Journal of Positive Sciences (JPS) eISSN: 2582-9351, https://positive-sciences.com JPS. 2026 Oct; 6(5): 188-249

Article info
Received on: 07.08.2026
Accepted on: 03.10.2026
Published on: 03.11.2026
doi: https://doi.org/10.52688/ASP655506

Research Article

Radius-Transformed Logistic Maps for High-Dimensional Data Generation: Statistical Characterization, Cryptanalysis, and Authenticated Encryption

Majed Ismael Sameer [1], Mohammed Rajih Jassim [2], Mohammed RASHEED [3,*]

[1] Department of computer and mathematical College, University KUFA, Najaf, Iraq
[2, 3] College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq

[*] rasheed.mohammed40@yahoo.com

ABSTRACT

The growing use of high-dimensional numerical data has renewed interest in chaos-based generation and encryption, yet nonlinear complexity does not necessarily imply statistical independence or cryptographic security. This study critically reassesses a radius-transformed logistic-map framework through mathematical analysis, reproducible numerical experiments, and cryptanalytic testing. Using a control parameter of 3.8, an initial state of 0.5, and radii of 0.776, 0.876, and 0.881, the investigation examines bifurcation behavior, finite-time Lyapunov exponents, numerical precision, statistical distributions, dimensionality, and encryption vulnerabilities. An exact sliding-window identity shows that adjacent multidimensional vectors share all but one coordinate. Across dimensions of 2, 4, 8, 16, and 32, participation-ratio effective rank increases from 1.913 to 27.862, whereas its normalized fraction decreases from 0.957 to 0.871. The one-, two-, and three-radius XOR keystreams exhibit empirical entropies of 6.99398, 6.99312, and 6.99416 bits per byte, respectively, although modulo-128 quantization limits the maximum to seven bits per symbol. A single plaintext-bit modification changes exactly one ciphertext bit, while deterministic keystream reuse permits recovery from known plaintext. The redesigned architecture separates nonlinear feature generation from AES-256-GCM authenticated encryption. Implementation tests confirm exact plaintext recovery, nonce-dependent ciphertext variation, and rejection of three deliberately modified authenticated inputs. These findings identify deterministic overlap, finite-precision sensitivity, and XOR malleability as critical limitations. The assessment also clarifies the limitations of isolated randomness tests. The principal contribution is a reproducible, evidence-based characterization of radius-transformed data and a defensible separation between chaotic numerical representations and standardized cryptographic protection, rather than a claim of a newly secure chaotic cipher.

Keywords: Logistic map; Radius-transformed sequences; High-dimensional data generation; Effective rank; Chaotic dynamics; Cryptanalysis; AES-256-GCM; Authenticated encryption

INTRODUCTION

Chaotic dynamics, including the logistic map, have been explored as a source of reproducible nonlinear structure and as a candidate ingredient of information-protection mechanisms [1]. Their appeal stems from nonlinear stretching, folding and positive finite-time sensitivity in appropriate parameter ranges [2].

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

However, an unpredictable-looking real-valued trajectory does not provide a security definition or an attacker model [3]. Cryptographic evaluation requires explicit treatment of key secrecy, nonce uniqueness, known-plaintext exposure, chosen-plaintext attacks and authentication [4].

The increasing complexity and dimensionality of digital information have created substantial challenges for data generation, numerical modeling, information representation, and secure communication [5]. Modern computational applications frequently involve high-dimensional datasets characterized by nonlinear relationships, statistical dependencies, and complex temporal behavior [6]. Generating such datasets using deterministic nonlinear systems is particularly attractive because their mathematical structures allow reproducible experimentation and systematic parameter control [7]. Within this context, chaos-based approaches provide a framework for constructing nonlinear numerical sequences and examining the relationships between dynamical complexity, statistical variability, and computational behavior [8]. Nevertheless, the suitability of these sequences for practical applications depends on properties beyond their apparent irregularity, including distributional characteristics, inter-dimensional correlations, numerical stability, and implementation efficiency [9].

Among discrete nonlinear dynamical systems, the logistic map has attracted considerable attention owing to its simple mathematical formulation and rich dynamical behavior [10]. Depending on the control parameter and initial condition, the system exhibits fixed points, periodic oscillations, period-doubling bifurcations, and chaotic regimes [11]. These characteristics make it suitable for investigating nonlinear state evolution and sensitivity to perturbations [12]. In particular, the transition from regular to chaotic dynamics can be examined through bifurcation diagrams and Lyapunov exponents, while the resulting numerical sequences can be characterized using entropy, correlation, and distributional measures [13]. However, the presence of a positive Lyapunov exponent describes the average exponential separation of nearby trajectories and does not necessarily imply statistical independence, cryptographic unpredictability, or resistance to adversarial attacks [14].

An additional challenge arises when chaotic sequences are transformed into high-dimensional numerical representations [15]. Although multidimensional constructions may increase the number of observable features, a larger nominal dimension does not automatically correspond to a proportional increase in independent information [16]. In particular, when high-dimensional vectors are assembled from consecutive or overlapping observations of a single deterministic trajectory, neighboring vectors may share substantial information [17]. This dependence can influence covariance structures, effective dimensionality, statistical redundancy, and the interpretation of generated datasets [18]. Consequently, evaluating a high-dimensional chaotic generator requires more than demonstrating its ability to produce numerical vectors with multiple coordinates [19]. A comprehensive assessment should also consider coordinate dependence, effective rank, temporal correlation, and the extent to which increasing dimensionality introduces additional statistical information [20].

The earlier radius-based manuscript generated overlapping coordinate vectors by selecting logistic states below fixed radii and mapping each accepted state through a square-root transformation [21]. It also built an encryption keystream from one to three radius sequences and XORed the resulting 7-bit integers with character values [22]. This research retains the geometric construction as the central mathematical object, but critically assesses whether dimensionality gains translate into independent information or security [23]. Related studies discuss image encryption using logistic maps, hyperchaos, coupled dynamics and nonlinear mappings, as well as finite-precision degradation in implemented chaos [24]. These literature examples motivate comparative evaluation, but do not validate the present design [25].

The radius-based transformation introduces a geometric interpretation of chaotic state evolution by mapping selected logistic values into nonnegative coordinates under prescribed radius constraints [26]. Such a construction offers an interpretable mechanism for generating bounded numerical features and examining their behavior as the number of dimensions increases [27]. However, the statistical characteristics of the resulting coordinates are governed by the original chaotic trajectory, the radius-selection rule, and the

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

nonlinear transformation [28]. Therefore, important questions remain regarding whether different radii produce meaningfully distinct information, whether the transformation preserves or modifies temporal correlations, and how overlapping coordinate windows affect the rank and dependence structure of the generated data [29]. Answering these questions is essential for establishing the mathematical significance and practical limitations of the proposed high-dimensional representation [30].

Another fundamental consideration is the distinction between deterministic chaos and cryptographic security. Several chaos-based encryption approaches have employed nonlinear maps, permutation mechanisms, diffusion procedures, and dynamically generated sequences to protect digital information [2, 4, 11, 24]. Although such methods may exhibit complex numerical behavior, security cannot be established solely through visual inspection of encrypted outputs, high empirical entropy, or sensitivity to initial conditions. In particular, XOR encryption driven by a deterministic keystream may reveal relationships between plaintext and ciphertext if the same keystream is reused. Furthermore, a scheme without an authentication mechanism may fail to detect unauthorized modifications to encrypted data. These limitations emphasize the need for explicit cryptanalysis, including known-plaintext and chosen-plaintext considerations, keystream reuse analysis, differential behavior, and integrity verification [1, 4, 29, 30]. This distinction is consistent with established cryptographic guidance requiring clear implementation specifications, key management, and systematic security evaluation.

The numerical implementation of chaotic systems introduces additional limitations that must be distinguished from their ideal mathematical behavior. In theoretical analysis, chaotic maps are commonly treated as real-valued dynamical systems, whereas practical computer implementations operate with finite-precision arithmetic. Rounding, truncation, and restricted numerical representations can alter trajectories, produce implementation-dependent sequences, and introduce periodic behavior absent from idealized descriptions. Such effects are especially important when encrypted data must be reproduced exactly during decryption. Consequently, investigating the divergence between single-precision and double-precision implementations provides useful evidence about numerical reproducibility and sensitivity, although it does not independently establish cryptographic strength [1, 4, 25–27].

Statistical characterization also represents an important component of evaluating chaos-derived numerical sequences. Measures such as Shannon entropy, frequency distributions, correlation coefficients, and effective rank provide complementary information about variability and dependence. However, these measures describe different properties and should not be interpreted interchangeably. A sequence may exhibit relatively high empirical entropy while retaining deterministic relationships or structural constraints that make it unsuitable for cryptographic applications. Similarly, weak pairwise correlation does not necessarily imply independence or resistance to prediction. Therefore, statistical evaluation should be interpreted alongside the mathematical structure of the generator and explicit cryptanalytic analysis rather than being treated as conclusive evidence of security [5, 28, 30].

From a practical perspective, a more defensible approach is to distinguish the role of chaos-derived numerical representations from the role of established cryptographic primitives. The logistic-map radius transformation can be investigated as a reproducible data-generation mechanism, whereas data confidentiality and integrity can be provided through authenticated encryption. In this study, the Advanced Encryption Standard operating in Galois/Counter Mode (AES-256-GCM) is considered as a standard-primitive security layer. This separation allows the mathematical properties of radius-transformed data to be evaluated independently of the confidentiality and authentication guarantees associated with the encryption mechanism. It also avoids attributing the established security properties of AES-GCM to the chaotic transformation itself [6–10, 29].

Despite extensive research on chaos-based encryption and nonlinear numerical generation, an important methodological gap remains in the evaluation of radius-conditioned logistic representations. Specifically, the relationships among radius selection, overlapping-window construction, nominal dimensionality, effective statistical dimensionality, finite-precision sensitivity, and encryption vulnerability require a unified

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

assessment. Many published approaches emphasize encryption-output variability or statistical randomness, whereas the present investigation focuses on the structural dependence inherent in the generated coordinates and the security consequences of the original deterministic XOR construction. Accordingly, the research does not assume that increasing the number of radii necessarily improves cryptographic security. Instead, it examines which properties can be demonstrated mathematically, which require numerical verification, and which must be supported by established cryptographic mechanisms.

To address these issues, the present study develops a unified mathematical and computational framework for evaluating radius-transformed logistic maps in high-dimensional data generation and encryption. The analysis considers five nominal dimensions (2D, 4D, 8D, 16D, and 32D), together with one-, two-, and three-radius keystream configurations. The investigation combines nonlinear dynamical analysis, statistical characterization, dimensionality assessment, numerical-precision experiments, differential cryptanalysis, and authenticated-encryption verification. Particular attention is devoted to identifying the distinction between increasing the number of generated coordinates and increasing the amount of statistically independent information.

The principal contributions of this study are summarized as follows:

(i) An unambiguous mathematical formulation of the continuous radius-transformed logistic generator and its corresponding discrete encryption rule is established, clarifying the roles of logistic parameters, radius thresholds, numerical transformations, and dimensional construction.

(ii) The dependence introduced by overlapping high-dimensional windows is examined analytically, while the deterministic malleability and keystream-reuse vulnerabilities of the original XOR construction are explicitly characterized.

(iii) Reproducible numerical experiments are conducted across five nominal dimensions and three encryption variants to evaluate statistical characteristics, effective rank, inter-dimensional dependence, and the relationship between radius configuration and generated numerical behavior.

(iv) The sensitivity of logistic trajectories to finite-precision implementation is investigated to assess numerical divergence, computational reproducibility, and practical limitations associated with deterministic chaotic calculations.

(v) An authenticated-encryption redesign based on established cryptographic primitives is introduced to separate the mathematical contribution of radius-transformed logistic representations from the requirements of confidentiality, integrity, and secure key management.

The remainder of this paper is organized as follows. Section 2 reviews the relevant literature and identifies the research gap. Section 3 presents the mathematical formulation of the logistic-map radius transformation, overlapping multidimensional representation, baseline XOR encryption, and authenticated-encryption redesign. Section 4 describes the computational methodology, experimental conditions, and evaluation criteria. Section 5 presents and discusses the dynamical, statistical, dimensionality, cryptanalytic, and performance results. Section 6 addresses reproducibility, implementation limitations, and responsible interpretation of the findings. Finally, Section 7 summarizes the principal conclusions and identifies directions for further investigation.

LITERATURE REVIEW AND RESEARCH GAP

Pecora and Carroll established foundational chaotic synchronization [2], while Fridrich developed geometric permutation/diffusion constructions [3]. Alvarez and Li subsequently stressed conventional design requirements and demonstrated vulnerabilities in logistic-like ciphers [1, 4]. Formal generator-evaluation

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

methodology is described in NIST SP 800-22, which explicitly warns that statistical tests cannot replace cryptanalysis [5]. NIST SP 800-90A provides standardized deterministic random-bit-generator methods [8]. AES and GCM are specified by FIPS 197 and SP 800-38D, respectively [6, 7]; HKDF, ChaCha20-Poly1305 and authenticated-encryption interfaces provide complementary engineering foundations [9, 10, 29].

Table 1. Positioning of the present work against selected literature**.**

Approach / references	Main emphasis	Limitation relative to this study
Chaos synchronization [2]	Dynamics/synchronization	Not an AEAD security guarantee
Geometric chaotic ciphers [3]	Permutation and diffusion	Implementation security requires analysis
Chaos cryptanalysis [1,4]	Threat models and attacks	Does not analyze the specific radius-window generator
Image/hybrid proposals [11–24]	Specialized image/data transformations	May target different formats and models
NIST standards and RFCs [5–10,29]	Statistical/cryptographic best practices	Do not address radius-feature dependence
Present study	Reproducibility + negative cryptanalysis + AEAD	No new encryption security proof is claimed

The research gap is not the absence of chaotic encryption schemes, but the lack of rigorous separation between nominal data dimensionality, statistical independence and cryptographic strength in radius-conditioned logistic implementations. This gap is addressed through mathematical identities and reproducible counterexamples, not by asserting that more radii automatically lead to stronger encryption.

MATHEMATICAL FORMULATION

LOGISTIC EVOLUTION, ACCEPTANCE AND TRANSFORMATION

The proposed mathematical framework is constructed by integrating nonlinear logistic-map dynamics with a radius-conditioned acceptance mechanism and a geometric coordinate transformation. The primary objective of this formulation is to generate bounded, reproducible numerical sequences that can subsequently be organized into multidimensional representations. Unlike approaches that directly employ raw chaotic states as numerical features, the present framework introduces a radius-dependent transformation to investigate how geometric constraints influence the distribution, variability, and dependence structure of the generated data. The mathematical model consists of three consecutive stages: nonlinear state evolution, threshold-based state selection, and radius-conditioned coordinate generation.

The first stage describes the temporal evolution of a scalar nonlinear dynamical system. The logistic map is selected because of its relatively simple mathematical structure and its ability to exhibit qualitatively different dynamical regimes under variations in the control parameter. These regimes include stable fixed points, periodic oscillations, period-doubling transitions, and chaotic behavior. The resulting sequence is deterministic for a specified initial condition and control parameter, although trajectories may become highly sensitive to small perturbations within suitable parameter regions. This property makes the logistic map useful for studying nonlinear numerical generation and sensitivity to computational precision. Nevertheless, the apparent irregularity of its trajectories should not be interpreted as proof of statistical independence or cryptographic unpredictability [25–27].

$$x_{t+1} = r x_t (1 - x_t), \qquad 0 < x_t < 1, \quad 0 < r \leq 4 \tag{1}$$

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

where (x_t) represents the system state at iteration (t), (x_{t+1}) denotes the subsequent state, and (r) is the nonlinear control parameter governing the dynamical behavior. The initial condition (x_0) determines the starting position of the trajectory. For the present numerical investigation, the control parameter is fixed at (r=3.8), while the initial state is selected as (x_0=0.5), consistent with the baseline configuration examined in this work. Unless explicitly stated otherwise, all calculations are performed using IEEE-754 double-precision floating-point arithmetic to ensure a consistent computational representation.

Although the logistic recursion generates values within a bounded interval under the stated parameter conditions, not every generated state is necessarily suitable for the subsequent geometric transformation. In particular, the proposed radius-conditioned mapping requires that the selected state remain smaller than a prescribed radius threshold. Therefore, a deterministic acceptance mechanism is introduced to extract a subsequence of logistic states satisfying this constraint. This procedure does not alter the underlying logistic recursion; instead, it filters the generated trajectory according to a fixed geometric condition. Consequently, the accepted sequence may have statistical characteristics different from those of the original unfiltered trajectory.

The accepted states are defined by

$$a_j(k) = x_{\tau_j}, \qquad \tau_j = \min\{t > \tau_{j-1} : x_t < k\} \tag{2}$$

where ($a_j(k)$) denotes the (j)-th accepted logistic state associated with radius (k), and (τ_j) represents the iteration index at which the corresponding acceptance condition is satisfied. The radius threshold (k) is restricted to (0<k<1). The accepted states are thus selected sequentially from the original logistic trajectory, preserving their temporal order while excluding values that do not satisfy the geometric constraint. The initial accepted index is determined by applying the rule after a specified starting iteration; this convention must remain consistent throughout data generation and reproducibility tests.

The selection mechanism is important because the acceptance rate depends on both the radius threshold and the distribution of states visited by the underlying logistic trajectory. A smaller radius generally imposes a more restrictive acceptance condition, whereas a larger radius permits a broader subset of states. However, the precise acceptance rate is governed by the trajectory and cannot be assumed to vary linearly with the threshold. Furthermore, filtering does not eliminate deterministic dependence between accepted states. This distinction becomes particularly relevant when the selected values are subsequently used to construct multidimensional vectors or encryption-related numerical sequences.

The third stage introduces a nonlinear geometric transformation that converts each accepted logistic state into a nonnegative coordinate associated with a prescribed radius. The transformation is motivated by the geometric relation between a radius and orthogonal coordinates in the first quadrant of a Cartesian plane. For a fixed radius, an accepted logistic value is treated as one coordinate, while the complementary coordinate is obtained through the corresponding square-root relation. This construction provides an interpretable method for mapping accepted states into bounded numerical features without requiring additional independent chaotic systems.

The transformed coordinate is calculated as

$$y_j(k) = \sqrt{k^2 - a_j(k)^2}, \qquad 0 < k < 1 \tag{3}$$

where ($y_j(k)$) represents the transformed coordinate associated with the (j)-th accepted state and radius (k). Since the acceptance condition guarantees (0<a_j(k)<k), the quantity inside the square root is nonnegative, ensuring that the generated coordinate remains real-valued. The resulting transformation satisfies the geometric relationship($\left(a_j^{(k)}\right)^2 + \left(y_j^{(k)}\right)^2 = k^2$), thereby associating each accepted state with a point on a quarter-circle of radius (k).

An important mathematical consequence of Eq. (3) is that the transformed coordinates remain bounded within the interval ((0, k)), while their statistical distribution is determined jointly by the logistic-map dynamics, the acceptance threshold, and the nonlinear square-root mapping. The transformation is monotonic decreasing with respect to the accepted state over the admissible interval, meaning that larger accepted logistic values produce smaller complementary coordinates. Accordingly, the transformation

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

changes the numerical distribution of the accepted sequence but does not create new independent randomness. The resulting coordinate values should therefore not be presumed uniformly distributed without supporting statistical evidence [25–27].

The proposed acceptance mechanism also avoids introducing artificial numerical values through an unnecessary absolute-value correction. Specifically, replacing the transformation in Eq. (3) with a square-root expression containing($abs(k^2 - x^2)$) would conceal states that violate the prescribed radius condition rather than enforcing the intended geometric constraint. In the present formulation, invalid states are rejected before transformation, ensuring consistency between the mathematical definition and its numerical implementation.

The radius-conditioned transformation provides the foundation for extending the scalar chaotic sequence into higher-dimensional representations. By arranging consecutive transformed coordinates into vectors, numerical datasets of different nominal dimensions can be constructed without changing the underlying one-dimensional logistic recursion. However, such dimensional expansion must be interpreted carefully because consecutive vectors may share coordinates and preserve temporal dependencies inherited from the original trajectory. Therefore, the mathematical structure of these dependencies, together with their influence on effective dimensionality and statistical redundancy, is examined in the following subsection.

Eqs. (1)–(3) establish a deterministic and reproducible mathematical procedure for generating radius-transformed logistic sequences. The framework combines nonlinear evolution, conditional state selection, and bounded geometric mapping, providing a basis for subsequent statistical and dimensionality analyses. Its scientific significance lies in characterizing how these operations affect numerical representations rather than assuming that geometric transformation or increased nominal dimensionality automatically produces independent information or stronger cryptographic security.

OVERLAPPING HIGH-DIMENSIONAL WINDOWS AND DEPENDENCE

The transformation of one-dimensional chaotic sequences into high-dimensional numerical representations constitutes a central component of the proposed framework. Although the logistic map evolves through a single scalar state variable, its successive outputs can be organized into multidimensional vectors using a sliding-window construction. This procedure provides a systematic mechanism for generating datasets with controllable nominal dimensionality while preserving the nonlinear characteristics inherited from the underlying chaotic trajectory. However, increasing the number of coordinates does not necessarily introduce an equivalent increase in statistically independent information. Therefore, the mathematical relationships between adjacent vectors must be explicitly examined to distinguish nominal dimensional expansion from effective dimensionality.

Following the radius-conditioned transformation described in Section 3.1, the accepted chaotic states are converted into a sequence of bounded numerical coordinates. These coordinates are subsequently arranged into overlapping windows of a prescribed dimension, denoted by (d). Each multidimensional vector consists of (d) consecutive transformed values extracted from the same radius-conditioned sequence. The resulting representation maintains the original temporal ordering of the generated coordinates and enables systematic investigation of how increasing dimensionality affects the statistical structure of the data.

For a selected radius (k), the (i)-th multidimensional vector is defined as

$$\mathrm{v}_i^{(d)} = [y_i(k), y_{i+1}(k), \dots, y_{i+d-1}(k)] \quad \textbf{(4)}$$

where ($\mathrm{v}_i^{(d)}$) denotes the (i)-th vector in a (d)-dimensional representation, and ($y_i(k)$) represents the transformed coordinate generated using the radius-dependent mapping in Eq. (3). The index (i) specifies the starting position of each window, while (d) determines the number of coordinates included in the corresponding vector. For a transformed sequence containing (N) observations, the unit-stride sliding-window procedure generates (N-d+1) vectors, provided that (N≥d).

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

This construction allows the same underlying chaotic trajectory to produce datasets with different nominal dimensions without modifying the logistic control parameter or the acceptance mechanism. In the present investigation, five dimensional configurations are considered: 2D, 4D, 8D, 16D, and 32D. These configurations provide a systematic basis for evaluating how dimensional expansion influences coordinate dependence, statistical variability, and the distribution of information across the generated features. Importantly, the resulting vectors should be regarded as structured representations of a common deterministic sequence rather than independent samples from an unspecified multidimensional probability distribution.

A distinguishing characteristic of the sliding-window construction is the deterministic overlap between consecutive vectors. When the window advances by one position, the final (d-1) coordinates of the preceding vector are reproduced as the first (d-1) coordinates of the subsequent vector. This relationship is exact and does not depend on the chaotic regime, radius threshold, or numerical distribution of the transformed sequence. Consequently, the generated dataset contains repeated coordinate information across neighboring observations, even when the individual numerical values exhibit irregular variations.

The coordinate-sharing relationship is expressed as

$$v_i^{(d)}[j+1] = v_{i+1}^{(d)}[j], \qquad j = 1, \dots, d-1 \tag{5}$$

where ($v_i^{(d)}[j]$) denotes the (j)-th coordinate of the (i)-th multidimensional vector. Equation (5) establishes that two consecutive vectors share exactly (d-1) coordinate values. Therefore, the fraction of shared coordinates increases with the nominal dimension when the sliding-window step remains fixed at one. For example, two consecutive 4D vectors contain three common numerical values, whereas two consecutive 32D vectors contain 31 common values.

This deterministic relationship has important consequences for interpreting the generated datasets. Although a larger number of dimensions increases the number of displayed coordinates, the additional coordinates are not generated independently of the existing sequence. Instead, they are obtained from successive observations of the same underlying dynamical process. As a result, adjacent multidimensional vectors exhibit structural dependence that cannot be eliminated merely by increasing the dimensionality. Such dependence should be distinguished from ordinary pairwise statistical correlation, because Eq. (5) describes exact equality between shifted coordinate positions in neighboring vectors.

The overlap structure also affects the interpretation of apparent complexity. A high-dimensional representation may display numerous coordinate fluctuations and a relatively broad distribution of numerical values, yet its underlying information content remains constrained by the deterministic generation procedure. Therefore, visual irregularity and increasing nominal dimensionality should not be interpreted as evidence of independent randomness or improved cryptographic security. Instead, the statistical properties of the resulting vectors must be evaluated using appropriate quantitative measures that account for dependence, redundancy, and the distribution of variance across coordinates.

To characterize the dimensional structure of the generated data, the empirical correlation matrix is constructed from the multidimensional observations. Let (R) denote the (d × d) sample correlation matrix obtained after arranging the generated vectors into a numerical data matrix. The eigenvalues of (R), denoted by
$(\lambda_1, \lambda_2, \dots, \lambda_d)$, describe the distribution of standardized variance across its principal directions. When variance is distributed approximately equally among the eigenmodes, the representation exhibits a relatively broad effective dimensional structure. Conversely, when most variance is concentrated in a smaller number of eigenmodes, the generated dataset contains substantial linear redundancy.

The effective dimensionality is evaluated using the participation-ratio measure, expressed as

$$d_{\text{eff}} = \frac{\left(\sum_{i=1}^{d} \lambda_i\right)^2}{\sum_{i=1}^{d} \lambda_i^2}, \qquad 1 \le d_{\text{eff}} \le d \tag{6}$$

where (d_{eff}) represents the participation-ratio effective rank, (d) is the nominal dimensionality, and (λ_i) denotes the (i)-th eigenvalue of the empirical correlation matrix. The numerator represents the squared sum

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

of the eigenvalues, whereas the denominator quantifies their squared concentration. This formulation provides a continuous measure of the number of substantially contributing eigenmodes, rather than relying solely on the number of nonzero eigenvalues.

The value of (d_{eff}) approaches the nominal dimension (d) when the eigenvalues are approximately equal, indicating that the standardized variance is distributed relatively evenly among the available principal directions. In contrast, a smaller value indicates that the variance is concentrated in fewer directions, suggesting stronger linear redundancy among the coordinates. Nevertheless, a high effective rank does not establish statistical independence, because nonlinear dependence and deterministic relationships between successive vectors may remain present even when the correlation matrix is close to the identity matrix.

To facilitate comparisons among different dimensional configurations, the ratio between effective and nominal dimensionality can be considered. This normalized measure indicates the proportion of nominal dimensions contributing to the participation-ratio estimate. It is particularly useful for interpreting whether dimensional expansion produces a proportional increase in effective rank or whether the relative contribution of additional coordinates gradually decreases.

In the present study, the effective-dimensionality analysis is performed for the 2D, 4D, 8D, 16D, and 32D configurations. The numerical results indicate that the effective rank increases with nominal dimensionality, while the normalized effective-rank fraction exhibits a decreasing trend. This observation suggests that increasing the number of coordinates distributes variance across additional principal directions, but does not produce a proportional increase in the effective dimensionality relative to the number of generated features. The detailed numerical findings are presented in Section 5.3 and summarized through the effective-rank diagram, Radar Plot, and Heatmap Matrix.

The effective-rank analysis is complemented by examination of the empirical Pearson correlation matrix. This analysis provides information about linear relationships among the generated coordinates and allows the identification of systematic dependence patterns inherited from the common logistic trajectory. However, the correlation matrix and overlap identity describe different aspects of the generated structure. The former quantifies linear associations among coordinate columns across observations, whereas the latter demonstrates exact coordinate reuse between neighboring sliding windows. Therefore, both forms of analysis are necessary for a meaningful interpretation of the dimensionality and redundancy characteristics of the proposed framework.

From a computational perspective, the sliding-window mechanism offers a straightforward and reproducible procedure for constructing high-dimensional datasets from a scalar sequence. The principal advantage is its ability to produce representations of different dimensions without requiring additional independent nonlinear dynamical systems. Nevertheless, the generated coordinates remain linked to a common trajectory, and the overlapping-window organization introduces exact temporal redundancy. These properties may be useful in applications requiring structured temporal embeddings or reproducible nonlinear features, but they should not be interpreted as evidence of independent high-dimensional randomness.

Eqs. (4)–(6) establish the mathematical basis for generating and characterizing overlapping high-dimensional representations from radius-transformed logistic sequences. Equation (4) defines the vector construction, Eq. (5) identifies the exact deterministic dependence between adjacent vectors, and Eq. (6) provides a quantitative measure of effective linear dimensionality. Together, these relationships demonstrate why nominal dimensional expansion must be evaluated alongside structural overlap, correlation, and effective rank. This formulation provides the theoretical foundation for interpreting the dimensionality-scaling experiments and the associated statistical visualizations discussed in Section 5.3.

LEGACY KEYSTREAM AND DIFFERENTIAL VULNERABILITY

The encryption mechanism implemented in the original radius-based logistic framework relies on the generation of a deterministic numerical keystream derived from nonlinear chaotic trajectories. In this approach, logistic-map states are first processed through the radius-conditioned acceptance and geometric transformation procedures described in Section 3.1. The resulting numerical coordinates are subsequently converted into integer values and combined with plaintext using the bitwise exclusive-OR (XOR) operation.

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

The primary motivation for employing this construction was to exploit the nonlinear behavior of logistic sequences and investigate whether introducing multiple radius-dependent transformations could increase ciphertext variability and improve encryption performance.

However, the security of a keystream-based encryption mechanism cannot be established solely through the apparent irregularity of its numerical outputs. A cryptographically secure construction must additionally address key secrecy, keystream unpredictability, message-dependent initialization, resistance to plaintext exposure, and prevention of keystream reuse. These requirements are particularly important for deterministic chaotic systems because identical initial conditions and system parameters reproduce identical numerical trajectories. Consequently, the original encryption mechanism is examined here as a baseline algorithm whose mathematical properties and security limitations can be evaluated explicitly [1,4].

RADIUS-CONDITIONED KEYSTREAM GENERATION

The original encryption algorithm generates a sequence of numerical masks by applying one or more radius-dependent transformations to logistic-map trajectories. Each transformed sequence is associated with a prescribed radius, and the resulting coordinates are combined multiplicatively when multiple radii are considered. The multiplication procedure introduces additional nonlinear numerical operations, but the resulting sequence remains completely determined by the selected control parameter, initial condition, radius thresholds, and arithmetic implementation.

To convert the continuous-valued transformed coordinates into discrete quantities suitable for XOR encryption, the resulting numerical values are multiplied by a fixed scaling factor, rounded to the nearest integer, and reduced modulo 128. For a system involving (m) radius-conditioned sequences, the integer keystream is mathematically defined as

$$s_i^{(m)} = \text{round}\left(10^5 \prod_{j=1}^{m} y_i\left(k_j\right)\right) \bmod 128 \quad \textbf{(7)}$$

where ($s_i^{(m)}$) denotes the integer keystream value at position (i), (m) represents the number of radius-conditioned sequences, ($y_i(k_j)$ denotes the (i)-th accepted and transformed coordinate associated with radius (k_j), and (10^5) is the fixed scaling factor employed in the original implementation. The modulo operation restricts the numerical output to the integer interval from 0 to 127.

The multiplication of several radius-conditioned coordinates changes the numerical distribution of the generated keystream. Nevertheless, this transformation does not introduce independent secret information or establish cryptographic unpredictability. All output values remain deterministic functions of the original system parameters. Therefore, increasing the number of radii should be interpreted as increasing the number of numerical transformations rather than necessarily increasing the effective security level.

An important consequence of Eq. (7) is the restriction imposed by the modulo operation. Since the keystream contains only 128 possible integer values, its maximum marginal Shannon entropy is limited to 7 bits per byte, even if all 128 values occur with equal probability. Furthermore, the most significant bit of each keystream byte is necessarily zero. This restriction introduces a structural weakness because the corresponding most significant bit of the plaintext remains unchanged during XOR encryption.

The scaling factor and rounding operation also introduce quantization effects by mapping different continuous-valued coordinate products to identical integers. Consequently, distinct transformed logistic values may produce the same keystream symbol. Although these operations make the numerical sequence suitable for discrete processing, they cannot be regarded as sufficient mechanisms for secure key generation. The resulting distribution and repetition characteristics must therefore be analyzed quantitatively rather than inferred from the nonlinear nature of the underlying system.

ENCRYPTION AND DECRYPTION OPERATIONS

After constructing the integer keystream, the original algorithm encrypts each plaintext value through a bitwise XOR operation. In the baseline implementation, the input text is represented as a sequence of numerical character values, while the keystream provides the corresponding masks. Encryption is performed

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

independently at each position, meaning that the ciphertext value at a particular position depends only on the corresponding plaintext value and keystream symbol.

The encryption and decryption operations are defined as

$$c_i = p_i \oplus s_i^{(m)}, \qquad p_i = c_i \oplus s_i^{(m)} \tag{8}$$

where (p_i) denotes the plaintext byte at position (i), (c_i) represents the corresponding ciphertext byte, ($s_i^{(m)}$) is the generated keystream byte, and ($\oplus$) denotes the bitwise XOR operation.

The reversibility of Eq. (8) follows directly from the algebraic properties of XOR. Applying the same keystream value to the ciphertext restores the original plaintext because the XOR of a binary value with itself is zero. Therefore, successful decryption confirms the functional consistency of the encryption and decryption procedures, provided that identical keystream values are reproduced.

Nevertheless, successful plaintext recovery does not independently demonstrate confidentiality or resistance to cryptanalysis. Since each plaintext byte is processed separately, the encryption operation does not provide diffusion across multiple byte positions. As a result, modifications to individual plaintext bits produce corresponding modifications at the same ciphertext positions when the keystream remains unchanged.

This property is particularly important when interpreting the avalanche effect. In conventional cryptographic evaluation, strong diffusion is often associated with substantial changes in output bits following a small change in the input. In contrast, the original XOR construction does not propagate a plaintext-bit modification across the remaining ciphertext because the keystream is generated independently of plaintext content. Therefore, the baseline construction cannot demonstrate a conventional plaintext avalanche effect through this XOR operation alone.

Another limitation arises from the deterministic reproduction of the keystream. When the same logistic parameters, initial state, radius configuration, and numerical implementation are reused, the resulting keystream is identical. The algorithm does not include a message-specific nonce or initialization mechanism that would ensure distinct keystreams for different encryption operations. Consequently, encrypting different messages with the same baseline configuration creates a direct relationship between their ciphertexts.

DIFFERENTIAL VULNERABILITY AND KEYSTREAM REUSE

The most significant security limitation of the original construction is its vulnerability to deterministic keystream reuse. Consider two plaintext sequences encrypted using the same keystream. Because XOR is associative and satisfies the self-cancellation property, applying XOR between the corresponding ciphertext values eliminates the shared keystream. The resulting expression depends only on the two plaintext values, revealing information about their relationship.

This vulnerability can be expressed mathematically as

$$c_i \oplus c_i' = p_i \oplus p_i' \tag{9}$$

where (c_i) and (c_i') represent ciphertext bytes obtained from two different plaintexts, (p_i) and (p_i') denote their corresponding plaintext bytes, and the same keystream value is assumed to have been used at position (i).

Equation (9) establishes an exact algebraic relationship rather than a statistical approximation. The keystream is eliminated entirely from the expression, meaning that ciphertext comparison reveals the XOR difference between the underlying plaintext values. This property is independent of the nonlinear behavior of the logistic map and remains valid regardless of the number of radius-conditioned sequences employed.

A related vulnerability occurs under a known-plaintext attack. If an adversary obtains a plaintext–ciphertext pair generated by the original algorithm, the corresponding keystream bytes can be recovered directly by applying XOR between the known plaintext and ciphertext. Once these keystream values are obtained, they can be used to recover the corresponding positions of another message encrypted with the same keystream.

Under chosen-plaintext conditions, the weakness becomes even more explicit. An adversary able to request the encryption of a selected plaintext can recover the corresponding keystream values. For example,

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

encrypting an all-zero byte sequence produces ciphertext bytes equal to the keystream itself. Therefore, the original construction does not provide resistance to chosen-plaintext attacks under a deterministic reusable-keystream model.

The differential behavior can also be examined by introducing a controlled single-bit modification into the plaintext while keeping the keystream fixed. Because the encryption is performed independently through XOR, the resulting ciphertext differs only at the modified bit position. Consequently, a single plaintext-bit flip produces exactly one ciphertext-bit change. This result demonstrates the absence of cross-position diffusion and explains why ciphertext variability alone is insufficient for evaluating encryption strength.

Importantly, the security weakness persists even if multiple radius-conditioned transformations are introduced. Multiplication of additional chaotic coordinates may change the keystream values and their statistical distribution, but it does not alter the algebraic structure of Eq. (8). Therefore, the differential relationship established in Eq. (9) remains valid whenever the same keystream is reused. Increasing the number of radii cannot independently eliminate this vulnerability.

KEY SECRECY AND SECURITY IMPLICATIONS

A further limitation of the original algorithm concerns the treatment of system parameters as encryption keys. In the baseline configuration, the logistic control parameter, initial condition, and radius thresholds were specified explicitly. Such published constants do not constitute secret cryptographic keys. Therefore, the corresponding keystream is reproducible by any party with access to the algorithm and its implementation details.

Even if the logistic parameters were replaced with secret numerical values, this modification alone would not establish security. A robust cryptographic design requires a well-defined key space, secure key-generation procedures, an explicit initialization mechanism, appropriate key separation, and resistance to practical cryptanalytic attacks. In addition, finite-precision arithmetic must be considered because the number of distinct representable states is limited and numerical implementations may introduce unexpected periodic behavior [1, 4, 30].

The original algorithm also lacks an authentication mechanism for detecting unauthorized modifications to ciphertext. Because XOR encryption is malleable, an attacker can alter selected ciphertext bits and induce corresponding changes in the decrypted plaintext without necessarily knowing the original message. Consequently, confidentiality and message integrity must be evaluated as distinct security properties.

These limitations demonstrate that the statistical complexity of radius-conditioned logistic sequences should not be confused with cryptographic security. A sequence may exhibit relatively high empirical entropy and nonlinear numerical variation while remaining vulnerable to deterministic recovery, known-plaintext exposure, chosen-plaintext attacks, or unauthorized modification.

IMPLICATIONS FOR THE PROPOSED FRAMEWORK

The preceding analysis establishes that the original radius-conditioned XOR algorithm is suitable as a reproducible numerical baseline for investigating chaotic sequence generation but should not be regarded as a secure encryption mechanism. Its weaknesses arise primarily from deterministic keystream reproduction, restricted output support, lack of diffusion, and absence of message authentication.

The findings also clarify the distinction between two objectives of the present research. The first is to characterize the mathematical and statistical properties of radius-transformed logistic sequences and their high-dimensional representations. The second is to protect digital information using cryptographic mechanisms with established security properties. These objectives should be investigated separately because improvements in numerical variability or nominal dimensionality do not necessarily translate into improvements in encryption security.

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

Accordingly, the original XOR procedure is retained only for comparative analysis and vulnerability assessment. The revised framework employs authenticated encryption based on established cryptographic primitives to provide a more defensible approach to confidentiality and integrity. This separation preserves the mathematical contribution of the radius-transformed logistic generator while avoiding unsupported security claims.

Eqs. (7)–(9) demonstrate the fundamental properties and limitations of the baseline encryption procedure. Equation (7) defines the deterministic radius-conditioned keystream, Eq. (8) establishes the reversible XOR encryption and decryption mechanism, and Eq. (9) identifies the exact differential relationship that arises when the same keystream is reused. Together, these expressions provide the mathematical foundation for the cryptanalytic experiments and differential-security evaluation presented in Section 5.4.

SENSITIVITY, ENTROPY AND VALID INTERPRETATION

The quantitative evaluation of radius-transformed logistic sequences requires a clear distinction between nonlinear dynamical behavior, statistical variability, and cryptographic security. Although these properties are frequently discussed together in chaos-based encryption research, they characterize fundamentally different aspects of a numerical system. Dynamical sensitivity describes how initially nearby trajectories evolve under repeated nonlinear transformations, whereas statistical measures evaluate the distribution and dependence of observed numerical outputs. Cryptographic security, in contrast, requires resistance to adversarial attempts to recover protected information or distinguish ciphertext from the output of an appropriately defined secure encryption process. Consequently, no single dynamical or statistical indicator can independently establish the security of a chaos-based encryption algorithm.

In the present study, three complementary quantities are employed to characterize the generated sequences: the finite-time Lyapunov exponent, Shannon entropy, and lag-one correlation coefficient. These measures are selected because they provide interpretable information about trajectory sensitivity, numerical diversity, and short-range linear dependence, respectively. Their combined analysis facilitates a more systematic assessment of how logistic-map dynamics and radius-conditioned transformations influence the generated data. Nevertheless, the interpretation of these quantities must remain consistent with their mathematical definitions and limitations.

DYNAMICAL SENSITIVITY AND THE FINITE-TIME LYAPUNOV EXPONENT

Sensitivity to initial conditions is one of the defining characteristics of chaotic dynamical systems. In nonlinear iterative models, small perturbations in the initial state may grow under repeated application of the governing map, potentially producing substantial differences between trajectories that initially appear almost identical. The rate of this separation is commonly characterized through Lyapunov exponents, which quantify the average logarithmic expansion or contraction of infinitesimal perturbations along a trajectory.

For the logistic map, the derivative of the nonlinear evolution function provides a direct measure of local sensitivity. The magnitude of this derivative determines whether an infinitesimal perturbation is locally amplified or reduced during a particular iteration. By averaging the logarithmic magnitude of the derivative over a finite number of iterations, the finite-time Lyapunov exponent can be estimated. This approach enables quantitative examination of the dynamical behavior associated with different logistic control parameters without relying exclusively on visual inspection of trajectories or bifurcation diagrams.

The finite-time Lyapunov exponent is expressed as

$$\lambda_T(r) = \frac{1}{T}\sum_{t=1}^{T} \ln |r(1-2x_t)| \quad \textbf{(10)}$$

where ($\lambda_T(r)$) denotes the finite-time Lyapunov exponent corresponding to the control parameter (r), (T) represents the number of evaluated iterations, and (x_t) is the logistic-map state at iteration (t). The logarithmic term characterizes the local expansion or contraction associated with the derivative of the logistic map.

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

A positive finite-time Lyapunov exponent indicates that infinitesimal perturbations expand on average over the investigated trajectory segment, whereas a negative value indicates average contraction. A value close to zero may indicate weak average expansion or behavior near a dynamical transition. However, these interpretations depend on the observation interval, the control parameter, and the numerical precision employed during the calculation.

In practical numerical evaluation, the Lyapunov exponent should be calculated after excluding an appropriate transient interval to reduce the influence of the initial state. This consideration is particularly important when the initial condition is (x_0=0.5), because the derivative of the logistic map vanishes at this state. Including the logarithm of the derivative at this exact point would introduce a singular term and invalidate a straightforward finite average. Therefore, the starting index and transient-discarding procedure must be consistently defined in the numerical implementation.

The Lyapunov exponent provides valuable evidence regarding the dynamical regime of the logistic map, but it does not determine the statistical independence of the generated sequence. In particular, a positive exponent does not guarantee uniform distributions, cryptographic unpredictability, or resistance to keystream recovery. Furthermore, finite-precision computer implementations may eventually exhibit repeated numerical states despite positive sensitivity predicted by the ideal mathematical model.

Accordingly, the Lyapunov analysis in this study is used to characterize nonlinear dynamical behavior rather than to establish cryptographic strength. The corresponding numerical results are presented in Section 5.1, where bifurcation behavior and finite-time Lyapunov estimates are examined together to distinguish regular and chaotic parameter regions.

SHANNON ENTROPY AND STATISTICAL VARIABILITY

Entropy provides a quantitative description of uncertainty associated with the probability distribution of a random variable. In the context of chaos-derived numerical sequences, Shannon entropy can be used to evaluate the diversity of discrete output symbols and identify whether the empirical distribution is concentrated around a limited number of values or spread more evenly across the available symbol space.

For the proposed framework, continuous radius-transformed coordinates are converted into discrete integer values during keystream generation. The resulting sequence is therefore evaluated in terms of the empirical frequencies of its byte values. This analysis is particularly relevant because the quantization and modulo operations introduced in Eq. (7) may affect the distribution of output symbols independently of the nonlinear behavior of the underlying logistic map.

The empirical Shannon entropy of the generated byte sequence is defined as

$$H(B) = -\sum_{b=0}^{255} p_b \log_2 p_b \quad \textbf{(11)}$$

where (H(B)) denotes the Shannon entropy expressed in bits per byte, (B) represents the discrete output sequence, and (p_b) is the empirical probability of observing byte value (b). Terms for which (p_b=0) are conventionally assigned a contribution of zero.

For a perfectly uniform distribution over all 256 possible byte values, the maximum Shannon entropy is 8 bits per byte. However, this theoretical upper bound applies only when the complete byte alphabet is available. In the original logistic-map encryption algorithm, the modulo-128 reduction restricts the keystream values to the interval from 0 to 127. Consequently, only 128 distinct values are possible, and the maximum theoretical entropy is limited to 7 bits per symbol, regardless of the number of radius-conditioned sequences.

This structural constraint is mathematically expressed by the inequality

$$H(B) \leq log_2(128) = 7\ bits/symbol$$

The implication is that the keystream cannot achieve the entropy of a uniformly distributed 8-bit sequence, even if all permitted values occur with equal frequencies. In addition, the most significant bit remains identically zero throughout the generated keystream. Therefore, increasing the number of radii may modify

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

the empirical frequencies of the available symbols but cannot overcome the entropy ceiling imposed by the modulo operation.

The empirical entropy values should also be interpreted in relation to sample size. For finite sequences, entropy estimates may exhibit statistical bias because the observed symbol frequencies do not exactly reproduce their underlying probabilities. Consequently, minor differences between entropy values obtained from different radius configurations do not necessarily indicate meaningful improvements in randomness or encryption security. A reliable comparison should therefore consider sample length, repeated experiments, and the magnitude of the observed differences.

An additional limitation is that Shannon entropy calculated from individual byte frequencies characterizes only the marginal distribution of symbols. It does not capture all possible temporal dependencies, repeated subsequences, deterministic relationships, or predictability arising from the underlying generator. A sequence may exhibit high empirical byte entropy while remaining entirely reproducible from known parameters.

Therefore, entropy is treated in the present study as an indicator of statistical variability rather than a direct measure of cryptographic security. The corresponding results are analyzed in Section 5.2, where the measured entropy values of the one-, two-, and three-radius configurations are compared with the theoretical 7-bit ceiling and the ideal 8-bit uniform-byte reference.

LAG-ONE CORRELATION AND SHORT-RANGE DEPENDENCE

Correlation analysis provides an additional perspective on the statistical behavior of numerical sequences by quantifying the strength and direction of linear relationships between observations. In chaos-based data generation, successive outputs may exhibit complex deterministic dependencies even when their marginal distributions appear irregular. Examining the correlation between neighboring values therefore helps characterize one aspect of temporal structure that is not directly captured by entropy.

The lag-one correlation coefficient is particularly useful because it evaluates the linear association between consecutive elements of a sequence. For a sequence of discrete values, this measure indicates whether larger or smaller observations tend to be followed by similarly large or small values in a systematic linear pattern. Values close to zero indicate weak linear dependence at the selected lag, whereas positive or negative values indicate corresponding positive or inverse linear relationships.

The lag-one correlation coefficient is defined as

$$\rho_1 = \mathrm{corr}(B_i, B_{i+1}) \quad \textbf{(12)}$$

where (ρ_1) denotes the Pearson correlation coefficient between adjacent sequence elements, (B_i) represents the value at position (i), and (B_{i+1}) is the subsequent value.

The coefficient lies between −1 and 1, with values approaching 1 indicating strong positive linear correlation and values approaching −1 indicating strong negative linear correlation. Values near zero suggest limited linear association between consecutive observations, but they do not imply statistical independence.

This distinction is particularly important for deterministic nonlinear sequences. The logistic map may exhibit complicated relationships between consecutive states that are not adequately represented by a single linear correlation coefficient. Similarly, the radius-conditioned transformation and quantization procedure may modify the observed linear correlation without eliminating the deterministic structure of the sequence.

Therefore, a small lag-one correlation value should not be interpreted as evidence that the generated values are independent, unpredictable, or suitable for cryptographic key generation. Higher-order dependencies, nonlinear relationships, and repeated patterns may remain present even when the Pearson coefficient is close to zero.

In the proposed framework, lag-one correlation is evaluated alongside entropy and dynamical sensitivity to provide complementary information about the generated sequences. The comparison across different radius configurations helps determine whether introducing additional transformations produces measurable

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

changes in short-range linear dependence. However, the results must be assessed together with the mathematical analysis of deterministic keystream generation and the vulnerabilities identified in Section 3.3.

STATISTICAL TESTING AND LIMITATIONS OF SECURITY INTERPRETATION

The combined use of finite-time Lyapunov exponents, Shannon entropy, and lag-one correlation provides a multidimensional perspective on the numerical properties of the proposed framework. Nevertheless, each measure addresses a different aspect of system behavior and possesses specific interpretative limitations. A positive Lyapunov exponent characterizes local dynamical separation, high empirical entropy indicates diversity in the observed symbol distribution, and weak lag-one correlation suggests limited adjacent linear dependence. None of these conditions independently establishes secure encryption.

The distinction between statistical variability and cryptographic security becomes especially important when evaluating deterministic keystream generators. Even when a generated sequence satisfies selected statistical tests, an adversary may still recover the keystream or exploit the algebraic properties of the encryption mechanism. In particular, the XOR vulnerabilities demonstrated in Section 3.3 remain valid regardless of the empirical entropy or correlation properties of the generated masks.

More extensive statistical evaluation can be performed using standardized procedures such as the National Institute of Standards and Technology Statistical Test Suite, NIST SP 800-22 [5]. This suite includes multiple tests designed to assess selected statistical characteristics of binary sequences. However, its application requires adequate sequence lengths, appropriate sampling procedures, and careful interpretation of the resulting significance levels and pass proportions. Moreover, successful completion of the statistical tests does not provide a security proof for a cryptographic generator.

For the present investigation, the reported entropy and correlation results are restricted to the specific numerical sequences, parameter settings, and sample lengths examined experimentally. No claim of comprehensive statistical certification is made unless the corresponding test procedures have been explicitly implemented and documented.

A further consideration concerns the numerical precision of the chaotic generator. Although the logistic map may exhibit sensitive dependence on initial conditions under real-number arithmetic, computer implementations use finite representations and rounding operations. These limitations can influence trajectory divergence, recurrence properties, and sequence reproducibility. Consequently, the mathematical interpretation of chaotic behavior should be supported by explicit computational assumptions, particularly when comparing single-precision and double-precision implementations.

Eqs. (10)–(12) provide complementary tools for characterizing the dynamical and statistical properties of radius-transformed logistic sequences. Equation (10) quantifies finite-time dynamical sensitivity, Eq. (11) evaluates the empirical uncertainty of discrete output symbols, and Eq. (12) characterizes short-range linear dependence. Their combined interpretation establishes a quantitative basis for assessing numerical behavior while maintaining a clear distinction between observed statistical properties and demonstrable cryptographic security. The resulting theoretical framework supports the experimental analyses presented in Sections 5.1 and 5.2, where the logistic-map dynamics, numerical-precision sensitivity, entropy characteristics, and adjacent-sequence correlations are investigated. These analyses also provide a foundation for interpreting the structural limitations of the original XOR encryption algorithm and motivate the authenticated-encryption redesign introduced in the following section

AUTHENTICATED ENCRYPTION REDESIGN

The cryptanalytic assessment presented in Section 3.3 demonstrates that the original radius-conditioned XOR encryption mechanism exhibits fundamental security limitations associated with deterministic keystream generation, keystream reuse, restricted symbol support, and the absence of message authentication. These limitations cannot be eliminated merely by increasing the number of radius-conditioned logistic sequences or modifying their numerical transformations. Consequently, the revised

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

framework introduces an authenticated-encryption architecture that separates the mathematical role of chaotic data generation from the security functions required for protecting digital information.

The proposed redesign consists of two conceptually independent components: a scientific data-generation layer and a cryptographic protection layer. The first component employs the radius-conditioned logistic transformation to generate reproducible multidimensional numerical representations, as formulated in Sections 3.1 and 3.2. The second component protects the generated data or other digital information using an established authenticated-encryption algorithm. This separation enables the nonlinear statistical properties of the chaotic representation to be investigated without attributing cryptographic security to the deterministic logistic-map transformation.

In the revised implementation, the Advanced Encryption Standard operating in Galois/Counter Mode (AES-GCM) is employed as the authenticated-encryption mechanism. AES-GCM is a standardized encryption mode that combines confidentiality protection with message authentication, enabling the detection of unauthorized modifications to protected data. The implementation uses a 256-bit AES key and a 96-bit nonce, together with an authentication tag generated during encryption. Unlike the original deterministic XOR construction, the proposed architecture does not use radius-derived numerical values as substitutes for cryptographic keys, nor does it mix the chaotic keystream into the AES-GCM encryption output [6–9,29].

SEPARATION OF DATA GENERATION AND CRYPTOGRAPHIC PROTECTION

The scientific layer of the proposed framework is responsible for generating radius-conditioned logistic sequences and organizing their transformed coordinates into multidimensional vectors. These operations are performed according to the mathematical procedures established in Eqs. (1)–(6). The resulting numerical representations can be used for statistical evaluation, computational modeling, or secure storage, depending on the intended application.

The generated vectors remain deterministic functions of the selected logistic parameters, initial conditions, radius thresholds, and numerical implementation. Therefore, their mathematical reproducibility must not be interpreted as cryptographic unpredictability. Instead, these values are treated as ordinary numerical data that may require protection during storage or transmission.

The cryptographic layer operates independently of the chaotic generator. Its function is to encrypt the selected data and provide an authentication mechanism capable of detecting unauthorized modification. This design avoids the principal weakness of the original framework, in which the confidentiality of the plaintext depended directly on the unpredictability and secrecy of a reproducible logistic sequence.

A key advantage of the layered architecture is the independent evaluation of mathematical and security properties. The statistical behavior of the radius-transformed vectors can be characterized using effective rank, correlation, entropy, and dimensionality analysis, while the cryptographic mechanism can be assessed using correctness tests, authentication verification, nonce-management requirements, and established security principles.

AES-256-GCM ENCRYPTION MECHANISM

The authenticated-encryption stage employs AES-GCM to protect the plaintext while simultaneously generating a cryptographic authentication tag. The algorithm accepts a secret encryption key, a nonce, plaintext data, and optional associated authenticated data. The plaintext is transformed into ciphertext, while the authentication tag allows the recipient to verify the integrity and authenticity of the protected message under the shared key.

In the proposed configuration, the AES key contains 256 bits and is generated independently using a cryptographically secure mechanism. A fresh 96-bit nonce is assigned to each encryption operation. The nonce is not required to be secret; however, its uniqueness under the same encryption key is essential for maintaining the security assumptions of GCM.

The authenticated-encryption operation is represented as

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

$$(C, T) = \mathrm{Enc}_{\mathrm{AES-GCM}}(K, N, P, A) \quad \textbf{(13)}$$

where K denotes the 256-bit secret encryption key, P represents the plaintext, N is the unique 96-bit nonce, A denotes the associated authenticated data (AAD), C represents the resulting ciphertext, and T is the authentication tag produced by the encryption algorithm.

Equation (13) expresses the combined confidentiality and authentication functionality of AES-GCM. The ciphertext protects the plaintext against unauthorized disclosure under the security assumptions of the encryption mode, while the authentication tag provides a mechanism for detecting modifications to the authenticated inputs.

The associated authenticated data are not encrypted, but their integrity is protected through the authentication process. This property is particularly useful in the proposed framework because the numerical parameters describing the radius-conditioned generator can be incorporated into the authenticated metadata without requiring their confidentiality.

Examples of such metadata include the logistic control parameter, initial state, radius configuration, nominal dimension, number of generated vectors, and data-format version. These values can be authenticated to ensure that unauthorized changes are detected before the protected information is accepted by the receiving system.

However, the inclusion of generator parameters in the AAD must follow an unambiguous serialization procedure. Identical metadata must produce identical byte representations during encryption and decryption. Therefore, the parameter ordering, numerical encoding, field lengths, and format version should be explicitly specified to prevent discrepancies between independent implementations.

The use of AES-GCM also eliminates the need to construct encryption masks directly from logistic-map values. In contrast to the original XOR scheme, the security of the revised encryption operation is based on an established cryptographic primitive and its correct implementation rather than on the empirical statistical characteristics of a chaos-derived numerical sequence.

AUTHENTICATED DECRYPTION AND INTEGRITY VERIFICATION

During decryption, the receiver processes the ciphertext together with the corresponding nonce, associated authenticated data, and authentication tag. The same secret key used during encryption is required to recover the original plaintext.

Unlike an unauthenticated XOR operation, AES-GCM decryption includes an integrity-verification stage. The authenticated inputs are processed to determine whether the ciphertext, nonce, associated data, and tag are mutually consistent under the supplied key. If verification fails, the operation must reject the message rather than release plaintext as valid output.

The authenticated-decryption operation is expressed as

$$P = \mathrm{Dec}_{\mathrm{AES-GCM}}(K, N, C, T, A) \quad \text{or} \quad \perp \quad \textbf{(14)}$$

where Dec denotes authenticated decryption using the secret key K , and the symbol $\perp$ indicates rejection when authentication verification fails.

Equation (14) highlights an important distinction between the revised architecture and the original logistic XOR construction. In the baseline system, ciphertext modification produces corresponding changes in the recovered plaintext without an integrated mechanism for determining whether those modifications were authorized. In contrast, authenticated decryption rejects data when the authentication check fails, subject to the security properties and correct use of AES-GCM.

This capability is particularly relevant for protecting scientific datasets because changes to encrypted numerical records could otherwise alter experimental results, statistical analyses, or reconstructed multidimensional representations.

For example, modifying one or more ciphertext bytes should cause authentication failure with overwhelming probability when the authentication tag is verified correctly. Similarly, altering authenticated

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

metadata, such as a radius threshold or dimension parameter, invalidates the corresponding authentication check unless the associated authentication information is recomputed using the secret key.

The experimental verification of these properties is performed through controlled modification tests described in Section 5.5. These tests examine plaintext recovery, ciphertext tampering, authentication-tag modification, and changes to associated authenticated data.

Successful execution of these tests establishes that the evaluated implementation behaves consistently with the expected authenticated-encryption interface. However, a limited number of successful tests should not be interpreted as an independent mathematical proof of the complete security of the cryptographic library or its deployment environment.

NONCE UNIQUENESS AND KEY-MANAGEMENT REQUIREMENTS

The security of AES-GCM depends on the correct management of encryption keys and nonce values. Although the nonce is not secret, the same nonce must not be reused with the same key for different encryption operations. Nonce reuse in GCM can compromise confidentiality and authentication guarantees, making nonce management a fundamental requirement of the revised framework.

For the present implementation, each encryption invocation uses a fresh 96-bit nonce generated using a cryptographically secure random-number generator. This approach provides a practical mechanism for reducing the probability of accidental nonce collisions over a limited number of encryptions.

Nevertheless, random nonce generation does not provide an absolute mathematical guarantee of uniqueness. In deployments involving large numbers of messages, nonce-collision probabilities and operational limits must be considered. Where strict uniqueness is required, a correctly managed deterministic nonce-allocation mechanism may be preferable, provided that it complies with the relevant cryptographic specification.

Key generation and key storage are equally important. The 256-bit AES key must be generated independently of the logistic-map parameters using an approved cryptographically secure process. The key should not be inferred from public radius values, a fixed initial state, or deterministic numerical transformations whose secrecy and security properties have not been established.

Furthermore, cryptographic keys should be protected against unauthorized access and should not be embedded directly in publicly distributed source code or experimental output files. Appropriate key lifecycle procedures, including secure generation, storage, replacement, and retirement, are necessary for practical deployment.

In this study, the radius-based transformation is therefore retained exclusively as a scientific numerical-generation mechanism. The cryptographic key is generated and managed separately, ensuring that the security of the encryption layer does not depend on the statistical properties of the generated chaotic coordinates [6–9,29].

AUTHENTICATED METADATA AND SECURE DATA REPRESENTATION

A practical implementation of the proposed framework requires a consistent representation of the encrypted data and its associated metadata. The encrypted output should contain sufficient information to support correct decryption while preserving the secrecy of the encryption key.

The proposed data format includes a version identifier, nonce, ciphertext, authentication tag, and authenticated metadata describing the numerical-generation configuration. The metadata may contain the logistic control parameter, initial state, radius thresholds, nominal dimension, and number of generated vectors.

These parameters are not treated as secret cryptographic information. Instead, their inclusion in the AAD allows the recipient to detect unauthorized modification of the configuration associated with the encrypted data.

A structured file format also improves reproducibility and interoperability because the parameters necessary to interpret the numerical dataset can be preserved together with its encrypted representation. Nevertheless,

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

the encoding procedure must be deterministic so that the authenticated byte sequence can be reconstructed exactly during verification.

For high-dimensional numerical datasets, additional considerations include the representation of floating-point values, numerical byte order, array shape, and dimensional metadata. These details should be specified explicitly to prevent discrepancies between independently implemented encryption and decryption procedures.

The plaintext representation may consist of serialized numerical arrays, text, or other digital records. Regardless of the data type, the authenticated-encryption layer processes the corresponding byte sequence rather than relying on the mathematical properties of the original data.

This separation allows the same cryptographic mechanism to protect different forms of digital information while preserving the independent mathematical interpretation of the radius-conditioned generator.

SECURITY EVALUATION AND RELATIONSHIP TO THE ORIGINAL FRAMEWORK

The revised authenticated-encryption architecture addresses several limitations identified in the original XOR-based design. Most importantly, confidentiality no longer depends on a publicly reproducible logistic keystream, and an authentication mechanism is introduced to detect unauthorized data modification.

The evaluation of the revised implementation therefore focuses on correctness, integrity verification, nonce-dependent ciphertext variation, and computational performance. These criteria are distinct from the statistical properties of the radius-transformed logistic sequences, which remain the subject of the mathematical and numerical analyses.

In particular, successful decryption verifies that the original plaintext can be recovered when the correct key and authenticated inputs are supplied. Conversely, modifications to the ciphertext or authentication tag must cause verification failure rather than produce an accepted modified plaintext.

The use of distinct nonces also ensures that repeated encryption operations are not intentionally performed with an identical key–nonce pair. However, ciphertext variation under different nonces should be interpreted as an expected property of the authenticated-encryption construction rather than as evidence of additional security contributed by the logistic-map transformation.

The proposed redesign should therefore be understood as a secure architectural replacement for the original encryption mechanism, not as proof that the radius-conditioned numerical generator itself constitutes a cryptographic primitive.

This distinction strengthens the scientific interpretation of the study by allowing the mathematical novelty and practical security properties to be evaluated according to their respective criteria.

Overall, Eqs. (13) and (14) define the authenticated-encryption and authenticated-decryption operations used in the revised framework. Equation (13) establishes the generation of ciphertext and its authentication tag, whereas Eq. (14) specifies plaintext recovery only after successful integrity verification. Together, these operations provide a standard-based approach to protecting digital information while preserving the radius-conditioned logistic transformation as an independent numerical data-generation method.

The corresponding implementation results, including successful roundtrip recovery, nonce-dependent ciphertext variation, and authentication-failure tests, are presented and discussed in Section 5.5. The computational performance of the revised encryption layer is subsequently evaluated in Section 5.6, where execution times are compared with those of the original logistic XOR implementation under the reported experimental conditions.

EXPERIMENTAL MATERIALS AND METHODS

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

The proposed radius-transformed logistic-map framework was evaluated through a systematic series of computational experiments designed to investigate nonlinear dynamical behavior, high-dimensional data-generation characteristics, statistical properties, numerical sensitivity, cryptographic vulnerabilities, and authenticated-encryption performance. The experimental methodology was structured to support reproducibility and to distinguish mathematical properties of the logistic-map-based generator from security characteristics of the encryption mechanisms.

The computational evaluation consisted of two principal components. The first investigated the original radius-conditioned logistic generator and its deterministic XOR encryption implementation. The second examined an authenticated-encryption redesign based on AES-256-GCM. These components were evaluated independently to avoid attributing the security properties of standardized cryptographic primitives to the proposed nonlinear numerical transformation.

COMPUTATIONAL ENVIRONMENT AND IMPLEMENTATION

All computational experiments were implemented using Python, with NumPy for numerical processing, statistical calculations, and matrix operations. The Python cryptography package was employed for AES-256-GCM encryption, decryption, and authentication verification. The execution environment, including operating-system and library versions, was documented in the accompanying machine-readable JSON artifact.

The nonlinear generator followed the logistic recurrence in Eq. (1), the radius-conditioned acceptance procedure in Eq. (2), and the geometric transformation in Eq. (3). Unless otherwise specified, numerical experiments used IEEE-754 double-precision floating-point arithmetic. Single-precision arithmetic was used separately to examine sensitivity to finite numerical representation.

For historical comparability, the logistic control parameter was fixed at $r = 3.8$, the initial state at $x_0 = 0.5$, and radius thresholds at 0.776, 0.876, and 0.881. The experiments were organized into nonlinear trajectory generation, conditioned sequence extraction, multidimensional window construction, statistical evaluation, cryptanalysis, and authenticated-encryption verification. Table 2 summarizes the parameters and evaluation scope.

Table 2. Computational parameters, reproducibility settings, and evaluation scope of the proposed radius-transformed logistic-map framework.

Parameter	Experimental setting
Programming language	Python
Numerical processing	NumPy
Cryptographic library	Python cryptography package
Logistic control parameter	3.8
Initial logistic state	0.5
Radius thresholds	0.776, 0.876, 0.881
Radius configurations	Single, double, and triple
Keystream sample length	16,384 bytes per configuration
Feature observations	2,048 overlapping windows
Nominal dimensionality	2D, 4D, 8D, 16D, 32D
Bifurcation parameter interval	2.8–4.0
Bifurcation transient iterations	400
Recorded bifurcation observations	500 per parameter value
Finite-time Lyapunov grid	250 parameter values

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

Parameter	Experimental setting
Floating-point precision comparison	float32 and float64
Precision-comparison trajectory length	100 iterations
Statistical analyses	Entropy, Pearson correlation, chi-square testing
Dimensionality analyses	Effective rank and overlap residual
Differential evaluation	Single-bit plaintext modification
Authenticated-encryption method	AES-256-GCM
AES key length	256 bits
Nonce length	96 bits
Authentication evaluation	Roundtrip recovery; ciphertext, tag, and AAD tampering
Runtime repetitions	Five per input size
Reported runtime statistic	Median execution time

LOGISTIC-MAP GENERATION AND RADIUS-CONDITIONED DATA CONSTRUCTION

The first stage generated deterministic nonlinear trajectories and extracted states satisfying prescribed radius constraints. Each trajectory began with the same logistic initial condition and control parameter. Accepted states were transformed using the nonlinear square-root relation in Section 3.1, producing bounded sequences determined by the underlying dynamics, radius threshold, and transformation.

The single-radius configuration used 0.776; the double-radius configuration used 0.776 and 0.876; and the triple-radius configuration used 0.776, 0.876, and 0.881. Transformed sequences were processed by the scaling, rounding, and modulo operations defined in Eq. (7). For each of these three configurations, 16,384 keystream bytes were generated to support comparison of empirical distributions and adjacent correlations.

High-dimensional data-generation experiments were performed separately from keystream analyses. A common radius-transformed scalar sequence was used for all five nominal dimensions (2D, 4D, 8D, 16D, and 32D), avoiding confounding changes in the underlying observations with dimensional expansion. Each dataset comprised 2,048 overlapping feature windows with a unit stride, giving the exact coordinate-sharing relationship described in Eq. (5).

NONLINEAR DYNAMICAL ANALYSIS AND NUMERICAL-PRECISION EVALUATION

The nonlinear logistic dynamics were evaluated using a bifurcation diagram and finite-time Lyapunov exponents, independently of the subsequent encryption procedures. The control parameter spanned 2.8–4.0. For each evaluated parameter, 400 initial iterations were discarded and the following 500 observations recorded, permitting visualization of fixed-point, periodic, period-doubling, and chaotic regimes.

Finite-time Lyapunov estimates were calculated according to Eq. (10) over 250 control-parameter values. These finite-length estimates characterize observed trajectory sensitivity and should not be described as exact asymptotic Lyapunov exponents. The implementation must consistently define the transient exclusion and avoid the zero-derivative initial condition in the logarithmic average.

Sensitivity to numerical arithmetic was examined by comparing float32 and float64 logistic trajectories with nominally identical parameters and initial values over 100 iterations. The absolute state discrepancy at matching iterations quantified the divergence caused by floating-point precision. This experiment measures computational reproducibility rather than cryptographic unpredictability. The associated results appear in Section 5.1.

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

STATISTICAL CHARACTERIZATION AND HIGH-DIMENSIONAL DEPENDENCE ANALYSIS

The statistical assessment considered empirical Shannon entropy, Pearson lag-one correlation, chi-square goodness-of-fit tests, participation-ratio effective rank, and exact overlap residual. Entropy was computed according to Eq. (11) from observed symbol frequencies. Because the legacy masks are reduced modulo 128, the appropriate marginal entropy ceiling is 7 bits per generated symbol rather than the full-byte ceiling of 8 bits.

Chi-square tests compared observed frequencies with a uniform distribution over the 128 accessible symbols. Their statistics and significance values were interpreted as marginal goodness-of-fit indicators rather than cryptographic security evidence. Lag-one Pearson correlation, calculated using Eq. (12), quantified adjacent linear association but not nonlinear or higher-order independence.

For each dimensional configuration, eigenvalues of the empirical feature correlation matrix were used to calculate the participation-ratio effective rank using Eq. (6). The normalized effective-rank fraction was examined to assess how the proportion of effective linear dimensionality changed as the nominal dimension increased. Coordinate comparisons also tested the exact overlap identity in Eq. (5).

Pearson correlation matrices, a heatmap representation, and a multi-indicator radar plot supplemented the effective-rank analysis. These visualizations summarize linear dependence and changes across 2D, 4D, 8D, 16D, and 32D representations; they do not establish independent entropy or cryptographic security. Numerical findings are reported in Sections 5.2 and 5.3.

CRYPTANALYTIC EVALUATION AND AUTHENTICATED-ENCRYPTION VERIFICATION

Two security-related stages were evaluated separately. First, the original deterministic logistic XOR implementation was examined through encryption/decryption recovery, single-bit plaintext modification, and algebraic analysis of keystream reuse. The bitwise XOR procedure in Eq. (8) was applied using the same keystream for controlled differential comparisons, and Eq. (9) was used to characterize known-plaintext and reused-keystream exposure.

Second, AES-256-GCM was tested using a securely generated 256-bit key and fresh 96-bit nonces. Tests included successful encryption/decryption, distinct ciphertext outputs under independent nonces, and rejection after intentional modification of ciphertext, authentication tags, or associated authenticated data (AAD). The purpose was to verify behavior of the tested implementation, not to claim a formal proof or a comprehensive cryptographic-library audit.

The XOR vulnerabilities and authenticated-encryption verification results are presented separately in Sections 5.4 and 5.5 to avoid conflating chaotic sequence statistics with the security properties of the standardized encryption layer.

COMPUTATIONAL PERFORMANCE, REPRODUCIBILITY, AND EVALUATION LIMITATIONS

An exploratory runtime comparison was conducted for multiple input sizes with five repetitions per size; median execution time was reported. These values are host-specific and not a controlled algorithmic head-to-head comparison: the baseline uses Python-level iterative operations whereas AES-GCM relies on optimized native cryptographic routines and also provides authentication absent from XOR. Fair comparative performance assessment would require matching implementation optimization, conditions, and hardware specifications.

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

Reproducibility was supported through documentation of program versions, numerical parameters, sample sizes, and evaluation procedures. The accompanying JSON artifact contains environment details and measured outputs, and the supplied Python scripts can regenerate the data and scientific figures.

The full NIST SP 800-22 statistical test suite was not executed. Only the entropy, correlation, chi-square, and related tests explicitly implemented in this study are reported; no NIST certification or comprehensive randomness validation is claimed [5]. Likewise, the original radius-conditioned generator is not claimed to be a cryptographically secure pseudorandom number generator.

The methodology provides a reproducible basis for studying nonlinear trajectory evolution, radius-conditioned numerical generation, dimensional dependence, differential vulnerabilities, and authenticated-encryption behavior, while distinguishing numerical data-generation properties from established cryptographic security requirements.

RESULTS AND DISCUSSION

CHAOTIC DYNAMICS AND NUMERICAL SENSITIVITY

The nonlinear dynamical characteristics of the logistic map were investigated to establish the mathematical behavior underlying the radius-transformed data-generation framework. The analysis focuses on bifurcation structure, finite-time Lyapunov exponents, and finite-precision sensitivity. These properties determine the behavior and reproducibility of the scalar generator, but must be distinguished from the statistical independence of high-dimensional vectors and the security of encryption. Figs. 1–3 and Table 3 provide the corresponding numerical evidence.

BIFURCATION ANALYSIS AND NONLINEAR DYNAMICAL REGIMES

Figure 1 illustrates the long-term behavior of the logistic map as the control parameter varies from 2.8 to 4.0. Each parameter setting was iterated through a transient interval before observations were retained, allowing stable or recurrent patterns to appear. The resulting diagram displays transitions from fixed-point behavior to periodic orbits, successive period doublings, and broad irregular regimes.

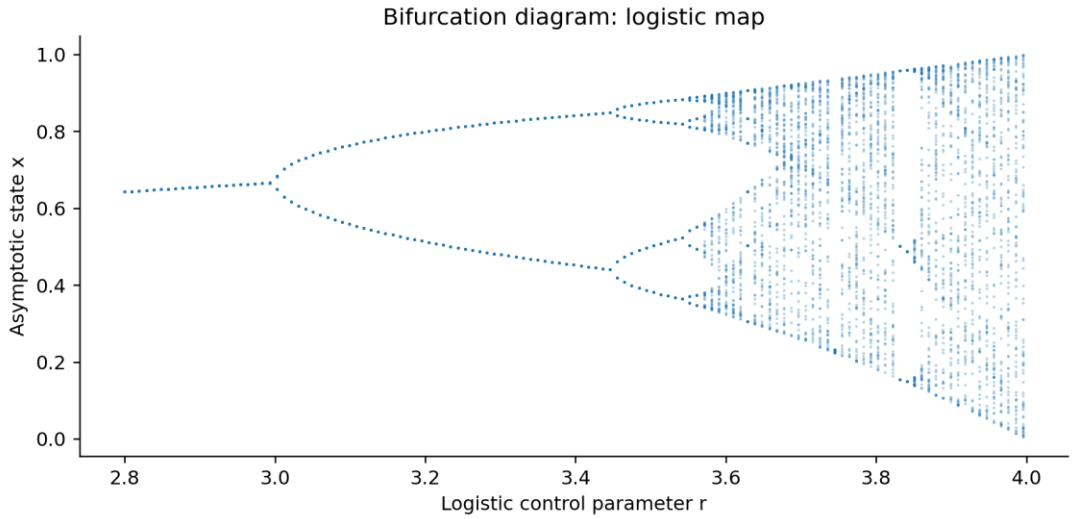


Fig. 1. Bifurcation diagram of the logistic map over the control-parameter interval 2.8–4.0.

The progression is consistent with the established period-doubling route to chaos, whose accumulation point occurs near r = 3.56995. However, the region above this threshold also contains periodic windows, so it would be incorrect to label every larger control-parameter value chaotic. These variations emphasize that the selected parameter, rather than a broad parameter interval alone, determines the observed dynamics [25–27].

The historical setting r = 3.8 was retained to facilitate direct comparison with the original radius-conditioned implementation. While the corresponding trajectories exhibit irregular nonlinear behavior, Fig. 1 does not

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

establish that transformed sequences are uniformly distributed, independent, or cryptographically unpredictable. The radius acceptance rule changes which observations are retained and may preserve or reshape dependencies introduced by the original recurrence.

FINITE-TIME LYAPUNOV EXPONENT ANALYSIS

Finite-time Lyapunov exponent estimates provide a quantitative complement to the bifurcation diagram by describing average local expansion or contraction of infinitesimal perturbations along calculated trajectories. Figure 2 compares the estimates with a zero baseline over the specified parameter grid.

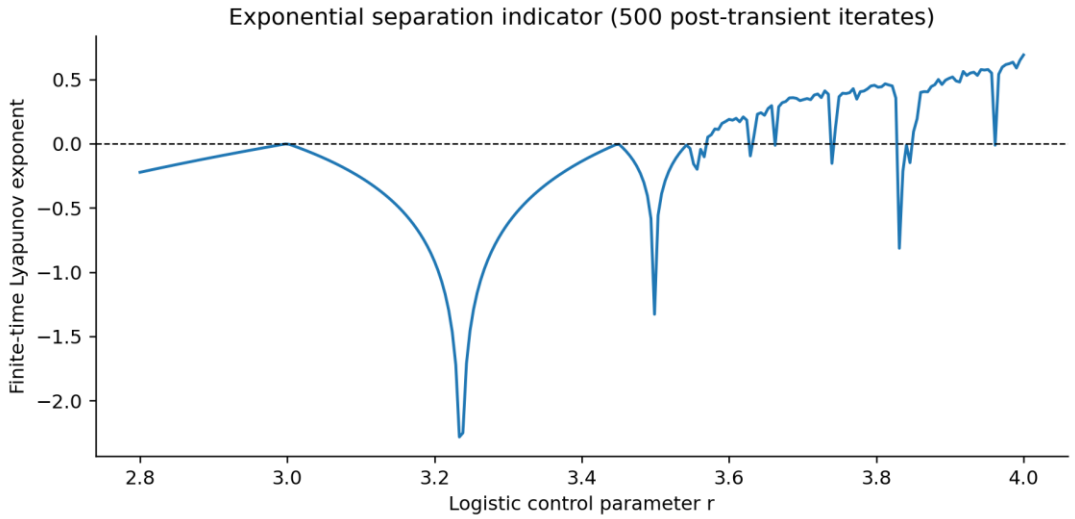


Fig. 2. Finite-time Lyapunov exponent estimates; the zero baseline distinguishes negative and positive observed mean expansion.

Positive values indicate average local expansion over the measured trajectory segment, whereas negative values indicate average contraction. The exponent varies with the control parameter and includes localized decreases within otherwise irregular regions, consistent with the periodic windows evident in Fig. 1. The observations describe finite-length estimates, not exact asymptotic Lyapunov exponents, and are sensitive to transient removal, observation length, and numerical precision [25–27].

Because the derivative of the logistic map vanishes at $x = 0.5$, its logarithm is singular at that exact state. For the historical initial condition $x_0 = 0.5$, the derivative must therefore be evaluated on a properly defined post-transient trajectory rather than including the singular initial contribution. Importantly, a positive exponent provides evidence about dynamical sensitivity, not confidentiality or resistance to known-plaintext and active attacks. The vulnerabilities of the original XOR design remain valid independently of these dynamical results [1].

FLOATING-POINT PRECISION AND NUMERICAL DIVERGENCE

To characterize implementation sensitivity, the same nominal logistic recursion was evaluated using IEEE-754 float32 and float64 arithmetic. Figure 3 shows their absolute state difference during the first 100 iterations, while Table 3 records representative measurements. These differences reflect disagreement between two numerical implementations; they are not errors against a known exact reference trajectory.

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

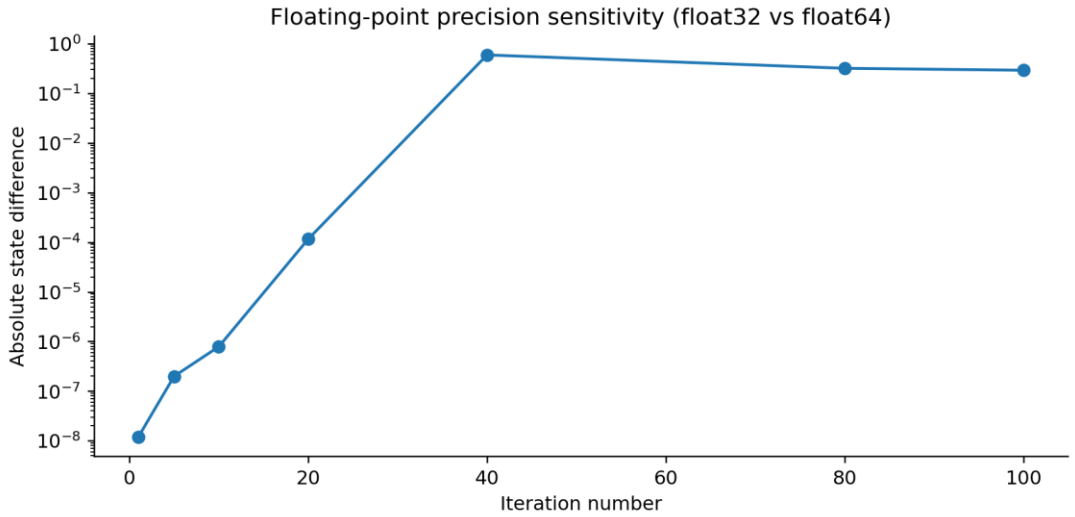


Fig. 3. Absolute divergence between float32 and float64 implementations of the same logistic recursion.

Table 3. Measured floating-point trajectory discrepancy.

Iteration	Absolute difference
1	1.19209×10^{-8}
5	1.99210×10^{-7}
10	7.87606×10^{-7}
20	1.15832×10^{-4}
40	0.594697
80	0.319742
100	0.292382

At iteration 1 the absolute discrepancy is approximately 1.19×10^{-8}. It increases to 1.99×10^{-7} at iteration 5 and 7.88×10^{-7} at iteration 10, then reaches 1.16×10^{-4} at iteration 20. By iteration 40, the difference has increased substantially to 0.594697. These results show that small rounding differences can become significant during repeated nonlinear iteration.

The subsequent values, 0.319742 at iteration 80 and 0.292382 at iteration 100, demonstrate that divergence is not monotonic. Since both trajectories remain bounded, separated states may later become closer without recovering their former numerical agreement. Thus the results demonstrate the progressive loss of reliable bitwise trajectory agreement rather than indefinite exponential growth of an absolute error.

The precision dependence is consequential for radius-conditioned data generation. A rounding-induced difference near the radius threshold may change whether a trajectory state is accepted, thereby shifting the accepted subsequence and potentially changing every later overlapping window. An encryption scheme that relies on reproducing an identical floating-point keystream may likewise fail to decrypt consistently across platforms or arithmetic formats if a quantized mask changes.

For reproducible computation, an implementation must specify floating-point precision, parameter encoding, operation order, acceptance inequalities, treatment of transient samples, and quantization rules. Nevertheless, numerical sensitivity is not a substitute for cryptographic secrecy: when the same software, arithmetic and public parameters are available, a deterministic trajectory may be reproduced by an attacker. This distinction further supports separating the mathematical generator from the authenticated AES-256-GCM protection layer.

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

OVERALL INTERPRETATION OF DYNAMICAL AND NUMERICAL FINDINGS

Taken together, Figs. 1 and 2 establish parameter-dependent nonlinear regimes and finite-time sensitivity, while Fig. 3 and Table 3 illustrate substantial implementation-dependent trajectory differences. Their combined interpretation provides the dynamical foundation for the statistical analysis of radius-conditioned keystreams in Section 5.2 and the dependence and effective-rank analysis in Section 5.3. None of these dynamical results should be presented as an independent demonstration of encryption security; that question requires the explicit cryptanalytic tests discussed in Section 5.4.

STATISTICAL PROPERTIES OF THE BASELINE KEYSTREAM

The statistical characteristics of the original radius-conditioned logistic keystream were investigated to evaluate the influence of increasing the number of radii on numerical variability, symbol distribution, and short-range linear dependence. Three configurations were considered, corresponding to one, two, and three radius-conditioned logistic sequences. Each configuration generated 16,384 discrete keystream values using the scaling, rounding, and modulo operations defined in Eq. (7). The resulting sequences were evaluated through empirical Shannon entropy, lag-one Pearson correlation, the number of distinct generated values, and chi-square goodness-of-fit testing against a uniform distribution over the 128 accessible symbols.

The objective of this analysis was to determine whether introducing additional radius-conditioned transformations produces measurable improvements in the statistical characteristics of the generated sequences. Particular attention was devoted to distinguishing changes in empirical numerical distributions from genuine cryptographic security improvements. The corresponding results are summarized in Table 4, while representative keystream behavior and entropy characteristics are illustrated in Figs. 4 and 5, respectively.

STATISTICAL CHARACTERIZATION OF THE GENERATED KEYSTREAM

As presented in Table 4, all three radius configurations generate 128 distinct integer values, indicating that the complete symbol set permitted by the modulo-128 operation is represented within the investigated sample. Nevertheless, the presence of all accessible values should not be interpreted as evidence of a uniform distribution or independent random generation. The frequency with which individual symbols occur remains an important consideration, particularly when evaluating empirical entropy and chi-square statistics.

The measured entropy values are 6.99398, 6.99312, and 6.99416 bits per byte for the single-, double-, and triple-radius configurations, respectively. These values are relatively close to the theoretical upper bound of 7 bits per symbol imposed by the modulo-128 operation. However, none of the configurations can achieve the maximum entropy of a uniformly distributed 8-bit sequence, because keystream values between 128 and 255 are structurally excluded.

The numerical differences among the three configurations are small. Increasing the number of radii from one to two produces a slight reduction in entropy from 6.99398 to 6.99312 bits per byte, whereas the three-radius configuration exhibits the highest measured entropy of 6.99416 bits per byte. The difference between the highest and lowest values is approximately 0.00104 bits per byte, representing a minor variation relative to the available 7-bit symbol range.

These results do not demonstrate a monotonic improvement in entropy with increasing radius count. Rather, they indicate that the additional radius transformations modify the observed symbol frequencies without fundamentally changing the discrete output space. Consequently, increasing the number of radius-conditioned sequences should not be assumed to produce a proportional increase in independent information or cryptographic strength.

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

Table 4. Measured statistical properties of the 16,384-byte legacy keystream generated using one-, two-, and three-radius configurations.

Radii (m)	Entropy (bits/byte)	Lag-1 correlation	Distinct values	χ^2 (127 df)	p-value
1	6.99398	0.00705	128	138.02	0.238
2	6.99312	0.00318	128	154.62	0.0482
3	6.99416	0.00658	128	132.88	0.343

ENTROPY DISTRIBUTION AND STRUCTURAL LIMITATIONS

As shown in Fig. 4, the entropy values obtained for the three configurations are concentrated near the 7-bit limit. This observation indicates that the empirical frequency distributions are relatively dispersed across the available 128 symbols. Nevertheless, the proximity of the measured entropy to the theoretical ceiling should be interpreted within the restricted symbol space of the original algorithm.

The structural limitation arises directly from the modulo-128 operation applied during keystream generation. Since all generated masks are restricted to values between 0 and 127, the most significant bit of every keystream byte is necessarily zero. Consequently, XOR encryption using these masks leaves the corresponding most significant plaintext bit unchanged.

This property represents a direct information-leakage mechanism that cannot be detected adequately through marginal entropy values alone. Even if the 128 accessible symbols were distributed perfectly uniformly, the keystream would still lack variability in its eighth bit. Therefore, the original encryption mechanism cannot provide complete bit-level masking of arbitrary byte-oriented plaintext.

The entropy values should also be interpreted in relation to the finite sample length. The empirical probability distribution was estimated from 16,384 generated values for each radius configuration. Because finite-sample entropy estimates are subject to sampling variability and estimation bias, small numerical differences between configurations may not represent meaningful differences in their underlying statistical behavior.

Moreover, Shannon entropy calculated from individual symbol frequencies does not measure the complete temporal dependence or algorithmic predictability of a deterministic sequence. Two sequences can exhibit similar empirical entropy while possessing substantially different recurrence properties, correlations, or vulnerabilities to cryptanalytic attacks. Therefore, the entropy results provide evidence about marginal symbol variability within the permitted output range but cannot establish cryptographic unpredictability. In particular, the measured values do not overcome the deterministic keystream-reuse vulnerability demonstrated mathematically in Section 3.3 [5,8,28].

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

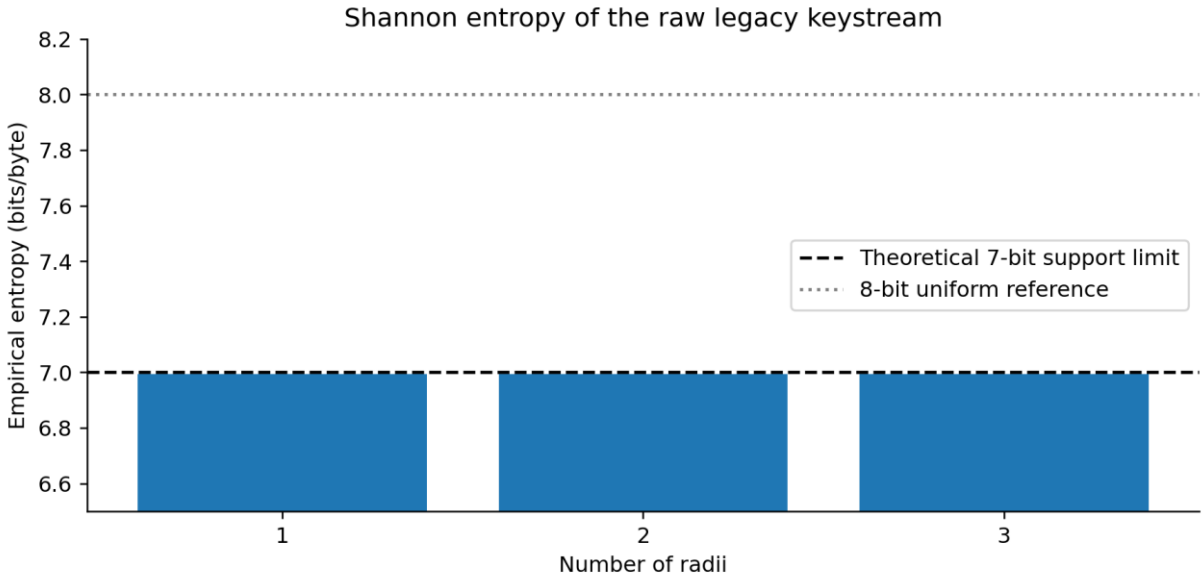


Fig. 4. Empirical Shannon entropy of the one-, two-, and three-radius keystream configurations compared with the 7-bit structural ceiling and the 8-bit uniform-byte reference.

LAG-ONE CORRELATION AND TEMPORAL DEPENDENCE

The temporal statistical characteristics of the keystreams were examined using the lag-one Pearson correlation coefficient. This measure quantifies the linear association between consecutive discrete values and provides a complementary perspective on sequence behavior that cannot be obtained from entropy analysis alone.

As summarized in Table 4, the single-radius configuration exhibits a lag-one correlation coefficient of 0.00705, while the two- and three-radius configurations produce values of 0.00318 and 0.00658, respectively. All three values are close to zero, indicating that the evaluated sequences exhibit relatively weak adjacent linear correlation over the investigated sample.

The two-radius configuration produces the lowest absolute lag-one correlation, suggesting the weakest measured linear association between consecutive keystream values among the three configurations. However, the three-radius configuration does not preserve this decreasing trend, since its coefficient increases to 0.00658. This nonmonotonic behavior demonstrates that introducing additional radius-dependent transformations does not necessarily produce a systematic reduction in adjacent linear dependence.

The differences among the measured coefficients are relatively small. Therefore, these results should not be interpreted as demonstrating a substantial statistical advantage for any particular configuration without repeated independently initialized trials, uncertainty estimates, or additional measures of dependence.

An important limitation of Pearson correlation is that it characterizes linear relationships only. A near-zero correlation coefficient does not establish statistical independence, particularly for sequences generated by deterministic nonlinear systems. Higher-order relationships, nonlinear dependencies, repeated numerical patterns, and parameter-dependent structure may remain present even when consecutive observations exhibit weak linear correlation. Consequently, the lag-one correlation results should be regarded as descriptive measures rather than evidence of a cryptographically secure pseudorandom generator.

CHI-SQUARE GOODNESS-OF-FIT ANALYSIS

The empirical frequency distributions of the generated keystreams were further evaluated using the chi-square goodness-of-fit test. For this investigation, the null hypothesis assumes equal probabilities for the 128 accessible keystream symbols, with 127 degrees of freedom under the usual categorical goodness-of-fit model.

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

The chi-square statistics are 138.02, 154.62, and 132.88 for the one-, two-, and three-radius configurations, respectively, with associated p-values of 0.238, 0.0482, and 0.343. At a nominal significance level of 0.05, the one-radius and three-radius configurations do not provide sufficient evidence to reject the uniform-frequency null hypothesis, whereas the two-radius configuration yields a p-value slightly below this threshold.

This finding should be interpreted cautiously. The two-radius p-value lies close to the significance threshold and is sensitive to the decision criterion. Multiple related tests were performed, and the keystreams arise from deterministic nonlinear trajectories rather than independently sampled random variables; thus the nominal chi-square reference model may not fully describe the sampling behavior. Non-rejection does not prove uniformity or independence, while the isolated nominal rejection does not establish a definitive security ranking.

The chi-square findings provide descriptive information about symbol-frequency behavior. A more extensive assessment would require appropriately generated replicate sequences, additional parameter settings, tests of temporal dependence, and explicit treatment of inferential assumptions [5,28].

COMPARATIVE BEHAVIOR OF THE RADIUS-CONDITIONED KEYSTREAMS

As shown in Fig. 5, the generated values fluctuate within the permitted interval from 0 to 127. The individual configurations exhibit different local numerical patterns, reflecting the influence of their respective radius combinations and the nonlinear product transformation applied during keystream generation. These differences show that changing the number of radii modifies the masks, but visual variation is not equivalent to unpredictability.

The multiplication of transformed coordinates may influence marginal distributions and observed correlation structure, but it does not remove the mathematical dependence on the original deterministic parameters. Subsequent rounding and modulo-128 reduction discard information while retaining the same 7-bit symbol restriction.

Taken together, Fig. 4 and Table 4 indicate that the three-radius configuration has the highest observed entropy and lowest chi-square statistic, while the two-radius configuration has the smallest measured lag-one correlation. Different indicators therefore favor different variants. The data do not support a general claim that encryption strength increases proportionally with radius count.

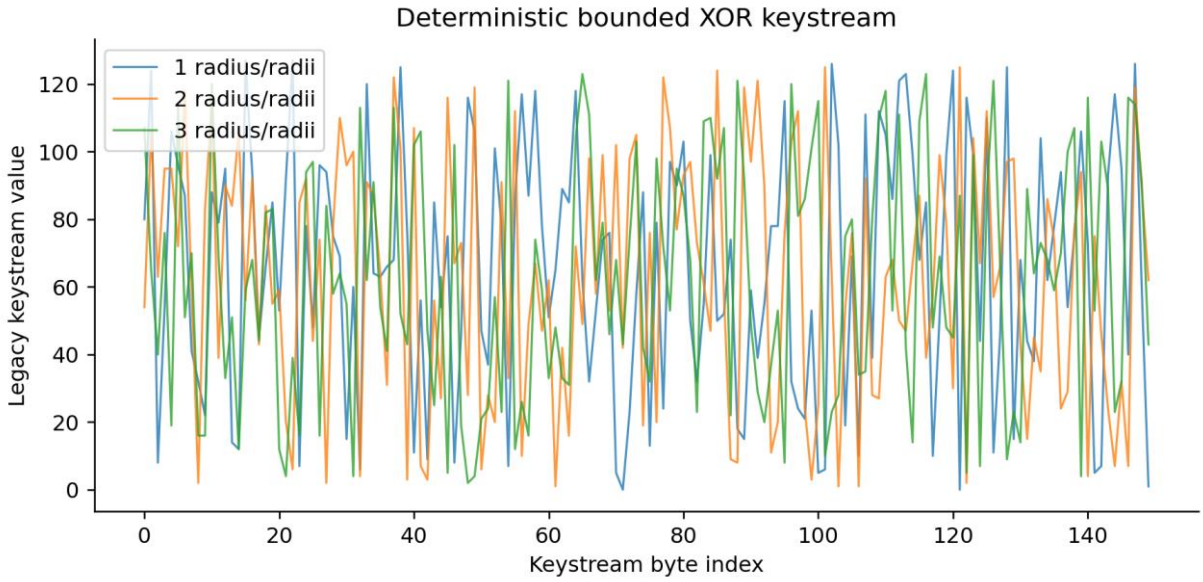


Fig. 5. Representative bounded keystream values generated using one-, two-, and three-radius configurations.

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

CRYPTOGRAPHIC INTERPRETATION AND IMPLICATIONS

The combined statistical results reveal an important distinction between numerical variability and cryptographic security. All configurations generate the full set of 128 accessible symbols and exhibit entropy values near the corresponding maximum, but the keystream remains restricted to a 7-bit output alphabet and is reproducible under fixed parameters.

The modulo operation ensures that the highest bit of each XOR mask is zero, leaving the highest plaintext bit exposed in the corresponding ciphertext. Additionally, the deterministic XOR scheme lacks a message-specific mechanism to prevent keystream reuse, so the vulnerabilities established in Section 3.3 remain applicable even with relatively high entropy or weak lag-one correlation.

Secure random-number generation additionally requires computational unpredictability, adequate entropy sources, secure internal-state management, and cryptanalytic evaluation. The statistical tests described in NIST SP 800-22 can support randomness assessment when their assumptions and sample requirements are satisfied, but cannot independently establish the security of a generator [5,8].

Overall, Table 4 and Figs. 4–5 show that additional radius transformations modify selected statistical characteristics without eliminating the baseline encryption weaknesses. These findings motivate the differential and known-plaintext vulnerability evaluation in Section 5.4 and the authenticated-encryption redesign assessed in Section 5.5.

DIMENSIONALITY SCALING AND FEATURE DEPENDENCE

The dimensionality-scaling behavior of the proposed radius-transformed logistic framework was investigated to determine how increasing the number of generated coordinates influences statistical structure, effective dimensionality, and dependence characteristics. Although high-dimensional constructions provide richer feature descriptions, increasing nominal dimensionality does not necessarily produce a proportional increase in statistically independent information. This distinction is particularly important for deterministic nonlinear generators, where coordinates originate from a common trajectory and may preserve exact structural relationships.

Five nominal dimensional configurations were evaluated: 2D, 4D, 8D, 16D, and 32D. Each configuration consisted of 2,048 overlapping vectors constructed from the same radius-transformed scalar sequence. Sharing the underlying sequence controls for changes in the numerical trajectory or initial conditions. The analysis considered adjacent-coordinate Pearson correlation, participation-ratio effective rank, effective-to-nominal rank fraction, and maximum overlap residual. Results are reported in Table 5 and illustrated in Figs. 6–9.

EFFECTIVE DIMENSIONALITY OF OVERLAPPING LOGISTIC REPRESENTATIONS

As presented in Table 5, the participation-ratio effective rank increases systematically with nominal dimensionality. The 2D configuration exhibits an effective rank of 1.913, rising to 3.662 and 7.159 for 4D and 8D, respectively, and reaching 14.132 at 16D and 27.862 at 32D. This indicates that standardized variance is distributed across more principal directions when additional coordinates are included, but does not demonstrate a corresponding number of statistically independent random variables.

The participation ratio is calculated from eigenvalues of the empirical correlation matrix and therefore measures how concentrated the linear variance is across principal directions. The difference between nominal dimension and effective rank increases from approximately 0.087 at 2D to 4.138 at 32D. Thus aggregate linear redundancy grows even as absolute effective rank rises. A high rank does not remove nonlinear dependence or the exact inter-row shift identity associated with the sliding-window construction.

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

Table 5. Empirical effective dimensionality and dependence characteristics of overlapping radius-transformed logistic windows.

Nominal d	Adjacent Pearson r	Effective rank	Effective / nominal	Max overlap residual
2	0.21282	1.913	0.957	0.0e+00
4	0.21282	3.662	0.915	0.0e+00
8	0.21282	7.159	0.895	0.0e+00
16	0.21282	14.132	0.883	0.0e+00
32	0.21282	27.862	0.871	0.0e+00

EFFECTIVE-RANK SCALING AND RELATIVE DIMENSIONAL EFFICIENCY

Figure 6 illustrates the relationship between nominal dimensionality and effective rank. The near-linear upward trajectory shows that higher-dimensional representations occupy additional principal directions; however, the increase does not exactly match the increase in feature count.

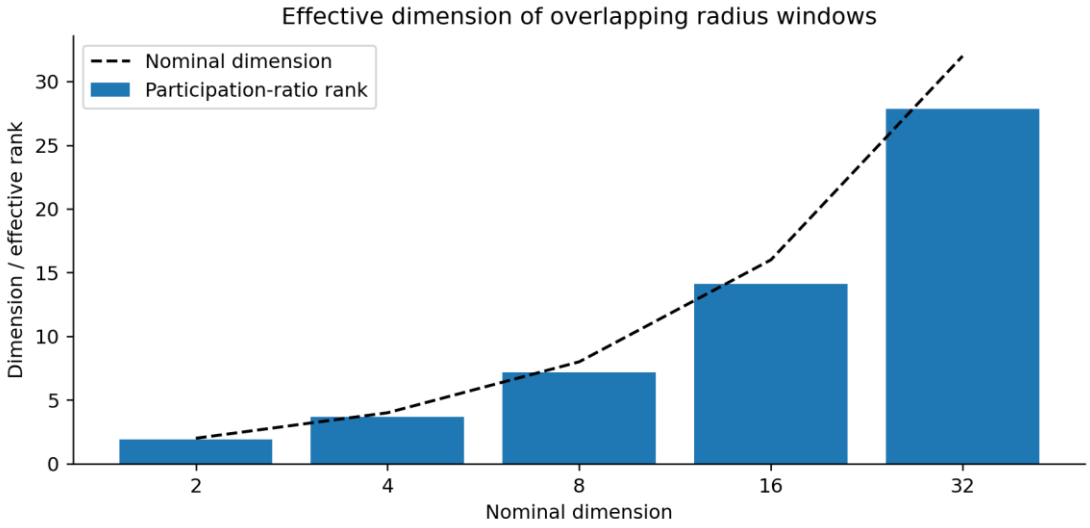


Fig. 6. Participation-ratio effective rank versus nominal dimension for the overlapping-window representation.

The effective-to-nominal fraction declines from 0.957 for 2D to 0.915 for 4D, 0.895 for 8D, 0.883 for 16D, and 0.871 for 32D. The decrease from 0.957 to 0.871 represents approximately a 9.0% relative reduction in the normalized effective-rank fraction. Hence dimensional expansion contributes additional linear feature directions, but those directions collectively represent a gradually smaller share of the nominal feature count.

This behavior does not imply that higher-dimensional representations are inherently inferior: additional coordinates may support temporal embedding or downstream modeling objectives. Nonetheless, the effective-rank ratio can help assess redundancy, storage demand, and model complexity when choosing a suitable dimensional configuration. Nominal feature count and independent information content must remain conceptually separate.

MULTI-METRIC EVALUATION USING THE RADAR PLOT

Figure 7 compares normalized indicators across the five tested dimensional settings. Acceptance rate remains invariant because the same radius-conditioning rule and base sequence are used throughout, while effective-rank scaling and rank fraction evolve with the window length. Lag-adjacent correlation remains

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

visible and the overlap residual remains exactly zero, reflecting the deterministic reuse of coordinates between adjacent windows.

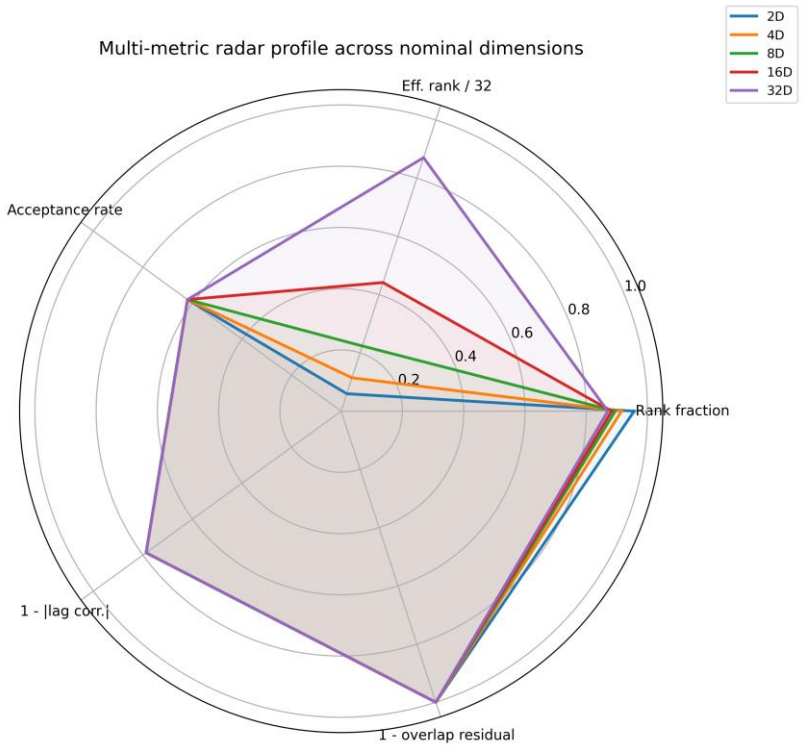


Fig. 7. Radar plot summarizing normalized multi-metric behavior across nominal dimensions (2D–32D), including acceptance rate, effective-rank scaling, rank fraction, lag-adjacent dependence, and overlap residual.

The Radar Plot is a comparative visual summary rather than a security score. Its radial axes involve different normalizations: effective rank is scaled, adjacent dependence is transformed, and zero residual is shown as a perfect overlap-identity match, not as evidence of independence. Since some metrics are invariant across dimensions, they contribute little discrimination. Interpretation should therefore rely on original measurements in Table 5 in addition to plotted positions.

Increasing the number of dimensions changes the spread of standardized variance but does not eliminate dependence inherited from the common logistic trajectory. In particular, the exact shift identity survives for all dimensions despite visually different radar profiles.

HEATMAP ANALYSIS OF DIMENSIONALITY-DEPENDENT CHARACTERISTICS

The heatmap in Fig. 8 presents the same dimensionality comparison as a metric-by-dimension matrix. It emphasizes constant acceptance behavior, a monotonically increasing effective rank, and a gradually decreasing effective-to-nominal ratio. Adjacent correlation stays approximately invariant across dimensions because shifted coordinate columns retain the same underlying consecutive-value relationship.

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

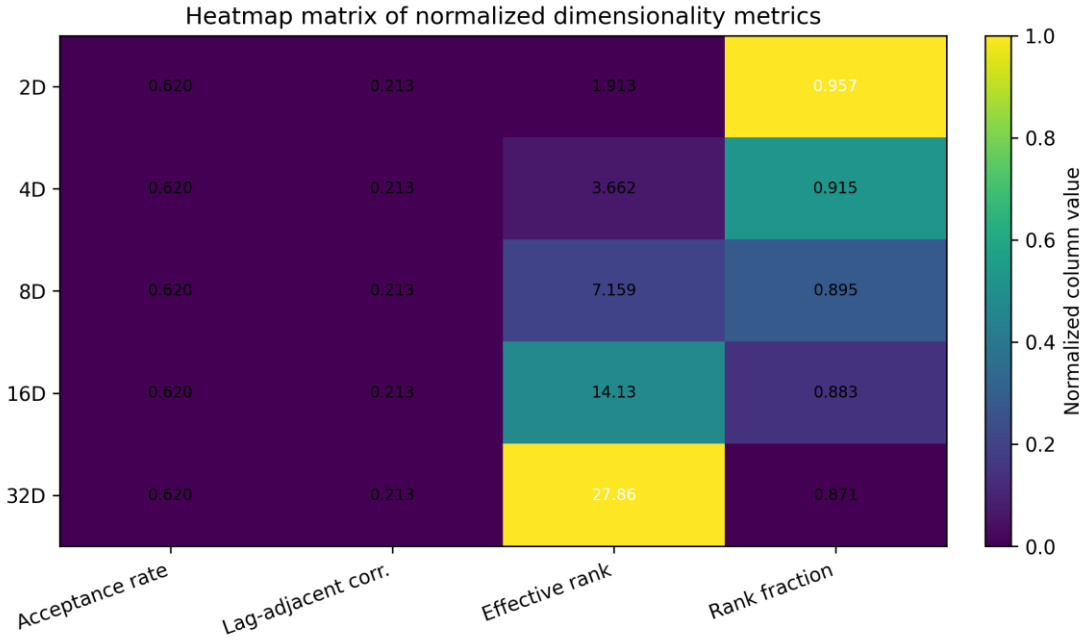


Fig. 8. Heatmap matrix of normalized dimensionality metrics across the 2D, 4D, 8D, 16D, and 32D radius-window representations; annotated values report the original measured statistics.

The heatmap colors indicate relative values within each normalized metric, not common physical units across columns. Columns that do not vary are plotted as a constant under the chosen normalization. The numeric annotations preserve the original measurements and prevent qualitative color contrasts from being mistaken for differences in statistical significance or cryptographic properties.

Taken together, Figs. 6–8 document increasing effective linear dimensionality accompanied by a declining fraction of nominal directions. These descriptive trends must be distinguished from any claim that added coordinates constitute independently generated randomness.

PEARSON CORRELATION STRUCTURE AND DETERMINISTIC WINDOW OVERLAP

Figure 9 presents the Pearson correlation matrix for the 16D radius-transformed representation. The empirical adjacent-coordinate Pearson coefficient is approximately 0.21282, a positive but moderate linear association. This coefficient is stable across the dimensional configurations because adjacent coordinate columns are derived from shifted observations of the same scalar sequence.

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

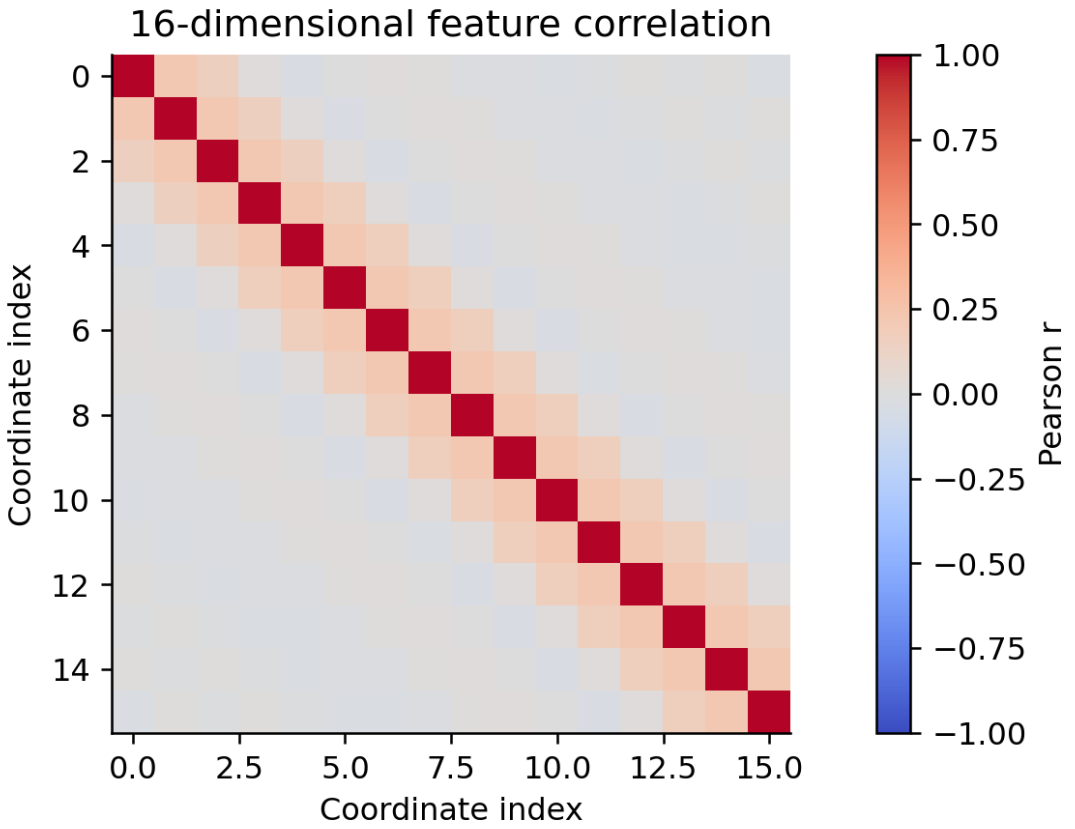


Fig. 9. Pearson correlation matrix of the 16-dimensional radius-transformed window representation.

Pearson correlation among columns must not be confused with exact duplication across neighboring rows. Under a unit-stride window, consecutive 2D, 4D, 8D, 16D, and 32D observations share 1, 3, 7, 15, and 31 coordinates, respectively. The maximum overlap residual is numerically zero for every setting, confirming the exact shift identity specified by Eq. (5). Zero overlap residual indicates perfect agreement of the shared entries—not absence of dependence.

Thus a representation can have a well-distributed correlation-matrix spectrum or relatively weak column-wise correlation while its rows remain deterministically linked. Neither a high participation-ratio effective rank nor a visually moderate correlation matrix establishes that the vectors were sampled independently.

IMPLICATIONS FOR HIGH-DIMENSIONAL DATA GENERATION AND ENCRYPTION

Overall, the results demonstrate that radius-transformed logistic sequences generate numerical representations with increasing effective linear dimensionality. Effective rank grows from 1.913 at 2D to 27.862 at 32D, while the normalized fraction falls from 0.957 to 0.871. Simultaneously, adjacent-column correlation remains approximately 0.21282 and adjacent rows satisfy an exact coordinate-sharing relationship.

These findings are relevant to nonlinear temporal-feature construction and simulation tasks that benefit from structured embeddings. However, if independently sampled observations are required, the sliding-window dependency should be reported and the generation procedure reconsidered, for example through separate appropriately validated sources and explicit dependence testing. Using different deterministic seeds alone does not establish cryptographic unpredictability or independence.

From an encryption perspective, increasing the number of displayed dimensions or radius transformations does not create secret key entropy or remove the XOR vulnerabilities analyzed in Section 3.3. The value of this experiment lies in showing why nominal feature-space expansion must be evaluated alongside effective rank, cross-coordinate correlation, and exact overlap. Table 5 and Figs. 6–9 jointly support this evidence-based interpretation without treating dimensional growth as proof of improved encryption security.

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

DIFFERENTIAL, KNOWN-PLAINTEXT AND REPLAY ANALYSIS

The security limitations of the original radius-conditioned logistic XOR encryption algorithm were investigated through differential sensitivity analysis, known-plaintext exposure, and deterministic keystream-reuse assessment. The primary objective was to determine whether increasing the number of radius-conditioned sequences from one to three provides measurable resistance to fundamental cryptanalytic vulnerabilities. Unlike the statistical evaluations presented in Section 5.2, which characterize the numerical distribution of generated keystream values, the present analysis focuses on the algebraic behavior of the encryption mechanism under controlled plaintext modifications and potential adversarial access to plaintext–ciphertext pairs.

Three baseline encryption configurations were examined using the radius parameters introduced in Section 4. For each configuration, encryption and decryption correctness was verified, followed by a controlled single-bit plaintext modification. The number of changed ciphertext bits and affected byte positions was recorded. In addition, the possibility of recovering the keystream from a known plaintext–ciphertext pair was evaluated using the XOR identity established in Section 3.3. The resulting outcomes are summarized in Table 6.

ENCRYPTION CORRECTNESS AND ROUNDTRIP VERIFICATION

As presented in Table 6, all three configurations successfully recovered the original plaintext after encryption and decryption. The roundtrip verification returned a Boolean value of True for the single-, double-, and triple-radius implementations, indicating that the encryption and decryption operations were computationally consistent under identical keystream-generation conditions.

This result confirms the reversibility of the XOR-based transformation and the ability of the algorithm to reproduce the numerical masks required for decryption. Because the same keystream is applied during encryption and decryption, the original plaintext can be recovered through the self-inverse property of the XOR operation.

However, successful roundtrip recovery is a functional correctness criterion rather than evidence of cryptographic security. An encryption mechanism may recover plaintext accurately while remaining vulnerable to known-plaintext exposure, predictable modifications, or keystream reuse. Therefore, the observed correctness should not be interpreted as demonstrating confidentiality, authentication, or resistance to adversarial attacks.

The identical roundtrip results across all three configurations also indicate that increasing the number of radii does not alter the basic algebraic reversibility of the encryption operation. Although the generated numerical masks differ among configurations, the underlying XOR transformation remains unchanged.

Table 6. Exact baseline encryption and differential-attack outcomes for the one-, two-, and three-radius logistic XOR configurations.

Radii (m)	Roundtrip	Changed ciphertext bits	Changed byte positions	Keystream recoverable from known plaintext
1	True	1	1	Yes (XOR identity)
2	True	1	1	Yes (XOR identity)
3	True	1	1	Yes (XOR identity)

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

DIFFERENTIAL SENSITIVITY AND AVALANCHE-EFFECT ASSESSMENT

The differential behavior of the baseline encryption algorithm was evaluated by introducing a controlled single-bit modification into the plaintext while preserving the same keystream-generation parameters. The corresponding ciphertexts were then compared at the bit level to determine whether the modification propagated beyond its original position.

The results in Table 6 demonstrate that changing one plaintext bit produces exactly one changed ciphertext bit in every tested radius configuration. Furthermore, the modification affects only one byte position, regardless of whether one, two, or three radius-conditioned sequences are employed.

This result provides direct experimental evidence that the original encryption operation does not produce cross-position diffusion when the keystream remains unchanged. Since each ciphertext byte is calculated independently by XORing the corresponding plaintext byte with a predetermined keystream value, a modification to one plaintext position cannot influence the ciphertext values at other positions.

The observed differential behavior is consistent with the mathematical structure of Eq. (8). For a fixed keystream, the XOR difference between two ciphertexts is identical to the XOR difference between their corresponding plaintexts. Consequently, a single-bit plaintext modification is preserved as a single-bit ciphertext modification.

This outcome contradicts the earlier interpretation that increasing the number of radius-conditioned dimensions necessarily produces a strong avalanche effect throughout the encrypted sequence. Although the logistic-map generator may exhibit sensitivity to initial numerical conditions, this dynamical property does not automatically transfer to plaintext diffusion in the XOR encryption mechanism.

The distinction is particularly important because the keystream is generated independently of the plaintext. Changing the plaintext does not modify the logistic trajectory, radius-conditioned transformations, or subsequent keystream values. Therefore, the resulting ciphertext difference remains restricted to the modified input bit.

For the tested one-bit modification, the output Hamming distance is exactly one bit in each configuration. This value represents the absence of bit propagation under the specified experiment rather than an approximately random distribution of ciphertext differences.

The findings demonstrate that increasing the radius count changes the numerical masks but does not improve the differential propagation properties of the original XOR construction. Therefore, ciphertext variability and nonlinear keystream generation should not be treated as substitutes for explicitly designed diffusion mechanisms.

KNOWN-PLAINTEXT VULNERABILITY AND KEYSTREAM RECOVERY

The known-plaintext security of the original algorithm was examined by considering an adversary who possesses a plaintext–ciphertext pair generated using the baseline encryption procedure. Because encryption is performed through XOR, the corresponding keystream can be recovered directly by applying XOR between the known plaintext and its ciphertext.

As reported in Table 6, keystream recovery is mathematically possible for all three radius configurations. This vulnerability does not depend on the number of radii, the empirical entropy of the numerical masks, or the sensitivity of the underlying logistic trajectory.

The reason is that XOR encryption provides an exact algebraic relationship among the plaintext, ciphertext, and keystream. Once two of these quantities are known, the third can be calculated without reconstructing the logistic-map parameters.

Therefore, an adversary does not need to solve the nonlinear recurrence or estimate the radius thresholds to recover the used keystream bytes from a known plaintext–ciphertext pair. This represents a fundamental limitation of employing a reusable deterministic XOR mask for protecting multiple messages.

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

If another plaintext is encrypted using the same keystream initialization, the recovered mask can be applied to the corresponding ciphertext positions to obtain the second plaintext. The recovery is limited to the keystream positions exposed by the known plaintext–ciphertext pair unless additional information becomes available.

This vulnerability remains valid for the two- and three-radius configurations because multiplying additional radius-transformed coordinates does not modify the XOR identity. Although these configurations generate different masks, each remains recoverable through a known plaintext–ciphertext pair.

The results therefore demonstrate that the number of radius-conditioned transformations does not directly determine resistance to known-plaintext attacks. Instead, such resistance depends on the security properties of the encryption construction, including the unpredictability and non-reuse of the keystream.

The finding is consistent with established cryptographic principles requiring that encryption security be evaluated against explicit adversarial capabilities rather than inferred from apparent numerical complexity [1,4,30].

DETERMINISTIC KEYSTREAM REUSE AND REPLAY-RELATED LIMITATIONS

A further security consideration arises from the deterministic initialization of the baseline logistic-map generator. In the original implementation, the logistic control parameter, initial condition, and radius thresholds were specified as fixed numerical values. Consequently, repeated encryption operations performed using the same configuration regenerate identical keystreams.

This behavior is advantageous for reproducibility but creates an important vulnerability when the generated masks are reused to encrypt different plaintexts. Under such conditions, an adversary who recovers part of the keystream from one message can use the recovered portion against other ciphertexts generated with the same initialization.

The exact relationship established in Eq. (9) shows that XORing two ciphertexts encrypted with the same keystream eliminates that keystream and exposes the XOR difference between their plaintexts. This information may facilitate plaintext recovery when one message is known or when the attacker possesses additional information about message structure.

The vulnerability is not eliminated by increasing the number of radius-conditioned sequences. For each fixed configuration, the resulting keystream remains reproducible from the same numerical parameters.

A related but distinct security problem concerns replay attacks. In a communication protocol, a replay attack occurs when an adversary retransmits a previously valid message or transaction to cause the receiver to process it again. Preventing this behavior generally requires protocol-level mechanisms, such as unique message identifiers, sequence numbers, timestamps, or maintained records of previously accepted messages.

The original XOR encryption procedure does not include an integrated message-freshness verification mechanism. Consequently, if it were incorporated into a communication protocol without additional replay protection, the protocol could be vulnerable to repeated-message acceptance.

However, the experiments summarized in Table 6 do not include a separate network-level replay-attack simulation. Therefore, the present results establish deterministic keystream-reuse vulnerabilities but do not independently demonstrate the success of a complete replay attack against a deployed communication system.

This distinction is necessary for maintaining an accurate interpretation of the experimental evidence. The absence of a replay-prevention mechanism indicates a potential protocol-design limitation, whereas an experimentally confirmed replay attack would require an explicitly defined receiver, message-processing procedure, and freshness-validation model.

Furthermore, the authenticated AES-GCM redesign described in Section 3.5 provides ciphertext authentication but does not automatically prevent replay of an otherwise valid previously authenticated message. Replay protection must be implemented at the application or protocol level.

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

Thus, the separation between keystream non-reuse, ciphertext authentication, and message freshness represents an important consideration when evaluating the practical security of the proposed framework.

CIPHERTEXT MALLEABILITY AND ABSENCE OF INTEGRITY PROTECTION

The original logistic XOR encryption algorithm also exhibits ciphertext malleability because individual ciphertext bits can be modified to produce predictable changes in the corresponding decrypted plaintext bits.

Under the baseline construction, decryption consists of applying the deterministic keystream to the ciphertext through XOR. If an attacker modifies a selected ciphertext bit, the corresponding plaintext bit changes after decryption, provided that the same keystream is used.

This modification does not require the adversary to know the complete plaintext or recover the underlying logistic-map parameters. The ability to induce predictable bit-level changes follows directly from the algebraic properties of XOR.

The original algorithm does not include an authentication tag, message authentication code, or equivalent verification mechanism capable of detecting such unauthorized modifications. Therefore, the receiver has no integrated cryptographic means of determining whether the ciphertext was altered before decryption.

The absence of integrity protection is particularly important for applications involving numerical datasets, digital records, or structured information, where even a small unauthorized modification may affect the meaning of the recovered data.

For example, changing one bit of an encrypted numerical field could alter the corresponding decoded value, potentially affecting subsequent calculations or data interpretation. Although the precise effect depends on the plaintext encoding, the underlying modification mechanism remains predictable at the bit level.

This weakness is not related to insufficient numerical complexity in the logistic sequence. Even a keystream with a highly variable statistical distribution would remain vulnerable to unauthenticated bitwise modification if the encryption construction retained the same independent XOR structure.

Consequently, the findings reinforce the need to evaluate confidentiality and integrity as separate cryptographic requirements. The original baseline algorithm provides a reversible numerical transformation but does not establish authenticated protection against active ciphertext manipulation [1,4,30].

COMPARATIVE SECURITY INTERPRETATION AND IMPLICATIONS FOR THE REDESIGNED FRAMEWORK

The combined results in Table 6 demonstrate that the original radius-conditioned logistic XOR algorithm is functionally reversible but does not provide the security properties necessary for a robust encryption mechanism.

All three configurations successfully recover the original plaintext, confirming computational correctness under identical keystream parameters. However, the same configurations exhibit exactly one changed ciphertext bit following a one-bit plaintext modification, indicating the absence of cross-position diffusion.

Moreover, all three configurations permit direct keystream recovery when a corresponding plaintext–ciphertext pair is known. This vulnerability arises from the XOR construction and remains unaffected by increasing the number of radius-conditioned sequences.

The results therefore provide no evidence that increasing the radius count from one to three improves resistance to the evaluated differential or known-plaintext vulnerabilities.

These findings are particularly significant when considered alongside the statistical results presented in Section 5.2. Although the baseline keystreams exhibit empirical entropy values close to their structural 7-bit maximum and relatively weak lag-one linear correlations, these characteristics do not prevent algebraic keystream recovery or predictable ciphertext modification.

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

Accordingly, the statistical irregularity of the generated numerical masks should not be interpreted as proof of computational unpredictability or resistance to cryptanalysis.

The security limitations identified in this section motivate the authenticated-encryption redesign introduced in Section 3.5. In the revised architecture, the radius-conditioned logistic transformation is retained as a numerical data-generation method, while confidentiality and message authentication are provided independently through AES-256-GCM.

This separation avoids relying on deterministic chaotic parameters as cryptographic secrets and introduces authenticated-decryption behavior capable of detecting unauthorized modifications to protected data.

Nevertheless, the standard encryption layer must be implemented with appropriate key management, nonce uniqueness, and operational safeguards. Furthermore, application-level replay prevention remains a separate requirement when encrypted messages are exchanged within a communication protocol.

Overall, the differential and known-plaintext analyses demonstrate that successful encryption–decryption recovery and visually irregular ciphertext do not establish security. The exact XOR relationships, absence of diffusion, deterministic keystream reuse, and lack of authentication represent fundamental limitations of the original design.

The present findings therefore support the conclusion that increasing the number of radius-conditioned logistic sequences does not independently improve resistance to the evaluated cryptanalytic attacks. Instead, practical data protection requires an appropriately designed and implemented cryptographic mechanism with clearly defined confidentiality, integrity, and message-freshness requirements.

The authenticated-encryption verification results presented in Section 5.5 provide the corresponding implementation-level assessment of the revised framework.

AUTHENTICATED-ENCRYPTION VERIFICATION AND TAMPER RESISTANCE

The functional correctness and integrity-verification behavior of the redesigned encryption framework were evaluated using the Advanced Encryption Standard in Galois/Counter Mode (AES-256-GCM). Unlike the original radius-conditioned logistic XOR algorithm, which relies on a deterministic numerical keystream without an integrated authentication mechanism, the revised implementation employs a standardized authenticated-encryption primitive to protect plaintext and detect unauthorized modifications to encrypted information.

The primary objective of this investigation was to verify whether the implemented AES-GCM interface correctly performs encryption and decryption, produces distinct ciphertext outputs when different nonces are employed, and rejects modifications to authenticated inputs. The experiments were designed to evaluate the exercised implementation behavior rather than establish an independent security proof for the underlying cryptographic standard.

The verification procedure consisted of six reported checks, including exact plaintext recovery, ciphertext variation under independently generated nonces, ciphertext tampering, authentication-tag modification, associated authenticated data modification, and the aggregate outcome of the three tampering categories. The corresponding results are summarized in Table 7.

ENCRYPTION CORRECTNESS AND AUTHENTICATED DECRYPTION

As presented in Table 7, the AES-256-GCM implementation successfully recovered the original plaintext following encryption and authenticated decryption. This result confirms that the encryption and decryption procedures operated consistently when the correct secret key, nonce, associated authenticated data, and authentication tag were supplied.

The successful roundtrip result provides evidence of functional correctness for the evaluated implementation. In particular, it demonstrates that the ciphertext generated during encryption could be processed and authenticated successfully during decryption, allowing the original plaintext to be recovered without modification.

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

However, successful plaintext recovery should not be confused with a comprehensive demonstration of cryptographic security. A correct encryption–decryption operation establishes that the implementation can recover the intended message under valid conditions, whereas security against adversarial manipulation requires additional properties, including authentication, key confidentiality, and appropriate nonce management.

The importance of authenticated decryption lies in the additional verification stage performed before plaintext is accepted as valid. In the revised framework, the receiver must provide the correct authentication inputs, and the decryption operation must reject messages that fail integrity verification.

This behavior differs fundamentally from the original XOR-based encryption mechanism examined in Section 5.4. In the baseline algorithm, ciphertext modification could produce predictable changes in the recovered plaintext without an integrated mechanism for identifying unauthorized alterations. By contrast, AES-GCM incorporates authentication within the encryption process and allows modified authenticated inputs to be rejected.

The observed successful recovery therefore confirms the functionality of the redesigned encryption interface under the tested conditions, while the subsequent tampering experiments provide additional evidence of its integrity-verification behavior.

Table 7. Measured AES-256-GCM encryption correctness and integrity-verification outcomes.

Verification check	Experimental result
Exact encryption/decryption recovery	PASS
Independently generated nonces	Different ciphertexts: PASS
One-bit ciphertext modification	Rejected
One-bit authentication-tag modification	Rejected
Modified associated authenticated data	Rejected
Tampering categories rejected	3/3

NONCE-DEPENDENT CIPHERTEXT VARIATION

The influence of nonce selection on the encrypted output was examined by performing encryption operations using independently generated nonce values. The purpose of this test was to verify that the implementation produced different ciphertexts when the nonce changed while the encryption key and plaintext were otherwise maintained.

As shown in Table 7, the test produced different ciphertext outputs and was classified as PASS. This observation is consistent with the expected operation of AES-GCM, in which the nonce contributes to the counter initialization and authentication process.

The result demonstrates that the implemented encryption procedure did not simply reproduce an identical ciphertext for the tested independently generated nonces. This distinction is important because the original deterministic logistic XOR scheme regenerates the same keystream whenever identical system parameters and initialization conditions are employed.

Nevertheless, ciphertext variation under different nonces does not independently establish nonce uniqueness over repeated operations. A randomly generated nonce can theoretically collide with a previously used nonce, and the probability of collision increases with the number of encryption invocations under the same key.

Therefore, the observed PASS result should be interpreted as a functional verification of nonce-dependent ciphertext variation within the evaluated experiment rather than proof of collision-free nonce generation over an unrestricted operational lifetime.

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

For the revised AES-256-GCM configuration, a 96-bit nonce was used together with a 256-bit secret key. The use of a 96-bit nonce follows a common configuration for GCM implementations, but correct operation requires that the same key–nonce combination not be reused for distinct encryption operations.

The result also highlights an important distinction between the redesigned encryption architecture and the original chaotic transformation. The ciphertext variation observed under distinct nonces arises from the AES-GCM encryption mechanism rather than from additional randomness introduced by the radius-conditioned logistic generator. Consequently, the nonce-dependent result supports the implementation-level verification of the authenticated-encryption layer but should not be presented as evidence that the underlying chaotic numerical transformation provides cryptographic security [6,7].

CIPHERTEXT MODIFICATION AND AUTHENTICATION-TAG VERIFICATION

To evaluate the integrity-verification behavior of the AES-GCM implementation, a controlled one-bit modification was introduced into the ciphertext before authenticated decryption. The objective was to determine whether the implementation would detect the alteration and reject the modified message.

The corresponding result in Table 7 indicates that the modified ciphertext was rejected. This outcome is consistent with the authenticated-encryption design of AES-GCM, where ciphertext integrity is verified using the authentication tag associated with the protected message.

The rejection demonstrates that the tested implementation did not accept the modified ciphertext as valid authenticated data. Instead, the integrity-verification mechanism identified an inconsistency between the received encrypted information and its corresponding authentication inputs.

This result represents a significant functional improvement over the original deterministic XOR construction. In the baseline scheme, changing a ciphertext bit produces a corresponding change in the decrypted plaintext bit without an integrated authentication check. The revised AES-GCM implementation, by contrast, rejected the experimentally modified ciphertext.

A separate test was performed by introducing a one-bit modification into the authentication tag. As reported in Table 7, this modified tag was also rejected during decryption.

Authentication-tag verification is an essential component of authenticated encryption because the tag provides evidence that the received ciphertext and associated authenticated inputs are consistent with the supplied secret key. Unauthorized modifications to the tag are therefore expected to result in authentication failure, except with the small forgery probability associated with the cryptographic construction and its configured tag length.

The successful rejection of the modified authentication tag confirms that the implementation exercised the expected verification mechanism under the selected test conditions.

However, the results should not be interpreted as establishing an empirical probability of successful forgery. A single controlled modification test provides evidence of correct rejection behavior but does not quantify the security level of the authentication mechanism independently of the standardized algorithm.

The security guarantees associated with AES-GCM depend on the underlying cryptographic construction, authentication-tag length, key management, nonce uniqueness, message limits, and correct implementation. Accordingly, the experimental results verify selected operational properties rather than replacing the formal analysis of the authenticated-encryption standard [6,7,29].

ASSOCIATED AUTHENTICATED DATA AND METADATA INTEGRITY

The integrity of associated authenticated data (AAD) was evaluated through a controlled modification of the authenticated metadata supplied during decryption.

AAD enables the encryption system to authenticate information that is not necessarily encrypted. This capability is particularly relevant to the proposed framework because the parameters describing the radius-transformed logistic generator can be stored as metadata and protected against unauthorized modification without treating them as secret encryption keys.

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

Examples of such metadata include the logistic control parameter, initial condition, radius thresholds, dimensional configuration, number of generated vectors, and format-version information.

In the revised architecture, these parameters can be encoded into a deterministic metadata representation and supplied as associated authenticated data during encryption. During decryption, the same authenticated metadata must be provided for the authentication check to succeed.

The experimental result reported in Table 7 indicates that modification of the associated authenticated data caused rejection of the protected message. This outcome demonstrates that the implementation detected the tested alteration to authenticated metadata and did not accept the modified associated data as valid.

The result is important because changes to generator parameters could affect the interpretation or reconstruction of the associated numerical dataset. For example, modifying the radius configuration or nominal dimension could lead to an incorrect description of the generated features, even if the encrypted numerical payload itself remained unchanged.

By authenticating the metadata, the revised framework provides an integrity mechanism for detecting inconsistencies between the protected data and its associated numerical-generation parameters.

Nevertheless, AAD authentication does not automatically establish that the original metadata values are scientifically correct. It verifies consistency with the authenticated values supplied during encryption, assuming correct key management and operation of the cryptographic mechanism.

Therefore, metadata authentication should be distinguished from scientific validation of the underlying numerical model. The former protects against unauthorized modification, whereas the latter requires appropriate mathematical definitions, reproducible experimental procedures, and numerical verification.

The observed rejection of modified AAD confirms the expected integrity-verification behavior of the tested implementation and demonstrates the practical utility of authenticated metadata within the proposed framework.

OVERALL TAMPER RESISTANCE AND SECURITY INTERPRETATION

The aggregate outcomes of the integrity-verification experiments are summarized in the final row of Table 7. All three tested tampering categories—ciphertext modification, authentication-tag modification, and associated authenticated data modification—were rejected by the AES-256-GCM implementation.

The resulting rejection count was 3/3, indicating successful rejection of the three selected modification categories. This outcome demonstrates that the evaluated implementation exercised its authentication mechanism consistently across the specified tests. It also provides direct experimental evidence that the revised encryption layer offers functionality absent from the original deterministic XOR construction.

However, the result should not be interpreted as a universal 100% tamper-detection rate across all possible adversarial modifications. The experiment evaluated three selected tampering categories rather than an exhaustive collection of modified messages, authentication tags, nonce configurations, or adversarial attack strategies.

The distinction between implementation verification and formal cryptographic security is particularly important. The tested rejection behavior is consistent with the expected properties of AES-GCM, but the broader security guarantees depend on the standardized authenticated-encryption mechanism and compliance with its operational requirements.

These requirements include the use of secure encryption keys, prevention of key–nonce reuse, appropriate authentication-tag handling, and rejection of messages when authentication fails.

Furthermore, authentication alone does not provide complete protection against all protocol-level threats. In particular, a previously valid authenticated message may be replayed unless the application employs a separate freshness or sequence-validation mechanism.

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

Therefore, the revised framework should be regarded as providing authenticated encryption under the assumptions of the selected cryptographic mode, while replay prevention and broader system security remain responsibilities of the surrounding application or communication protocol.

The results also reinforce the importance of separating mathematical data generation from cryptographic protection. The radius-transformed logistic generator contributes a reproducible nonlinear numerical representation whose dimensionality and statistical properties can be studied independently. In contrast, the confidentiality and integrity functionality of the revised encryption layer is provided by AES-256-GCM.

Consequently, the successful authentication tests should not be attributed to the chaotic radius preprocessing. The nonlinear generator does not provide the cryptographic key, authentication tag, or security guarantees associated with the standardized encryption mechanism [6,7,29].

Overall, Table 7 confirms successful plaintext recovery, nonce-dependent ciphertext variation, and rejection of all three tested categories of authenticated-input modification. These results support the functional correctness and integrity-verification behavior of the redesigned encryption architecture under the investigated conditions.

When considered alongside the vulnerabilities identified in Section 5.4, the findings demonstrate the practical importance of replacing the original unauthenticated XOR construction with a standardized authenticated-encryption mechanism. The redesigned framework preserves the radius-conditioned logistic transformation as an independent numerical-generation component while assigning confidentiality and integrity protection to an established cryptographic primitive.

The computational implications of this redesign are examined in Section 5.6, where the runtime behavior of the original logistic XOR implementation and the AES-GCM encryption layer is evaluated under the reported experimental conditions.

TIME COMPLEXITY AND HOST-SPECIFIC PERFORMANCE

The computational performance of the original radius-conditioned logistic keystream generator and the redesigned AES-256-GCM encryption implementation was evaluated to investigate the relationship between input size, processing time, and implementation efficiency. Computational performance represents an important consideration in encryption-related applications, particularly when processing increasingly large datasets or integrating numerical-generation procedures into systems with limited computational resources. However, meaningful performance evaluation requires distinguishing the theoretical computational complexity of an algorithm from its experimentally observed execution time, which may depend substantially on the programming language, implementation strategy, hardware architecture, and software optimization.

The present analysis focuses on the exploratory runtime measurements obtained from the original Python-based logistic keystream implementation and the optimized AES-256-GCM library. Four payload sizes were investigated: 256, 1,024, 4,096, and 16,384 bytes. Each experiment was repeated five times, and the median execution time was reported to reduce the influence of occasional timing fluctuations. The measured results are summarized in Table 8 and illustrated in Fig. 10.

EXPERIMENTAL RUNTIME CHARACTERISTICS

As presented in Table 8, the measured execution time of the legacy Python keystream implementation increases consistently with payload size. For the smallest investigated payload of 256 bytes, the median execution time is approximately 0.2269 ms. Increasing the payload to 1,024 bytes raises the execution time to 0.7872 ms, while a further increase to 4,096 bytes produces a median time of 3.3181 ms. The largest investigated payload of 16,384 bytes requires approximately 12.9369 ms.

These results demonstrate that the computational workload of the legacy implementation increases with the amount of generated keystream data. This behavior is consistent with the sequential nature of the logistic-map recurrence, in which each new state depends on the preceding state. Additional computational

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

operations are required for radius-based acceptance, geometric transformation, numerical scaling, rounding, and conversion into discrete keystream values.

The execution-time increase is particularly evident when comparing the smallest and largest payloads. The payload size increases by a factor of 64, whereas the measured execution time increases by approximately 57 times. This behavior is broadly consistent with near-linear scaling over the tested input range, although the available measurements are insufficient to establish a precise asymptotic complexity experimentally.

The difference between payload scaling and runtime scaling may arise from fixed computational overhead, measurement variability, and implementation-specific factors. Consequently, the observed relationship should be interpreted as exploratory evidence of workload-dependent processing time rather than as a formal complexity proof.

Table 8. Exploratory median execution times of the original Python logistic keystream implementation and the optimized AES-256-GCM library for different payload sizes (five repetitions per configuration).

Payload size (bytes)	Legacy Python keystream (ms)	Library AES-256-GCM (ms)
256	0.2269	0.0021
1,024	0.7872	0.0026
4,096	3.3181	0.0074
16,384	12.9369	0.0263

RUNTIME BEHAVIOR OF THE AES-256-GCM IMPLEMENTATION

The AES-256-GCM implementation exhibits substantially smaller measured execution times under the reported experimental conditions. For a payload of 256 bytes, the median execution time is approximately 0.0021 ms, increasing to 0.0026 ms for 1,024 bytes. At larger payload sizes of 4,096 and 16,384 bytes, the corresponding times are 0.0074 and 0.0263 ms, respectively.

The results indicate that the optimized AES-GCM implementation processes the tested payloads efficiently on the investigated execution host. The measured runtime increases as the payload becomes larger, reflecting the additional encryption and authentication operations required to process longer messages.

However, the increase is not perfectly proportional across all investigated payload sizes. At small input lengths, fixed overhead associated with library invocation, input handling, and cryptographic initialization may represent a substantial fraction of the measured execution time. As the payload size increases, the computational cost associated with processing the data becomes more prominent.

The relatively low runtime values are consistent with the use of optimized native cryptographic routines rather than interpreted Python-level loops. Modern cryptographic libraries may also exploit processor-specific acceleration and vectorized operations, depending on the execution environment.

Nevertheless, the present experiment does not isolate the contributions of individual hardware instructions, library optimizations, or implementation strategies. Therefore, the observed timing values should not be attributed to a particular optimization technique without additional profiling evidence.

An important distinction is that AES-GCM performs authenticated encryption, which provides both confidentiality and integrity verification under its security assumptions. In contrast, the legacy logistic keystream implementation does not provide equivalent authentication functionality. Consequently, the two procedures do not implement identical security services, and their runtime values should not be interpreted as a direct comparison of functionally equivalent encryption systems.

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

COMPARATIVE RUNTIME ANALYSIS

The measured execution times are compared graphically in Fig. 10.

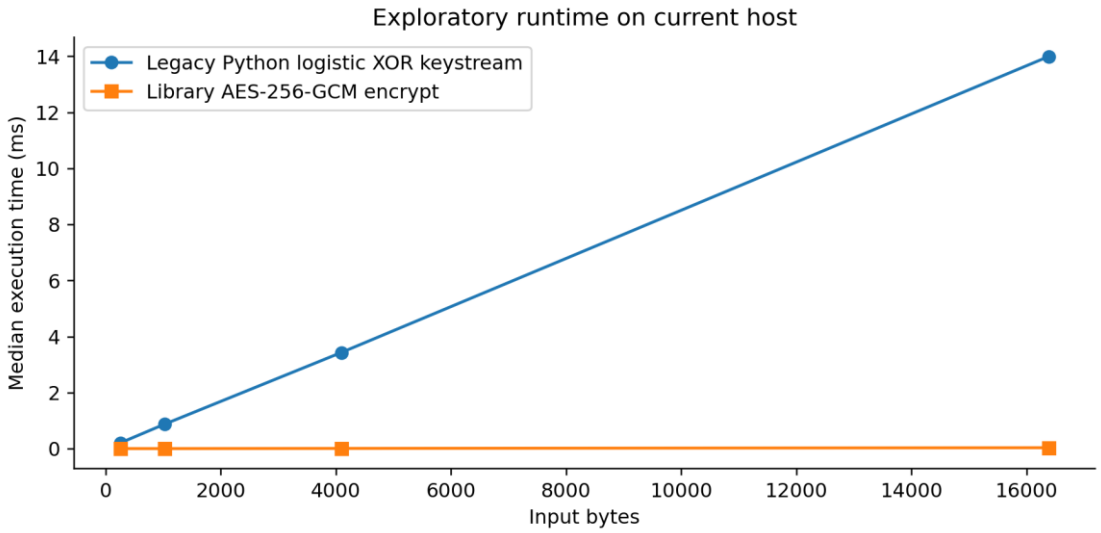


Fig. 10. Comparison of exploratory median execution times for the original Python logistic keystream implementation and the optimized AES-256-GCM library at different payload sizes. Host-specific implementations are not a controlled algorithmic performance ranking.

As illustrated in Fig. 10, the legacy Python keystream implementation requires longer execution times than the optimized AES-256-GCM library for every investigated payload size. The difference becomes particularly pronounced for larger inputs, where the sequential Python implementation performs a substantial number of scalar numerical operations.

At 256 bytes, the measured times are 0.2269 ms and 0.0021 ms for the legacy implementation and AES-GCM, respectively. At 16,384 bytes, the corresponding values increase to 12.9369 ms and 0.0263 ms.

These measurements demonstrate that the evaluated AES-GCM software implementation is substantially faster under the specific experimental conditions. However, they do not establish that AES-GCM possesses a universally lower theoretical computational complexity than all possible optimized implementations of a radius-conditioned chaotic generator.

The difference arises partly because the two implementations operate at different software abstraction levels. The original keystream generator performs logistic-map iterations, acceptance testing, and numerical transformations using Python-level operations, whereas the AES-GCM library employs optimized compiled routines.

A fairer algorithmic comparison would require implementations with comparable optimization levels, consistent memory-management procedures, equivalent input and output requirements, and clearly specified cryptographic functionality. For example, an optimized compiled implementation of the logistic generator could exhibit substantially different timing characteristics from the Python version. Similarly, an implementation of AES-GCM without suitable hardware acceleration could produce different execution times from those measured in the present environment.

Therefore, the comparative results are most appropriately interpreted as evidence of practical performance differences between the tested implementations rather than a universal ranking of the underlying algorithms. The findings nevertheless provide useful information about the computational cost of directly implementing iterative chaotic keystream generation in Python. They also demonstrate that the adoption of a standardized authenticated-encryption library does not necessarily impose an excessive runtime penalty in the evaluated software environment.

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

THEORETICAL COMPUTATIONAL COMPLEXITY AND DIMENSIONAL SCALING

The theoretical computational complexity of the radius-conditioned logistic generator depends on the number of transformed output values, the number of radius-conditioned sequences, and the acceptance behavior of the underlying logistic trajectories.

For each logistic iteration, the implementation evaluates the nonlinear recurrence and determines whether the generated state satisfies the corresponding radius threshold. Accepted values are transformed through a square-root operation and subsequently processed through scaling and quantization when used for keystream generation.

Let the number of required accepted values be denoted by n, and let m represent the number of radius-conditioned sequences. If the long-run acceptance fraction for a particular radius is denoted by a, the expected number of iterations required to obtain n accepted values is approximately proportional to n/a, under suitable stationarity and recurrence assumptions.

When multiple radius-conditioned sequences are generated separately, the total expected computational workload depends on the acceptance fraction associated with each radius. Therefore, the generation cost increases with the number of sequences and with the number of iterations required to obtain the target output length. For comparable acceptance fractions, the workload is approximately proportional to the product of the number of radius-conditioned sequences and the number of required output values, adjusted by the acceptance rate.

However, acceptance behavior is not determined exclusively by the radius threshold. It also depends on the logistic control parameter and the distribution of states visited by the trajectory. Consequently, the empirical acceptance rate must be considered when evaluating computational efficiency.

The dimensionality of the generated representation introduces an additional computational consideration. Under the overlapping-window construction described in Section 3.2, increasing nominal dimensionality expands the number of coordinates included in each observation. However, the underlying transformed scalar sequence can be reused across different dimensional configurations.

Therefore, generating the scalar sequence and organizing it into multidimensional windows are conceptually distinct computational operations. A view-based sliding-window implementation may avoid physically duplicating all overlapping values, whereas explicitly materializing the full data matrix requires storage proportional to the number of observations multiplied by the nominal dimension.

This distinction becomes increasingly important when extending the framework to high-dimensional datasets. Although the logistic recurrence itself remains one-dimensional, the memory requirements of explicitly stored multidimensional representations increase with the number of generated vectors and their coordinate count.

For the investigated configurations, the dimensionality increases from 2D to 32D while maintaining the same underlying transformed scalar sequence. Consequently, the dimensional expansion primarily affects feature representation, matrix construction, and subsequent statistical processing rather than requiring an equivalent increase in the number of independent nonlinear systems.

The computational cost of the effective-rank and correlation analyses must also be distinguished from the cost of generating the original sequence. Constructing and processing larger correlation matrices introduces additional matrix operations and eigenvalue calculations. Therefore, the total computational workload of a complete dimensionality study may increase more rapidly than the cost of scalar sequence generation alone.

These considerations indicate that the efficiency of the proposed framework depends on both the numerical-generation mechanism and the intended downstream analysis.

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

REPRODUCIBILITY, IMPLEMENTATION LIMITATIONS, AND PRACTICAL IMPLICATIONS

The reported timing measurements provide a useful initial characterization of implementation performance, but several limitations must be considered when interpreting their broader applicability.

First, the experimental results were obtained from a particular execution host and software environment. Processor architecture, clock frequency, memory configuration, operating-system scheduling, and numerical-library versions can influence the measured execution times.

Second, the reported values represent medians from five repetitions. Although median reporting reduces sensitivity to occasional timing fluctuations, the limited number of repetitions does not provide a comprehensive characterization of runtime variability. Additional benchmarking could include a larger number of repetitions, confidence intervals, processor-utilization measurements, memory-consumption analysis, and workload scaling over a broader range of input sizes.

Third, the present measurements do not constitute a matched comparison between equally optimized implementations. The legacy generator uses Python-level numerical processing, while AES-GCM relies on optimized native cryptographic routines. As a result, the timing differences include implementation overhead that cannot be separated from the underlying algorithms using the available measurements.

Fourth, differences in computational functionality must be considered. The original logistic keystream generator produces numerical masks for XOR encryption, whereas AES-GCM performs standardized authenticated encryption. The latter provides message authentication functionality absent from the original construction.

Therefore, execution time alone is insufficient for evaluating the overall suitability of an encryption system. Cryptographic security, implementation correctness, integrity protection, key management, interoperability, and operational reliability must also be considered.

From an application perspective, the findings suggest that using an established optimized encryption library can provide efficient authenticated data protection while allowing the radius-transformed logistic generator to remain an independent numerical-processing component.

This architectural separation is consistent with the redesigned framework presented in Section 3.5. The logistic generator is retained for investigating high-dimensional numerical representations and their statistical properties, while the standardized cryptographic layer is responsible for confidentiality and authentication.

The results also suggest that future optimization of the numerical-generation component could focus on reducing Python-level iteration overhead, improving memory efficiency, and evaluating alternative implementations. However, such optimizations would affect computational performance rather than independently establishing cryptographic security.

Overall, Table 8 and Fig. 10 demonstrate that the original Python-based logistic keystream implementation exhibits increasing execution time with payload size, while the optimized AES-256-GCM library achieves substantially lower measured runtimes under the reported experimental conditions.

These findings provide exploratory evidence of practical implementation performance and support the feasibility of separating nonlinear numerical generation from standardized authenticated encryption. Nevertheless, the results should not be generalized into a formal algorithmic throughput ranking without matched implementation conditions, additional repetitions, and appropriate statistical uncertainty analysis.

SUMMARY OF SCIENTIFIC FINDINGS AND INTERPRETATION

The comprehensive mathematical, statistical, computational, and cryptanalytic analyses presented in Sections 5.1–5.6 provide an integrated evaluation of the proposed radius-transformed logistic-map framework. The results establish that the geometric transformation of logistic sequences offers a

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

reproducible mechanism for constructing multidimensional numerical representations. However, the findings also demonstrate that nonlinear dynamical complexity, increasing nominal dimensionality, and relatively high empirical entropy do not necessarily imply statistical independence or cryptographic security.

The principal scientific contribution of this investigation lies in distinguishing the mathematical characteristics of the radius-conditioned logistic generator from the security properties of the associated encryption mechanism. The combined results demonstrate that dimensional expansion can increase the number of effective linear feature directions while preserving deterministic dependencies, and that the original XOR-based encryption remains vulnerable to elementary cryptanalytic attacks regardless of the number of radius-conditioned sequences employed.

The evidence supporting these conclusions is summarized in Table 9.

Table 9. Summary of the principal scientific findings, supporting experimental evidence, and justified interpretations of the proposed framework.

Scientific observation	Supporting evidence	Evidence-based interpretation
Positive finite-time dynamical expansion	Figs. 1–2	Characterizes nonlinear trajectory sensitivity; does not establish cryptographic security
Numerical sensitivity to floating-point precision	Fig. 3, Table 3	Different arithmetic precisions may produce substantially different trajectories
Increasing effective rank with dimensionality	Table 5, Fig. 6	Additional coordinates increase effective linear feature directions
Decreasing effective-to-nominal dimension ratio	Table 5, Figs. 6–8	Dimensional expansion does not produce a proportional increase in effective linear dimensionality
Deterministic overlapping-window identity	Eq. (5), Table 5, Fig. 9	Consecutive vectors share exactly d−1 coordinate values
Upper 7-bit keystream entropy limit	Eq. (7), Table 4, Fig. 5	The most significant plaintext bit remains unchanged under the baseline XOR mask
Limited plaintext differential propagation	Eq. (8), Table 6	A single plaintext-bit modification changes only one ciphertext bit
Known-plaintext keystream recovery	Eqs. (8)–(9), Section 5.4	Reusing the deterministic XOR keystream compromises confidentiality
Authenticated-encryption integrity verification	Table 7	AES-256-GCM rejected all three tested tampering categories
Host-specific computational performance	Table 8, Fig. 10	The optimized AES-GCM implementation was faster under the tested conditions, without establishing a universal algorithmic ranking

MATHEMATICAL AND DIMENSIONALITY-RELATED FINDINGS

The nonlinear dynamical analysis demonstrated that the logistic map exhibits parameter-dependent transitions between regular, periodic, and chaotic regimes. The bifurcation structure and finite-time Lyapunov exponent estimates confirmed the presence of regions characterized by positive average local expansion. These results establish the nonlinear dynamical foundation of the proposed numerical generator.

Nevertheless, the finite-time Lyapunov exponent characterizes the local sensitivity of the underlying scalar recurrence rather than the independence or cryptographic unpredictability of the generated sequence.

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

The numerical-precision investigation further demonstrated the sensitivity of the logistic trajectory to arithmetic representation. The absolute difference between single-precision and double-precision implementations increased from approximately 1.19×10^{-8} at the first iteration to 0.594697 at iteration 40, before decreasing to 0.292382 at iteration 100.

These findings indicate that numerical divergence is substantial but nonmonotonic, reflecting the bounded nonlinear characteristics of the system. The results also emphasize that reproducibility requires explicit specification of arithmetic precision, evaluation order, and numerical transformations.

The dimensionality-scaling analysis revealed a systematic increase in the participation-ratio effective rank as the nominal dimension increased from 2D to 32D. Specifically, the effective rank increased from 1.913 to 27.862, indicating that additional coordinates contributed to a broader distribution of standardized variance across principal directions.

However, the effective-to-nominal dimensionality ratio decreased from 0.957 to 0.871 over the same interval. This reduction demonstrates that the relative effective dimensionality becomes smaller as additional coordinates are incorporated into the generated representation.

Importantly, the exact overlapping-window identity establishes that neighboring vectors share d−1 coordinate values. The maximum overlap residual was numerically zero for every investigated dimensional configuration.

The combined interpretation of Table 5 and Figs. 6–9 therefore demonstrates that increased nominal dimensionality does not eliminate deterministic structural dependence.

Consequently, the proposed high-dimensional construction is more appropriately described as a reproducible nonlinear feature-generation method than as a mechanism for creating statistically independent multidimensional random samples.

STATISTICAL PROPERTIES AND LIMITATIONS OF THE ORIGINAL KEYSTREAM

The statistical analysis of the original radius-conditioned keystreams demonstrated that the one-, two-, and three-radius configurations exhibited relatively high marginal entropy within their restricted output alphabet.

The measured entropy values were 6.99398, 6.99312, and 6.99416 bits per byte, respectively. These results indicate that the observed symbol frequencies were relatively dispersed across the 128 accessible keystream values.

Nevertheless, the empirical entropy values remained subject to the theoretical 7-bit upper bound imposed by the modulo-128 operation. Consequently, none of the configurations could generate an unrestricted 8-bit keystream distribution.

The lag-one Pearson correlation coefficients were relatively small, ranging from 0.00318 to 0.00705. However, the changes in correlation were not monotonic with increasing radius count, and the results did not demonstrate a systematic improvement across all statistical indicators.

Similarly, chi-square testing produced p-values of 0.238, 0.0482, and 0.343 for the three configurations. The two-radius result fell slightly below the nominal 0.05 significance threshold, indicating that the empirical symbol-frequency behavior was not uniformly favorable across the configurations under the adopted reference test.

These findings highlight the importance of evaluating multiple statistical characteristics rather than relying on a single entropy or correlation metric.

More fundamentally, the results demonstrate that relatively high empirical entropy does not eliminate deterministic vulnerabilities in the encryption mechanism. Since every keystream value is restricted to the interval from 0 to 127, the most significant mask bit remains zero, allowing the corresponding plaintext bit to pass through the XOR operation unchanged.

Therefore, the statistical analysis does not support the claim that increasing the number of radii necessarily produces stronger cryptographic protection.

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

CRYPTANALYTIC FINDINGS AND AUTHENTICATED-ENCRYPTION PERFORMANCE

The differential and known-plaintext analyses provided direct evidence of the security limitations associated with the original deterministic XOR construction.

For all three radius configurations, encryption and decryption successfully recovered the original plaintext. However, a single plaintext-bit modification resulted in exactly one changed ciphertext bit and one affected byte position.

This behavior demonstrates the absence of cross-position diffusion under the tested fixed-keystream conditions.

Furthermore, the XOR identity establishes that the keystream can be recovered directly from a known plaintext–ciphertext pair. When the same keystream is reused, the recovered values can be applied to corresponding positions of other ciphertexts.

These vulnerabilities remain present regardless of the number of radius-conditioned sequences because they arise from the algebraic structure of the encryption operation.

The original framework also lacks an authentication mechanism capable of detecting unauthorized ciphertext modifications. Therefore, the baseline encryption procedure cannot provide the integrity protection required for secure handling of digital information.

The authenticated-encryption redesign addressed these limitations by separating the radius-transformed logistic generator from the cryptographic protection mechanism.

The AES-256-GCM implementation successfully recovered the original plaintext, produced different ciphertexts using independently generated nonces, and rejected modifications to ciphertext, authentication-tag information, and associated authenticated data.

All three tested tampering categories were rejected, as summarized in Table 7.

These results verify selected correctness and integrity behaviors of the implemented AES-GCM interface. However, they should not be interpreted as an independent security proof or as evidence of cryptographic properties contributed by the logistic transformation.

The confidentiality and integrity properties of the redesigned architecture depend on the standardized authenticated-encryption primitive, together with correct key management, nonce handling, and implementation procedures.

COMPUTATIONAL PERFORMANCE AND PRACTICAL IMPLICATIONS

The exploratory computational-performance analysis demonstrated increasing execution times for the original Python-based logistic keystream implementation as payload size increased.

The median execution time increased from 0.2269 ms for a 256-byte payload to 12.9369 ms for 16,384 bytes.

In comparison, the optimized AES-256-GCM library required approximately 0.0021 ms and 0.0263 ms for the corresponding payload sizes.

These measurements indicate that the evaluated AES-GCM implementation was substantially faster under the investigated host-specific conditions.

However, the timing results do not establish the inherent computational superiority of AES-GCM over every possible implementation of a radius-conditioned chaotic generator. The observed differences reflect distinct programming environments, software optimization levels, computational operations, and security functionality.

The legacy implementation performs iterative numerical calculations through Python-level operations, whereas AES-GCM benefits from optimized cryptographic routines. Consequently, a controlled comparison using matched implementation conditions would be required to establish broader performance conclusions.

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

Nevertheless, the measured results support the practical feasibility of using a standard authenticated-encryption library alongside the independent numerical-generation framework.

OVERALL SCIENTIFIC SIGNIFICANCE

The combined results provide several important insights into the relationship between nonlinear dynamics, dimensionality, statistical variability, and cryptographic protection.

First, the radius-transformed logistic construction provides a mathematically defined and reproducible approach to generating bounded multidimensional numerical representations. Its nonlinear characteristics and configurable dimensionality may be useful for studying structured numerical features and deterministic dynamical behavior.

Second, the effective-rank and correlation analyses demonstrate that nominal dimensional expansion should not be equated with independent information generation. Although the effective rank increases with dimension, deterministic coordinate overlap and residual dependence remain present.

Third, the statistical analysis demonstrates that high empirical entropy and weak adjacent linear correlation are insufficient to establish cryptographic security. The restricted keystream alphabet and exact XOR relationships introduce vulnerabilities that remain independent of the observed numerical irregularity.

Fourth, the differential and known-plaintext results establish that the original radius-conditioned XOR construction does not provide adequate protection against the evaluated attacks. Increasing the number of radii does not remove these structural weaknesses.

Finally, the authenticated-encryption redesign demonstrates the practical importance of separating mathematical data-generation procedures from established cryptographic protection mechanisms.

The principal contribution of the present investigation is therefore not the demonstration of a new cryptographically secure chaotic cipher, but the rigorous characterization of a radius-transformed logistic representation, the identification of its dimensional and statistical limitations, and the evidence-based redesign of its encryption architecture.

Overall, the findings establish a clearer mathematical and computational understanding of radius-conditioned logistic sequences while providing practical guidance for their responsible use. The resulting framework supports reproducible nonlinear data generation and authenticated digital-data protection without conflating dynamical complexity with cryptographic security.

These conclusions provide the foundation for the discussion of reproducibility, limitations, and future research directions presented in Section 6.

REPRODUCIBILITY, LIMITATIONS AND RESPONSIBLE INTERPRETATION

The reproducibility and scientific reliability of computational studies depend on the transparent reporting of numerical procedures, implementation conditions, parameter configurations, and evaluation methods. These considerations are particularly important for chaos-based numerical systems because their theoretical sensitivity to initial conditions and practical dependence on finite-precision arithmetic may lead to substantial variations in generated trajectories across different implementations. Accordingly, the present investigation emphasizes reproducible experimentation, explicit recognition of methodological limitations, and careful interpretation of the relationship between nonlinear dynamics, statistical variability, dimensionality, and cryptographic security.

The experimental findings presented in Sections 5.1–5.7 establish that the radius-transformed logistic framework provides a deterministic mechanism for generating bounded multidimensional numerical representations. However, they also demonstrate that increasing the number of generated coordinates does not necessarily produce statistically independent information and that the original XOR encryption mechanism possesses fundamental vulnerabilities that cannot be eliminated merely by increasing the number of radius-conditioned transformations.

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

The following discussion addresses the principal reproducibility requirements, numerical and statistical limitations, cryptographic considerations, and future research directions associated with the proposed framework.

COMPUTATIONAL REPRODUCIBILITY AND EXPERIMENTAL TRANSPARENCY

All computational experiments were conducted using explicitly defined numerical procedures and parameter settings. The original logistic-map configuration employed a control parameter of 3.8, an initial condition of 0.5, and radius thresholds of 0.776, 0.876, and 0.881. These values were retained to enable direct evaluation of the historical baseline algorithm and permit comparison among one-, two-, and three-radius configurations.

The numerical-generation procedures, statistical analyses, multidimensional feature construction, and cryptographic verification experiments were implemented using Python and associated computational libraries. The execution environment and measured numerical results were documented in accompanying machine-readable artifacts, while the source code and generated figures were provided as supporting materials for reproducing the reported experiments.

The reproducibility package includes the computational procedures used to generate the logistic trajectories, apply radius-conditioned transformations, construct overlapping multidimensional vectors, calculate statistical measures, and evaluate the original deterministic XOR encryption mechanism.

The corresponding experimental outputs include the bifurcation analysis, finite-time Lyapunov exponent estimates, floating-point sensitivity measurements, keystream entropy and correlation statistics, effective-rank calculations, overlap verification, cryptanalytic results, and exploratory runtime measurements.

An important distinction must be maintained between deterministic and nondeterministic components of the framework. The original logistic-map generator employs fixed numerical parameters and therefore produces reproducible outputs when identical numerical representations and evaluation procedures are used. In contrast, the authenticated-encryption demonstrator employs independently generated random nonces. Consequently, repeated AES-GCM encryption operations are not expected to produce identical ciphertexts when different nonces are used.

To reproduce the authenticated-encryption tests, the same plaintext, secret key, nonce, and associated authenticated data must be supplied when exact ciphertext reproduction is required. Where independently generated random nonces are employed, reproducibility should instead be evaluated through functional properties such as successful authenticated recovery and rejection of deliberately modified inputs.

Numerical reproducibility also requires consideration of implementation details beyond the mathematical recurrence. In particular, floating-point precision, parameter representation, operation ordering, rounding conventions, and radius-acceptance rules may affect the resulting sequences.

Therefore, the present study reports the relevant numerical settings and distinguishes bitwise reproducibility from statistical consistency. Although deterministic configurations facilitate repeated numerical experiments, equivalent mathematical expressions implemented using different arithmetic environments are not necessarily guaranteed to produce identical floating-point trajectories.

NUMERICAL PRECISION AND DYNAMICAL LIMITATIONS

The theoretical logistic map is defined as a nonlinear dynamical system over real-valued states, whereas its practical implementation relies on finite-precision numerical representations. This distinction introduces limitations that must be considered when interpreting chaotic sensitivity and long-term trajectory behavior.

As demonstrated in Section 5.1, the trajectories generated using single-precision and double-precision arithmetic exhibit substantial divergence after repeated iterations. The measured absolute difference increases from approximately 1.19×10^{-8} at the first iteration to 0.594697 at iteration 40, followed by nonmonotonic variations at later iterations.

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

These results confirm that the computed trajectories are sensitive to numerical representation. However, the observed differences should not be interpreted as direct evidence of cryptographic unpredictability or increased encryption strength.

Finite-precision implementations possess a limited number of representable numerical states. Consequently, deterministic iterations performed within a finite state space must eventually revisit previously encountered states or enter repeated numerical behavior, assuming a fixed deterministic transition rule.

The onset and length of such numerical cycles depend on the arithmetic representation, parameter selection, and implementation details. The present investigation did not perform an exhaustive determination of cycle lengths or recurrence periods across all possible initial conditions and radius configurations.

Therefore, the reported finite-time dynamical results should not be generalized into claims about arbitrarily long nonrepeating trajectories.

A further limitation arises from the radius-conditioned acceptance mechanism. Small numerical differences may influence whether individual logistic states satisfy the prescribed threshold, particularly when the generated values lie close to the acceptance boundary.

Such differences can modify the set of accepted observations and subsequently affect the transformed coordinates and multidimensional sliding-window representations.

Accordingly, future investigations should examine numerical recurrence characteristics, acceptance-boundary sensitivity, and the effects of alternative arithmetic implementations. These analyses would provide a more complete understanding of the computational limitations associated with radius-transformed logistic sequences.

STATISTICAL AND DIMENSIONALITY-RELATED LIMITATIONS

The statistical analyses presented in Sections 5.2 and 5.3 characterize selected properties of the generated sequences and multidimensional representations. Nevertheless, the employed measures describe specific aspects of numerical behavior and should not be interpreted as comprehensive indicators of independence or unpredictability.

The empirical entropy values obtained for the one-, two-, and three-radius keystream configurations are close to the theoretical maximum of 7 bits per symbol imposed by the modulo-128 operation. However, this result characterizes the marginal distribution of accessible symbols rather than the complete dependence structure of the generated sequence.

Similarly, the relatively small lag-one Pearson correlation coefficients indicate weak adjacent linear relationships under the investigated conditions but do not establish statistical independence. Nonlinear dependencies, longer-range correlations, and deterministic relationships may remain present despite low pairwise linear correlation.

The chi-square goodness-of-fit analysis also requires cautious interpretation. The reported p-values describe deviations from a specified uniform-frequency reference distribution for the selected sequences. Because the underlying observations are generated by deterministic nonlinear trajectories, the assumptions conventionally used to interpret chi-square reference distributions may not be fully satisfied.

Consequently, isolated p-values should not be treated as definitive evidence that the generated sequences are random or suitable for cryptographic applications.

The dimensionality analysis introduces a further important limitation. Although the participation-ratio effective rank increases from 1.913 for the 2D representation to 27.862 for the 32D configuration, the normalized effective-rank fraction decreases with dimensional expansion.

More importantly, the overlapping-window construction introduces an exact deterministic relationship between consecutive vectors. Neighboring observations share all but one coordinate, and the measured maximum overlap residual is zero for every investigated dimensional configuration.

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

Therefore, the generated vectors cannot be regarded as independently sampled multidimensional observations.

The effective-rank measure characterizes the distribution of linear variance across principal directions and does not establish nonlinear independence, intrinsic geometric dimension, or independent cryptographic entropy.

The present investigation was also limited to five nominal dimensional configurations and a specified radius-conditioned sequence. Although this design enabled controlled comparison across dimensions, broader conclusions would require additional parameter configurations, independent trajectories, alternative window lengths, and validation datasets.

Future research should therefore distinguish among nominal dimensionality, linear effective rank, nonlinear dependence, and the number of statistically independent observations.

CRYPTOGRAPHIC SECURITY LIMITATIONS AND RESPONSIBLE INTERPRETATION

The original radius-conditioned logistic XOR algorithm was evaluated as a deterministic baseline for understanding the relationship between chaotic numerical generation and encryption behavior.

The cryptanalytic results establish that the baseline construction is vulnerable to keystream reuse, known-plaintext exposure, and predictable ciphertext modification.

The experimental differential analysis further demonstrates that modifying a single plaintext bit changes exactly one ciphertext bit when the keystream remains fixed. This behavior confirms the absence of cross-position diffusion under the investigated conditions.

These limitations arise from the algebraic structure of XOR encryption and cannot be eliminated merely by increasing the number of radius-conditioned logistic transformations.

Accordingly, the present study does not claim that the original chaotic generator constitutes a cryptographically secure pseudorandom number generator or that the proposed high-dimensional transformation independently provides secure encryption.

The authenticated-encryption redesign was introduced to separate numerical data generation from cryptographic protection. In this architecture, AES-256-GCM provides the encryption and authentication functionality, while the radius-transformed logistic generator remains an independent numerical-processing component.

The reported AES-GCM experiments verified successful plaintext recovery, ciphertext variation under different tested nonces, and rejection of three selected categories of authenticated-input modification.

However, these verification experiments do not constitute an independent mathematical security proof or comprehensive cryptographic certification. The broader security guarantees associated with AES-GCM depend on correct implementation and compliance with established requirements for key management, nonce uniqueness, authentication-tag verification, and operational limits [1,4,8].

Furthermore, authenticated encryption does not automatically prevent replay attacks involving previously valid authenticated messages. Appropriate message-freshness mechanisms must be provided by the surrounding application or communication protocol when replay resistance is required.

The present work also does not claim successful completion of the full NIST SP 800-22 statistical test suite. Although selected entropy, correlation, and frequency-distribution analyses were performed, the complete standardized battery was not executed.

The NIST statistical framework provides methods for examining selected properties of binary sequences, but passing such tests would not independently establish cryptographic security [5].

Therefore, any future proposal to use radius-transformed logistic sequences directly as cryptographic key material or encryption keystreams would require substantially stronger analysis, including explicit security definitions, independent cryptanalysis, key-space evaluation, and implementation-level testing.

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

These considerations reinforce the importance of maintaining a clear distinction between nonlinear dynamical complexity, statistical variability, and demonstrable resistance to cryptographic attacks.

FUTURE RESEARCH DIRECTIONS AND SCIENTIFIC DEVELOPMENT

The findings of the present investigation identify several directions for extending the mathematical characterization and practical evaluation of radius-transformed logistic sequences.

First, future studies should examine a broader range of logistic control parameters, initial conditions, and radius thresholds. Such investigations would clarify how parameter selection influences acceptance rates, coordinate distributions, temporal dependence, effective rank, and numerical reproducibility.

Second, the dimensionality analysis should be extended beyond the configurations investigated in this study. Alternative window lengths, nonoverlapping representations, and independently initialized trajectories could be examined to determine how construction choices influence structural redundancy and effective dimensionality.

Comparative evaluation against established numerical feature-generation methods would also be valuable. Such comparisons should consider statistical distributions, dependence measures, computational efficiency, and application-specific usefulness rather than relying exclusively on entropy or visual complexity.

Third, a more comprehensive investigation of finite-precision effects should be undertaken. In particular, numerical cycle lengths, repeated-state behavior, acceptance-boundary sensitivity, and cross-platform reproducibility should be examined using controlled arithmetic implementations.

Fourth, additional statistical analyses should be performed using larger samples and more extensive evaluation procedures. Where appropriate, future investigations may apply the full NIST SP 800-22 statistical test suite with suitable sample lengths, independent sequence configurations, and complete reporting of the relevant statistical outcomes [5].

However, statistical testing should remain complementary to, rather than a substitute for, cryptographic analysis.

Fifth, future work may investigate the usefulness of radius-transformed logistic representations in numerical simulation, nonlinear feature extraction, temporal-data modeling, and controlled synthetic-data generation. These applications may provide a more direct basis for evaluating the scientific contribution of the proposed geometric transformation.

Finally, any future chaos-based encryption primitive developed from the present numerical framework should undergo independent security evaluation before practical deployment. Such evaluation should include known-plaintext and chosen-plaintext attacks, differential analysis, keystream-reuse assessment, key-management review, finite-state analysis, and implementation-level testing [1,4,8].

The development of a new cryptographic primitive should not be inferred solely from the combination of nonlinear transformations, multiple radii, or high-dimensional numerical representations.

Overall, the limitations identified in this study do not invalidate the mathematical usefulness of the radius-transformed logistic construction. Instead, they define the appropriate scope of its demonstrated contribution and establish a transparent foundation for further research.

The proposed framework should therefore be interpreted primarily as a reproducible approach to nonlinear numerical data generation, accompanied by an independent authenticated-encryption mechanism for protecting digital information.

This interpretation preserves the scientific value of the mathematical analysis while avoiding unsupported claims regarding independent randomness, cryptographic unpredictability, or security improvements arising solely from dimensional expansion.

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

CONCLUSION

This study presented a comprehensive mathematical, statistical, and cryptanalytic investigation of radius-transformed logistic maps for high-dimensional data generation and digital information protection. By integrating nonlinear dynamical analysis, dimensionality characterization, numerical-precision evaluation, and encryption-security assessment, the proposed framework establishes an evidence-based distinction between chaotic numerical complexity and demonstrable cryptographic security. The findings confirm that radius-conditioned logistic transformations provide a deterministic and reproducible mechanism for constructing bounded multidimensional numerical representations, while revealing fundamental limitations associated with statistical dependence and the original XOR-based encryption approach.

The dimensionality analysis demonstrated that the participation-ratio effective rank increased from 1.913 at 2D to 27.862 at 32D, whereas the effective-to-nominal dimensionality ratio decreased from 0.957 to 0.871. Moreover, consecutive multidimensional vectors shared exactly $d - 1$ coordinates, confirming persistent deterministic overlap despite increasing nominal dimensionality. The radar plot, heatmap matrix, and Pearson correlation analysis further demonstrated that dimensional expansion does not automatically generate statistically independent information. Numerical-precision experiments additionally revealed substantial, nonmonotonic divergence between float32 and float64 logistic trajectories, emphasizing the importance of explicitly defined arithmetic procedures for reproducible implementation.

The statistical evaluation of the original one-, two-, and three-radius keystream configurations yielded empirical entropy values of 6.99398, 6.99312, and 6.99416 bits per byte, respectively. Nevertheless, the modulo-128 operation restricted the output to 128 possible symbols, imposing a maximum entropy of 7 bits per symbol and leaving the most significant plaintext bit unchanged during XOR encryption. Differential analysis confirmed that modifying one plaintext bit changed exactly one ciphertext bit, demonstrating the absence of cross-position diffusion. Furthermore, deterministic keystream reuse enabled known-plaintext recovery and predictable ciphertext modification, establishing vulnerabilities that cannot be eliminated merely by increasing the number of radii.

To address these limitations, the revised architecture separated nonlinear numerical generation from cryptographic protection through an independently implemented AES-256-GCM authenticated-encryption layer. The implementation successfully recovered the original plaintext, produced different ciphertexts with distinct tested nonces, and rejected all three evaluated categories of authenticated-input tampering. These results verify the tested implementation behavior, while the underlying confidentiality and integrity guarantees remain dependent on correct use of the standardized cryptographic primitive.

The principal scientific contribution of this work is therefore the rigorous characterization of radius-transformed logistic representations, the mathematical identification of their dimensional dependencies, and the experimentally supported reassessment of their encryption limitations. Future investigations should extend parameter-sensitivity studies, evaluate independent and nonoverlapping sampling strategies, conduct comprehensive statistical testing, and compare the proposed generator against established numerical alternatives. Overall, the findings provide a reproducible foundation for nonlinear high-dimensional feature generation while demonstrating that reliable digital-data protection requires independently validated cryptographic mechanisms rather than chaotic complexity alone.

REFERENCES

[1] G. Alvarez and S. Li, “Some basic cryptographic requirements for chaos-based cryptosystems,” *International Journal of Bifurcation and Chaos*, vol. 16, no. 8, pp. 2129–2151, 2006, doi: https://doi.org/10.1142/S0218127406015970.
[2] L. M. Pecora and T. L. Carroll, “Synchronization in chaotic systems,” *Physical Review Letters*, vol. 64, no. 8, pp. 821–824, 1990, doi: https://doi.org/10.1103/PhysRevLett.64.821.

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

[3] J. Fridrich, "Symmetric ciphers based on two-dimensional chaotic maps," *International Journal of Bifurcation and Chaos*, vol. 8, no. 6, pp. 1259–1284, 1998, doi: https://doi.org/10.1142/S021812749800098X.
[4] G. Alvarez and S. Li, "Cryptanalyzing a nonlinear chaotic algorithm (NCA) for image encryption," *Communications in Nonlinear Science and Numerical Simulation*, vol. 14, no. 11, pp. 3743–3749, 2009, doi: https://doi.org/10.1016/j.cnsns.2009.02.033.
[5] L. E. Bassham III, A. L. Rukhin, J. Soto, J. R. Nechvatal, M. E. Smid, E. B. Barker, S. D. Leigh, M. Levenson, M. Vangel, D. L. Banks, N. A. Heckert, J. Dray, and S. Vo, "A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications," *NIST Special Publication*, SP 800-22 Rev. 1a, 2010, doi: https://doi.org/10.6028/NIST.SP.800-22r1a.
[6] M. Dworkin, "Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC," *NIST Special Publication*, SP 800-38D, 2007, doi: https://doi.org/10.6028/NIST.SP.800-38D.
[7] National Institute of Standards and Technology, "Advanced Encryption Standard (AES)," *Federal Information Processing Standards Publication*, FIPS 197, updated edition, 2023, doi: https://doi.org/10.6028/NIST.FIPS.197-upd1.
[8] E. Barker and J. Kelsey, "Recommendation for Random Number Generation Using Deterministic Random Bit Generators," *NIST Special Publication*, SP 800-90A Rev. 1, 2015, doi: https://doi.org/10.6028/NIST.SP.800-90Ar1.
[9] H. Krawczyk and P. Eronen, "HMAC-based Extract-and-Expand Key Derivation Function (HKDF)," *Internet Engineering Task Force*, RFC 5869, 2010, doi: https://doi.org/10.17487/RFC5869.
[10] Y. Nir and A. Langley, "ChaCha20 and Poly1305 for IETF Protocols," *Internet Engineering Task Force*, RFC 8439, 2018, doi: https://doi.org/10.17487/RFC8439.
[11] S.-S. Yu, N.-R. Zhou, L.-H. Gong, and Z. Nie, "Optical image encryption algorithm based on phase-truncated short-time fractional Fourier transform and hyper-chaotic system," *Optics and Lasers in Engineering*, vol. 124, p. 105816, 2020, doi: https://doi.org/10.1016/j.optlaseng.2019.105816.
[12] X. Wang, H. Zhao, L. Feng, X. Ye, and H. Zhang, "High-sensitivity image encryption algorithm with random diffusion based on dynamic-coupled map lattices," *Optics and Lasers in Engineering*, vol. 122, pp. 225–238, 2019, doi: https://doi.org/10.1016/j.optlaseng.2019.04.005.
[13] X. Wang, J. Zhang, and G. Cao, "An image encryption algorithm based on ZigZag transform and LL compound chaotic system," *Optics & Laser Technology*, vol. 119, p. 105581, 2019, doi: https://doi.org/10.1016/j.optlastec.2019.105581.
[14] C. Zhu and K. Sun, "Cryptanalyzing and improving a novel color image encryption algorithm using RT-enhanced chaotic tent maps," *IEEE Access*, vol. 6, pp. 18759–18770, 2018, doi: https://doi.org/10.1109/ACCESS.2018.2817600.
[15] X. Zhang and X. Wang, "Digital image encryption algorithm based on elliptic curve public cryptosystem," *IEEE Access*, vol. 6, pp. 70025–70034, 2018, doi: https://doi.org/10.1109/ACCESS.2018.2879844.
[16] M. Guan, X. Yang, and W. Hu, "Chaotic image encryption algorithm using frequency-domain DNA encoding," *IET Image Processing*, vol. 13, no. 9, pp. 1535–1539, 2019, doi: https://doi.org/10.1049/iet-ipr.2019.0051.
[17] G. R. W. Thoms, R. Muresan, and A. Al-Dweik, "Chaotic encryption algorithm with key controlled neural networks for intelligent transportation systems," *IEEE Access*, vol. 7, pp. 158697–158709, 2019, doi: https://doi.org/10.1109/ACCESS.2019.2950007.
[18] H. Nematzadeh, R. Enayatifar, H. Motameni, F. G. Guimarães, and V. N. Coelho, "Medical image encryption using a hybrid model of modified genetic algorithm and coupled map lattices," *Optics and Lasers in Engineering*, vol. 110, pp. 24–32, 2018, doi: https://doi.org/10.1016/j.optlaseng.2018.05.009.
[19] Q. Xu, K. Sun, and C. Zhu, "A visually secure asymmetric image encryption scheme based on RSA algorithm and hyperchaotic map," *Physica Scripta*, vol. 95, no. 3, p. 035223, 2020, doi: https://doi.org/10.1088/1402-4896/ab52bc.
[20] J. Xu, P. Li, F. Yang, and H. Yan, "High intensity image encryption scheme based on quantum logistic chaotic map and complex hyperchaotic system," *IEEE Access*, vol. 7, pp. 167904–167918, 2019, doi: https://doi.org/10.1109/ACCESS.2019.2952140.

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

[21] S. M. R. Farschi and H. Farschi, "A novel chaotic approach for information hiding in image," *Nonlinear Dynamics*, vol. 69, no. 4, pp. 1525–1539, 2012, doi: https://doi.org/10.1007/s11071-012-0367-5.
[22] C. Liu and Q. Ding, "A modified algorithm for the logistic sequence based on PCA," *IEEE Access*, vol. 8, pp. 45254–45262, 2020, doi: https://doi.org/10.1109/ACCESS.2020.2978503.
[23] S. Li, G. Chen, and X. Mou, "On the dynamical degradation of digital piecewise linear chaotic maps," *International Journal of Bifurcation and Chaos*, vol. 15, no. 10, pp. 3119–3151, 2005, doi: https://doi.org/10.1142/S0218127405014052.
[24] L. Kocarev, "Chaos-based cryptography: A brief overview," *IEEE Circuits and Systems Magazine*, vol. 1, no. 3, pp. 6–21, 2001, doi: https://doi.org/10.1109/7384.963463.
[25] R. M. May, "Simple mathematical models with very complicated dynamics," *Nature*, vol. 261, no. 5560, pp. 459–467, 1976, doi: https://doi.org/10.1038/261459a0.
[26] M. J. Feigenbaum, "Quantitative universality for a class of nonlinear transformations," *Journal of Statistical Physics*, vol. 19, no. 1, pp. 25–52, 1978, doi: https://doi.org/10.1007/BF01020332.
[27] R. M. May and G. F. Oster, "Bifurcations and dynamic complexity in simple ecological models," *The American Naturalist*, vol. 110, no. 974, pp. 573–599, 1976, doi: https://doi.org/10.1086/283092.
[28] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948, doi: https://doi.org/10.1002/j.1538-7305.1948.tb01338.x.
[29] D. McGrew, "An interface and algorithms for authenticated encryption," *Internet Engineering Task Force*, RFC 5116, 2008, doi: https://doi.org/10.17487/RFC5116.
[30] C. E. Shannon, "Communication theory of secrecy systems," Bell System Technical Journal, vol. 28, no. 4, pp. 656–715, 1949, doi: https://doi.org/10.1002/j.1538-7305.1949.tb00928.x.

Appendix A. Implementability and repeatable evaluation

The accompanying package contains: measured_results.json; source code build_chaos_full.py; and ten high-resolution scientific figures. Run the script in a Python environment containing numpy, scipy, matplotlib, python-docx and cryptography. Values are reproducible for the deterministic numerical sections; AES-GCM nonce bytes and wall-clock timings are expected to differ across runs. No external network access is required for the computations. A full NIST SP 800-22 suite or security formalization is outside this reported experiment and is not represented as completed.

For production code: derive or create keys using approved key-generation or key-derivation methods, use a unique 96-bit GCM nonce per key, include version metadata in AAD, reject tampered ciphertext by authentication failure, and never fall back to the legacy XOR method [6–9,29].

Appendix:

Code python for N-Dimensional Logistic Map:

```
import numpy as np
r_logistic = 3.8
k1 = 0.776
target_length = 77
x = 0.5
logistic_values=[]
while len(logistic_values) < target_length + 3:
   x = r_logistic * x * (1 – x)
   if x < k1
      logistic_values.append(k)
x_list=[]
y_list=[]
z_list=[]
w_list=[]
for i in range(target_length):
   x1 = logistic_values[i]
   x2 = logistic_values[i + 1]
```

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

```
    x3 = logistic_values[i + 2]
    x4 = logistic_values[i + 3]
    x = np.sqrt(k1**2 - x1**2)
    y = np.sqrt(k1**2 - x2**2)
    z = np.sqrt(k1**2 - x3**2)
    w = np.sqrt(k1**2 - x4**2)
    x_list.append(x)
    y_list.append(y)
    z_list.append(z)
    w_list.append(w)
with open("output3.txt", "w", encoding="utf-8") as f:
    for i in range(target_length):
        f.write(f"{i+1}\t{x_list[i]:.12f}\t{y_list[i]:.12f}\t{z_list[i]:.12f}\t{w_list[i]:.12f}\n”)
```

Encryption code python two dimension :

```
import tkinter as tk
from tkinter import filedialog
import numpy as np
root = tk.Tk()
root.withdraw()
file_path = filedialog.askopenfilename(filetypes=[("Text Files", "*.txt")])
with open(file_path, 'r', encoding='utf-8') as file:
    text = file.read()
if text.endswith(" "):
    text = text.rstrip()
ascii_values = [ord(c) for c in text]
print("Saving plain ASCII numeric values...")
input_ascii_path = filedialog.asksaveasfilename(defaultextension=".txt", filetypes=[("Text Files", "*.txt")])
with open(input_ascii_path, 'w', encoding='utf-8') as f:
    f.write(' '.join(str(val) for val in ascii_values))
print("Plain ASCII numeric values saved.")
iterations = len(ascii_values)
print(f"Number of characters: {iterations}")
x = 0.5
r = 3.8
k1 = 0.776
k2 = 0.876
y1 = []
x1 = x
for _ in range(1000):
    x1 = r * x1 * (1 - x1)
    val = np.sqrt(abs(k1**2 - x1**2))
    if x1 < k1:
        y1.append(val)
if len(y1) < iterations:
    raise ValueError("Not enough logistic values for k1")
logistic_k1 = y1[:iterations]
y2 = []
x2 = x
for _ in range(1000):
    x2 = r * x2 * (1 - x2)
    val = np.sqrt(abs(k2**2 - x2**2))
    if x2 < k2:
        y2.append(val)
```

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

```
if len(y2) < iterations:
    raise ValueError("Not enough logistic values for k2")
logistic_k2 = y2[:iterations]
logistic_combined = [a * b for a, b in zip(logistic_k1, logistic_k2)]
scaled_values = [round(val * 100000) % 128 for val in logistic_combined]
encrypted_values = [a ^ b for a, b in zip(ascii_values, scaled_values)]
encrypted_text = ''.join(chr(val) for val in encrypted_values)
print("Saving encrypted numeric values...")
save_path = filedialog.asksaveasfilename(defaultextension=".txt", filetypes=[("Text Files", "*.txt")])
with open(save_path, 'w', encoding='utf-8') as f:
    f.write(' '.join(str(val) for val in encrypted_values))
print("Done.")
```

Decryption code python two dimension :

```
import tkinter as tk
from tkinter import filedialog
import numpy as np
import time
start_time = time.time()
root = tk.Tk()
root.withdraw()
file_path = filedialog.askopenfilename(filetypes=[("Text Files", "*.txt")])
with open(file_path, 'r', encoding='utf-8') as file:
    text = file.read()
encrypted_values = list(map(int, text.strip().split()))
iterations = len(encrypted_values)
print(f"Number of encrypted values: {iterations}")
x = 0.5
r = 3.8
k1 = 0.776
k2 = 0.876
y1 = []
x1 = x
for _ in range(1000):
    x1 = r * x1 * (1 - x1)
    val = np.sqrt(abs(k1**2 - x1**2))
    if x1 < k1:
        y1.append(val)
if len(y1) < iterations:
    raise ValueError("Not enough logistic values generated for k1")
logistic_k1 = y1[:iterations]
y2 = []
x2 = x
for _ in range(1000):
    x2 = r * x2 * (1 - x2)
    val = np.sqrt(abs(k2**2 - x2**2))
    if x2 < k2:
        y2.append(val)
if len(y2) < iterations:
    raise ValueError("Not enough logistic values generated for k2")
logistic_k2 = y2[:iterations]
logistic_combined = [a * b for a, b in zip(logistic_k1, logistic_k2)]
scaled_values = [round(val * 100000) % 128 for val in logistic_combined]
decrypted_values = [a ^ b for a, b in zip(encrypted_values, scaled_values)]
```

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com

```
decrypted_text = ''.join(chr(val) for val in decrypted_values)
print("Saving decrypted text...")
output_path = filedialog.asksaveasfilename(defaultextension=".txt", filetypes=[("Text Files", "*.txt")])
with open(output_path, 'w', encoding='utf-8') as f:
    f.write(decrypted_text)
print("Decryption complete.")
print(f"Execution time: {time.time() - start_time:.2f} seconds")
```

*Corresponding author
Mohammed RASHEED,
College of Production Engineering & Metallurgy, University of Technology- Iraq, Baghdad 10066, Iraq
e-mail: rasheed.mohammed40@yahoo.com