Journal of Online Engineering Education
ISSN: 2158-9658 Volume: 17 Issue: 1
Article Received: 16 January 2026 **Revised**: 15 March 2026 **Accepted**: 14 April 2026

Simulation-Based Performance Evaluation of a GA-Optimized Logistic Map for Lightweight Cryptographic Key Generation in IoT Systems

Shajeela Beegom B A[1], Dr. Karthigai S[2]

PhD Scholar[1],

Assistant Professor[2],

[1,2]Department of Computer Science, Vet Institute of Arts and Science (Co-Education) College, Thindal, Erode

Abstract: The rapid advancement of the Internet of Things (IoT) has highlighted the need for cryptographic methods that can function under stringent computational limitations. Chaotic systems, such as the logistic map, have drawn considerable attention from researchers because of their complex features, extreme sensitivity to initial conditions, and unpredictable behavior. The choice of suitable control parameters significantly affects the encryption effectiveness of these chaotic systems. To create lightweight cryptographic keys, especially for Internet of Things (IoT) applications, this study presents an advanced version of the logistic map using a genetic algorithm. This technique maximizes the entropy and minimizes the autocorrelation by optimizing the map parameters. The findings show that it is fascinating to see how binary cryptographic keys are created from chaotic sequences, proving their effectiveness in guaranteeing safe communication between IoT devices. Experts assessed the methodology using NIST statistical tests for randomness, entropy autocorrelation analysis, and execution time. Notably, the GA-optimized Logistic map demonstrated increased efficiency and randomness compared to the Tent and Chebyshev maps. According to the experimental results, this framework offers a safe and effective cryptographic solution specifically designed for devices with limited resources.

Keywords: Internet of Things (IoT); Lightweight Cryptography; Logistic Map; Genetic Algorithm; Chaotic Maps; Cryptographic Key Generation; Entropy Optimization; NIST Randomness Tests.

1. INTRODUCTION

The Internet of Things (IoT) has quickly spread to smart city infrastructures, transportation, agriculture, industrial automation, and healthcare [6] [7]. Modern IoT systems comprise interconnected sensors, embedded processors, and wireless communication devices that constantly exchange sensitive data with each other. Despite the benefits of IoT technology, most IoT devices have serious drawbacks, such as low processing power, memory, and power consumption [6]. These limitations render traditional cryptographic algorithms, such as RSA and AES, computationally expensive for lightweight environments.

Lightweight cryptography is an essential area of research for secure IoT communication [6]. The aim of lightweight cryptography systems is to minimize computational complexity, memory usage, and execution overhead while preserving adequate security. In recent years, chaotic systems have emerged as appealing choices for lightweight cryptography applications owing to their nonlinear dynamics, pseudorandom characteristics, and high sensitivity to initial conditions [2] [7].

Owing to its straightforward mathematical structure and effective sequence generation capability, the logistic map is widely used in different chaotic systems [3] [9]. However, the selection of suitable initial conditions and control parameters has a significant impact on the random quality of the logistic maps [8]. Cryptographic

Journal of Online Engineering Education
ISSN: 2158-9658 Volume: 17 Issue: 1
Article Received: 16 January 2026 **Revised**: 15 March 2026 **Accepted**: 14 April 2026

unpredictability may be weakened, and entropy may be decreased by poor parameter selection. This study suggests a genetic algorithm-optimized logistic map for lightweight cryptographic key generation to overcome this restriction [6] [8].

The proposed framework generates secure 256-bit cryptographic keys that are appropriate for lightweight IoT environments while increasing entropy, reducing autocorrelation, and improving statistical randomness. Under the same simulation conditions, a comparative analysis was conducted with respect to the Tent and Chebyshev chaotic maps.

2. RELATED WORK

Owing to their nonlinear behavior and pseudo-random properties, chaotic systems have been thoroughly studied for lightweight cryptography applications [7]. Logistic tents, Chebyshev, Lorenz, and Henon maps are frequently employed for the creation of cryptographic sequences and secure communication systems [2] [9]. Nazish and others. demonstrated improved statistical randomness and suggested improved logistic maps for lightweight pseudo-random generation [3]. The Alawida group. examined improved logistic chaotic systems for applications involving the generation of secure random numbers [9]. Rahman and company. enhanced IoT communication security by combining AES encryption with logistic-map-based key generation [10].

Additionally, optimization methods for enhancing chaotic parameter selection have been investigated in recent studies. Genetic algorithms are evolutionary optimization techniques that use iterative fitness evaluations to determine the ideal chaotic parameters [8]. In chaotic cryptographic systems, prior research has shown that GA optimization enhances entropy randomness quality and autocorrelation behavior [6] [8]. Although previous research has improved chaotic randomness, many studies primarily concentrate on statistical validation while offering minimal computational evaluation and limited comparative analysis [7].

To overcome these constraints, this study presents a simulation-based comparative performance assessment of a GA-optimized logistic map for lightweight IoT cryptographic applications.

3. PROPOSED METHODOLOGY

The proposed framework integrates genetic algorithm optimization, chaotic sequence generation and statistical randomness assessment.

3.1 Model with Logistic Map

The mathematical representation of a logistic map is as follows

$$xn+1 = rxn(1 - xn), xn\in(0, 1), \quad r\in(3.\ 7,\ 4)\cdot$$

parameter is between 3. 7 and 4 chaotic behavior takes place.[2], [3]. The logisticx^n maps advantages where the control parameter is denoted by r and the chaotic state variable by . When the control

include low computational complexity, efficient sequence generation, high sensitivity to initial conditions and lightweight implementation suitability [7] [9].

3.2 Optimization using Genetic Algorithm

The logistic map has strong chaotic behavior, but the quality of its randomness is highly dependent on the choice of parameters [8]. The proposed framework uses a genetic algorithm to enhance chaotic parameter tuning.

The Genetic Algorithm optimizes with Initial condition x_0 and with the logistic control parameter r.

The optimization objectives include:

•Maximizing entropy

Journal of Online Engineering Education
ISSN: 2158-9658 Volume: 17 Issue: 1
Article Received: 16 January 2026 **Revised**: 15 March 2026 **Accepted**: 14 April 2026

- Minimizing autocorrelation
- Improving statistical randomness
- Enhancing cryptographic unpredictability

The optimization process includes:

1. Population initialization
2. Fitness evaluation
3. Selection
4. Crossover
5. Mutation
6. Iterative convergence

Each chromosome is represented as: Chromosome = (x_0 , r). The entropy-based fitness evaluation improves randomness quality and cryptographic strength [6], [8].

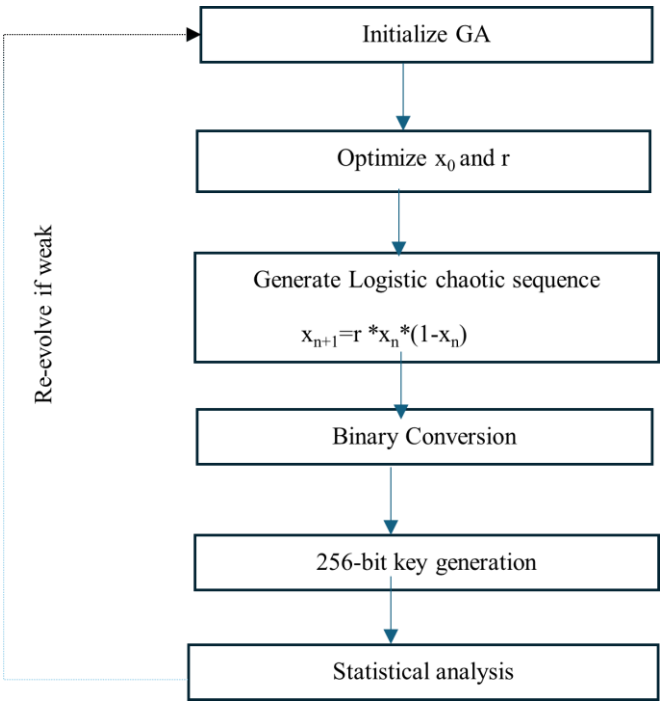
3.3 Procedure for Key Generation

Threshold quantization is used to transform the chaotic sequences produced by the logistic map into binary form after the optimized parameters are obtained. A 256-bit cryptographic key that can be used for random number generation, secure communication, lightweight encryption and authentication is created by grouping the generated binary stream.

3.4 The suggested System Architecture

A genetic algorithm was first used to optimize the proposed architecture to get the optimal chaotic parameters. Chaotic sequences were then generated using optimized parameters.

Figure 1: Proposed System Architecture



Journal of Online Engineering Education
ISSN: 2158-9658 Volume: 17 Issue: 1
Article Received: 16 January 2026 **Revised**: 15 March 2026 **Accepted**: 14 April 2026

To create cryptographic keys, the produced sequences are transformed into binary bits. Finally, to confirm the randomness quality in Figure 1 (System Architecture), entropy analysis, autocorrelation assessment and NIST statistical testing were performed.

4. SIMULATION ENVIRONMENT

In a software-based simulation environment the suggested framework was put into practice using Python. The Genetic Algorithm optimization framework was implemented using DEAP Matplotlib for graphical visualization, SciPy for statistical analysis and NumPy for chaotic sequence generation [8]. The experiments were conducted using the parameters listed in Table1.

Table1: Simulation Parameters Used in the Proposed Framework

Parameters	**Value**
Population size	50
Number of generations	100
Mutation rate	0.1
Crossover rate	0.8
Key length	256 bits
Logistic iterations	5000

Entropy analysis, autocorrelation analysis, NIST statistical testing, execution time analysis and computational complexity evaluation were used to assess the suggested framework.

Entropy analysis is used to gauge the degree of randomness. Higher entropy values indicate greater unpredictability. The statistical dependence between neighboring bits was assessed using autocorrelation analysis. Increased randomness is indicated by lower autocorrelation. The random characteristics were validated using NIST SP 800-22 statistical tests, such as the Frequency Test, Runs Test, Block Frequency Test and Approximate Entropy Test [5].

Algorithm

Algorithm 1: GA-Optimized Chaotic Key
Generation Input:

• Population size P • Number of generations G Output:

• Optimized 256-bit cryptographic key

Step 1: Initialize random population

Step 2: Generate chromosomes represented as (x0, r)

Step 3: Generate chaotic sequences using Logistic map

Step 4: Convert chaotic sequences into binary bits

Step 5: Evaluate fitness using entropy and autocorrelation

Step 6: Select chromosomes with highest fitness

Step 7: Apply crossover operation

Step 8: Apply mutation operation

Journal of Online Engineering Education
ISSN: 2158-9658 Volume: 17 Issue: 1
Article Received: 16 January 2026 **Revised**: 15 March 2026 **Accepted**: 14 April 2026

Step 9: Generate new population

Step 10: Repeat until convergence

Step 11: Obtain optimized parameters

Step 12: Generate optimized chaotic sequence

Step 13: Form 256-bit cryptographic key

Step 14: Perform statistical evaluation

Step 15: Return optimized cryptographic key

Until strong randomness characteristics are attained using the Genetic Algorithm iteratively that enhances chaotic parameters.

5. RESULTS AND DISCUSSION

The ideal chaotic parameters were successfully reached by the suggested Genetic Algorithm with the optimized parameters, and it is in table 2.

Table 2: Optimized Chaotic Parameters Obtained Using Genetic Algorithm

Initial Condition (x0)	0.487214
Control Parameter (r)	3.998721

5.1 Entropy Analysis:

Table 3: Entropy Comparison of Chaotic Key Generation Methods

Method	**Entropy**
GA-Optimized Logistic Map	7.98
Tent Map	7.91
Chebyshev Map	7.89

Using Table 3, an entropy value of 7.98 was obtained by the suggested GA-optimized logistic map which is near the optimal randomness value of 8 [5] [9]. Entropy values for the tent and Chebyshev maps were

7. 91 and 7. 89 respectively.

5.2 Autocorrelation Analysis:

Table 4: Autocorrelation Comparison of Chaotic Maps

Method	**Autocorrelation**
GA-Optimized Logistic Map	0.0021
Tent Map	0.0154
Chebyshev Map	0.0217

Journal of Online Engineering Education
ISSN: 2158-9658 Volume: 17 Issue: 1
Article Received: 16 January 2026 **Revised**: 15 March 2026 **Accepted**: 14 April 2026

Table 4, the autocorrelation value of the suggested framework was 0.0021, while 0.0154 for the tent map and 0. 0217 for the Chebyshev map. Improved unpredictability is indicated by the reduced autocorrelation [8].

5.3 NIST Statistical Testing:

From table 5, you can see the major NIST statistical tests performed using the Frequency Test, Runs Test, Block Frequency Test and Approximate Entropy Test were all passed by the produced sequences. The NIST acceptance condition p 0. 01 was met by all of the obtained p-values [5].

Table 5: NIST SP 800-22 Statistical Test Results

Test	**P-Value**
Frequency Test	0.6812
Runs Test	0.7345
Block Frequency Test	0.5921
Approximate Entropy Test	0.7118

5.4 Execution Time Analysis:

Table 6: Execution Time Comparison of Chaotic Key Generation Methods

Method	**Execution Time (ms)**
GA-Optimized Logistic Map	1.8
Tent Map	2.4
Chebyshev Map	3.1

The proposed framework demonstrated a lower computational overhead than the Tent and Chebyshev maps. The execution time of the GA-optimized Logistic map was measured as 1.8 milli second (ms) compared with 2.4 ms and 3.1 ms for the Tent and Chebyshev maps, respectively. **5.5 Figures**

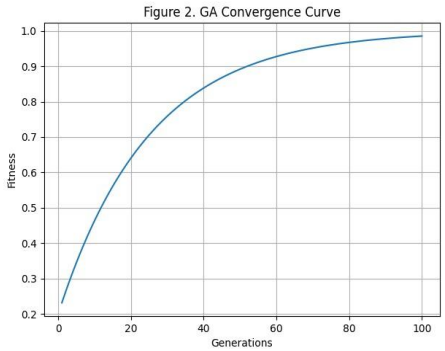


Figure 2: Genetic Algorithm convergence behavior showing gradual fitness improvement during optimization.

Journal of Online Engineering Education
ISSN: 2158-9658 Volume: 17 Issue: 1
Article Received: 16 January 2026 **Revised**: 15 March 2026 **Accepted**: 14 April 2026

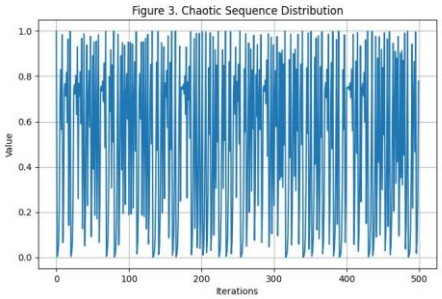


Figure 3: Distribution of generated chaotic values using the optimized logistic map.

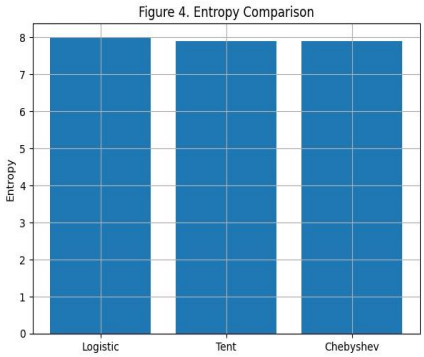


Figure 4: Entropy comparison among Logistic, Tent, and Chebyshev chaotic maps.

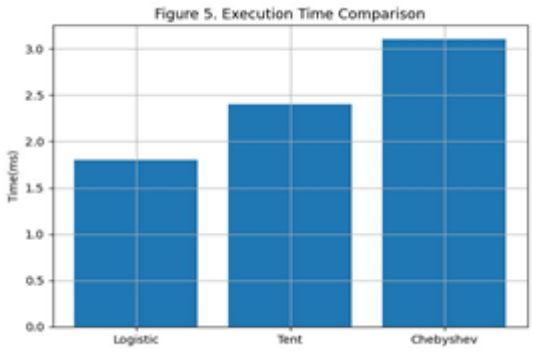


Figure 5: Execution time comparison of different chaotic maps for lightweight cryptographic applications.

6. COMPUTATIONAL COMPLEXITY ANALYSIS

The computational complexity of the proposed framework mainly depends on Genetic Algorithm optimization and chaotic sequence generation. The approximate computational complexity is expressed as:

$O(G \times P \times N)$; where: G = Number of generations, P = Population size and N = Sequence length

The following simulation parameters were used:

$50 \times 100 \times 5000 = 25{,}000{,}000$ operations

The optimization process evaluated approximately 25,000,000 chaotic operations. Although the Genetic Algorithm introduces additional computational processing, the lightweight structure of the logistic map maintains manageable computational requirements suitable for IoT devices [6], [7].

7. CONCLUSION

In this study, a GA-optimized logistic map for lightweight cryptographic key generation in IoT systems was evaluated for performance using simulations. The suggested framework uses a genetic algorithm to optimize chaotic parameters, improving the randomness characteristics [6] [8].

Strong entropy performance, low autocorrelation, successful NIST statistical validation, and effective execution time were all demonstrated in the experimental evaluation. The suggested framework achieves superior randomness quality and lightweight computational efficiency, as confirmed by a comparative analysis against Tent and Chebyshev chaotic maps [9] [10].

The results show that GA-enhanced chaotic systems offer a lightweight cryptographic solution that works well for secure IoT communication. FPGA implementation, embedded hardware acceleration, adaptive chaotic optimization, and real-time IoT deployment will be the main areas of future research.

REFERENCES

[1] Shajeela Beegom, B. A., Karthigai, S., & Selvanayaki, K., "Survey on AI-Enhanced Chaotic Key Generation for Public Key Cryptography."

[2] T. K. Alshekly, E. A. AlBahrani, and L. Ben Ayed, "Code-Based Cryptography and Chaotic Maps as Pseudo-Random Bit Generator," Engineering, Technology & Applied Science Research, vol. 15, no. 6, pp. 29041–29048, 2025.

[3] J. Mir Nazish, M. Javid, and M. T. Banday, "Enhanced logistic map with infinite chaos and its applicability in lightweight and high-speed pseudo-random bit generation," Cybersecurity, vol. 8, p. 24, 2025.

[4] A. A. Mahdi and M. M. Hoobi, "Enhanced Key Generation Method Using Deep Q-Networks Algorithm with Chaotic Maps," Mesopotamian Journal of CyberSecurity, vol. 5, no. 3, pp. 927–952, 2025.

[5] National Institute of Standards and Technology (NIST), A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications (SP 800-22 Rev. 1a), 2010.

[6] P. Nayak and G. Panda, "Lightweight chaotic key generation for secure IoT communication using Genetic Algorithm," IEEE Internet of Things Journal, vol. 9, pp. 14123–14135, 2022.

[7] H. Zhang, J. Li, and G. Chen, "Chaos-based cryptography for secure IoT applications: A review," Security and Communication Networks, 2020.

[8] X. Wu and X. Wang, "Genetic Algorithm based parameter tuning for chaotic cryptosystems," Nonlinear Dynamics, vol. 95, pp. 1113–1126, 2019.

[9] A. Alawida et al., "Enhancing logistic chaotic map for improved cryptographic security in random number generation," Journal Information Security and Applications, vol. 80, 2024.

[10] Z. Rahman et al., "Enhancing AES using chaos and logistic map-based key generation for IoT security," Electronics, vol. 11, no. 7, 2022.

[11] Shajeela Beegom B A and Dr. Karthigai S, "Genetic Algorithm Optimized Chaotic Logistic Map for Lightweight Cryptographic Key Generation in IoT Systems," Journal of Computer Science & Technology, submitted for review, 2026.