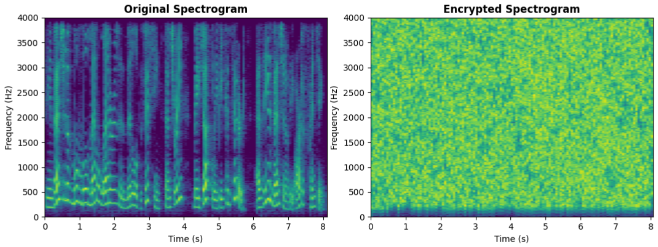
| Metric | Original | Encrypted | Decrypted |
|---|---|---|---|
| **Test 1: LJ001-0005.wav** | | | |



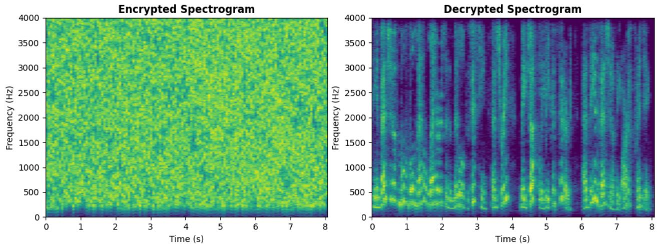
(a) Waveform: Original vs Encrypted



(b) Waveform: Encrypted vs Decrypted



(c) Spectrogram: Original vs Encrypted

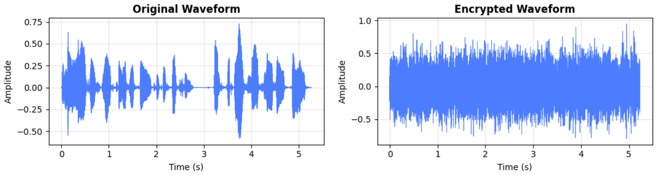


(d) Spectrogram: Encrypted vs Decrypted

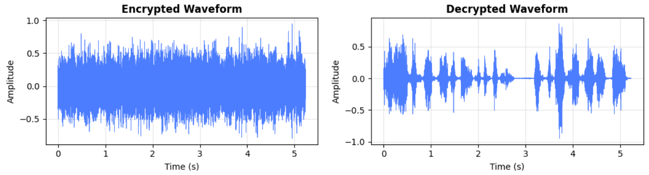
| Metric | Original | Encrypted | Decrypted |
|---|---|---|---|
| Duration (s) | 8.11 | 8.06 | 8.06 |
| Total Samples | 64,888 | 64,512 | 64,512 |
| Sample Rate (Hz) | 8000 | 8000 | 8000 |
| Mean | -0.000000 | -0.000000 | 0.000001 |
| Std Dev | 0.084840 | 0.169132 | 0.087291 |
| Min Value | -0.577631 | -0.792704 | -0.950000 |
| Max Value | 0.660921 | 0.950000 | 0.861269 |
| RMS | 0.084840 | 0.169132 | 0.087291 |

*Encryption Time: 0.030s* *Decryption Time: 0.143s*
*Key:* $r = 3.99, x_0 = 0.123456789, \mu = 0.7, y_0 = 0.654321987$

**Test 2: LJ001-0016.wav**



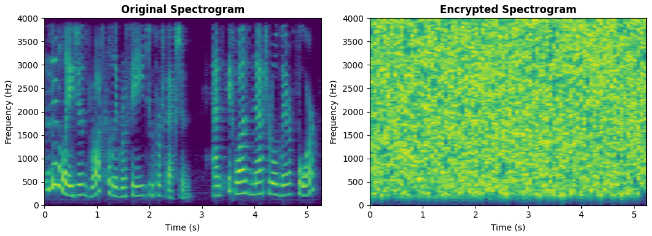
(a) Waveform: Original vs Encrypted



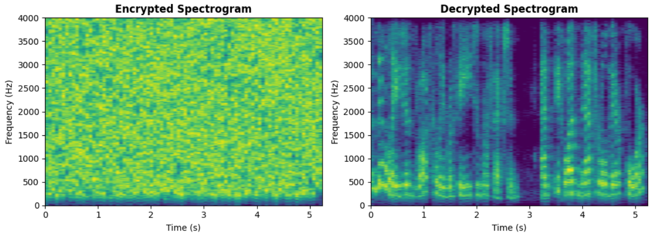
(b) Waveform: Encrypted vs Decrypted

*Continued on next page*

*Continued from previous page*



(c) Spectrogram: Original vs Encrypted

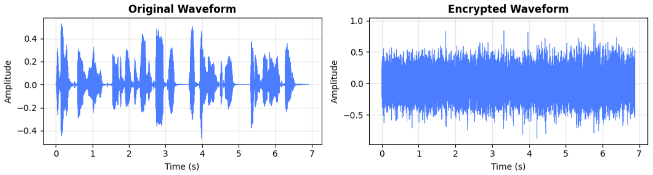


(d) Spectrogram: Encrypted vs Decrypted

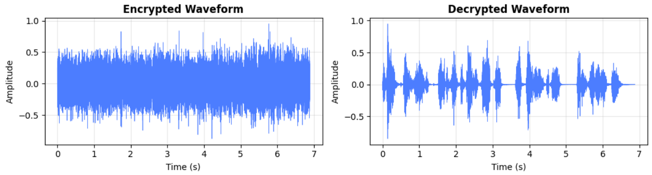
| **Metric** | **Original** | **Encrypted** | **Decrypted** |
|---|---|---|---|
| Duration (s) | 5.27 | 5.23 | 5.23 |
| Total Samples | 42,132 | 41,856 | 41,856 |
| Sample Rate (Hz) | 8000 | 8000 | 8000 |
| Mean | 0.000004 | 0.000003 | 0.000008 |
| Std Dev | 0.098797 | 0.191385 | 0.116273 |
| Min Value | -0.587300 | -0.799290 | -0.950000 |
| Max Value | 0.729306 | 0.950000 | 0.864223 |
| RMS | 0.098797 | 0.191385 | 0.116273 |

*Encryption Time: 0.021s* *Decryption Time: 0.020s*
*Key:* $r = 3.99, x_0 = 0.123456789, \mu = 0.7, y_0 = 0.654321987$

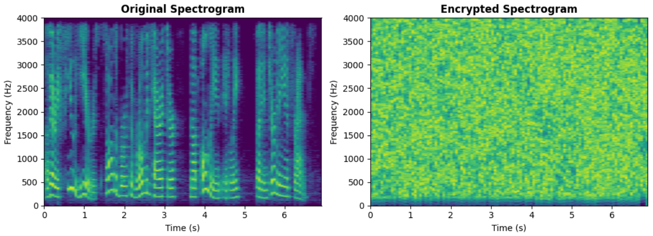
## Test 3: LJ001-0030.wav



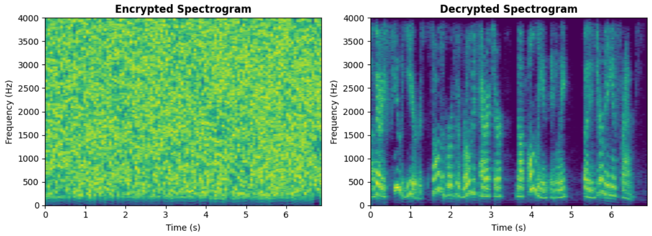
(a) Waveform: Original vs Encrypted



(b) Waveform: Encrypted vs Decrypted



(c) Spectrogram: Original vs Encrypted



(d) Spectrogram: Encrypted vs Decrypted

| | | | |
|---|---|---|---|
| Duration (s) | 6.92 | 6.88 | 6.88 |
| Total Samples | 55,321 | 55,040 | 55,040 |
| Sample Rate (Hz) | 8000 | 8000 | 8000 |
| Mean | 0.000006 | -0.000020 | 0.000016 |
| Std Dev | 0.074104 | 0.172482 | 0.093349 |
| Min Value | -0.466789 | -0.873199 | -0.849609 |
| Max Value | 0.528894 | 0.949982 | 0.950000 |
| RMS | 0.074104 | 0.172482 | 0.093349 |

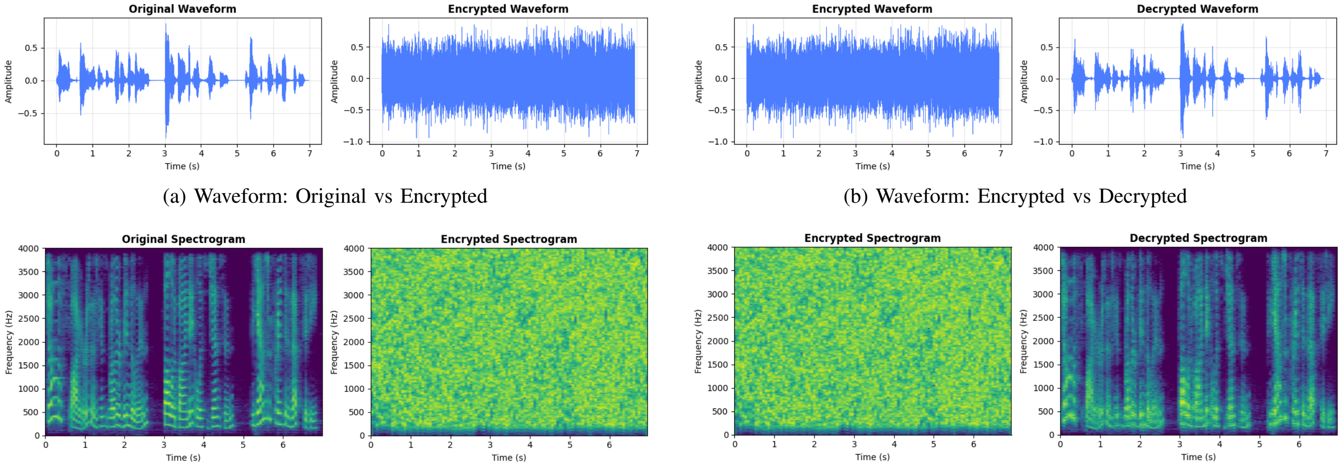
*Encryption Time: 0.026s* *Decryption Time: 0.026s*
*Key:* $r = 3.99, x_0 = 0.123456789, \mu = 0.7, y_0 = 0.654321987$

*Continued on next page*

*Continued from previous page*

| **Metric** | **Original** | **Encrypted** | **Decrypted** |
|---|---|---|---|

### Test 4: LJ001-0044.wav



(a) Waveform: Original vs Encrypted

(b) Waveform: Encrypted vs Decrypted

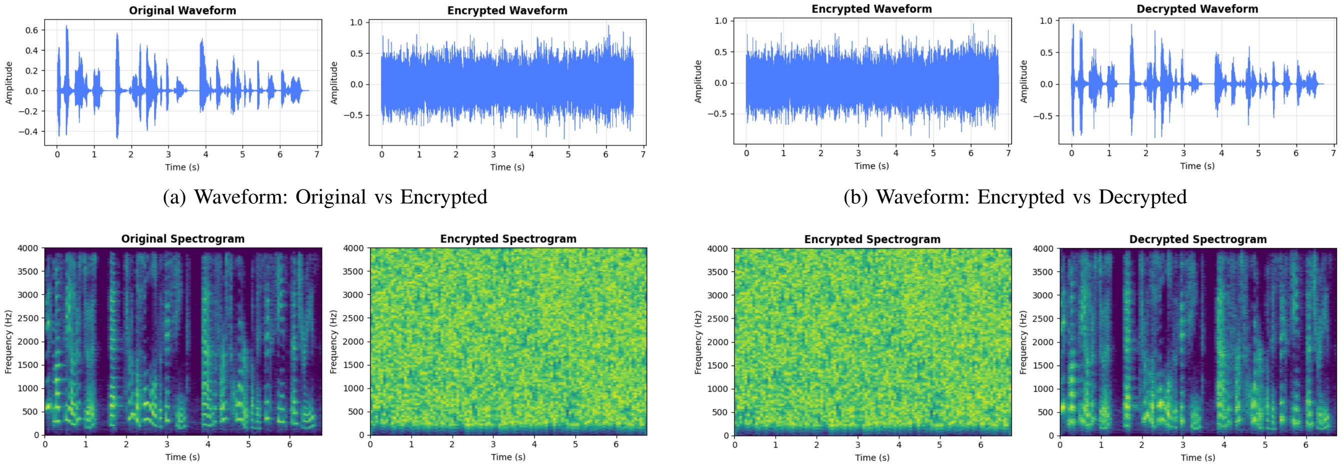
(c) Spectrogram: Original vs Encrypted

(d) Spectrogram: Encrypted vs Decrypted

| **Metric** | **Original** | **Encrypted** | **Decrypted** |
|---|---|---|---|
| Duration (s) | 6.98 | 6.94 | 6.94 |
| Total Samples | 55,878 | 55,552 | 55,552 |
| Sample Rate (Hz) | 8000 | 8000 | 8000 |
| Mean | 0.000009 | -0.000004 | 0.000014 |
| Std Dev | 0.080384 | 0.208809 | 0.087425 |
| Min Value | -0.886748 | -0.950012 | -0.950000 |
| Max Value | 0.869798 | 0.868011 | 0.873768 |
| RMS | 0.080384 | 0.208809 | 0.087425 |

*Encryption Time: 0.024s* *Decryption Time: 0.026s*
*Key:* $r = 3.99, x_0 = 0.123456789, \mu = 0.7, y_0 = 0.654321987$

### Test 5: LJ001-0060.wav



(a) Waveform: Original vs Encrypted

(b) Waveform: Encrypted vs Decrypted

(c) Spectrogram: Original vs Encrypted

(d) Spectrogram: Encrypted vs Decrypted

| **Metric** | **Original** | **Encrypted** | **Decrypted** |
|---|---|---|---|
| Duration (s) | 6.78 | 6.74 | 6.74 |
| Total Samples | 54,206 | 53,888 | 53,888 |

*Continued on next page*

*Continued from previous page*

| **Metric** | **Original** | **Encrypted** | **Decrypted** |
|---|---|---|---|
| Sample Rate (Hz) | 8000 | 8000 | 8000 |
| Mean | 0.000000 | -0.000024 | -0.000004 |
| Std Dev | 0.065236 | 0.182338 | 0.096303 |
| Min Value | -0.479124 | -0.890900 | -0.851742 |
| Max Value | 0.646948 | 0.949982 | 0.950000 |
| RMS | 0.065236 | 0.182338 | 0.096303 |

*Encryption Time: 0.024s*   *Decryption Time: 0.047s*

*Key:* $r = 3.99, x_0 = 0.123456789, \mu = 0.7, y_0 = 0.654321987$