Comprehensive Evaluation of Encryption Algorithms: A Study of 22 Performance Tests

1st Areeg Fahad Rasheed
Network Department
Al-Nahrain University
Baghdad, Iraq
areeg.fahad@coie-nahrain.edu.iq

2nd M. Zarkoosh
Software Engineering
Baghdad, Iraq
m94zarkoosh@gmail.com

3rd Safa F. Abbas
Network Department
Al-Nahrain University
Baghdad, Iraq
safaaf.abbas@gmail.com

***Abstract*—Encrypting private information is a critical step in preventing unauthorised access or reading. However, selecting a trustworthy encryption technique is crucial. While many encryption algorithms are produced yearly, only a few established methods exist to assess their performance. Some examples of such methods include GB/T 32915-2016 from SCA, SP 800-22 from NIST, and AIS 20 and AIS 31 from BSI. These methods only do fifteen tests, which may need more to determine how well the encryption scheme works. This study aims to propose a software programmer[1] that can assess the efficacy of any encryption scheme by running its encrypted data through a series of twenty-two tests. To do this, the proposed software is built on top of the Tinker framework based on the Python programming language. The proposed software is tested by evaluating the performance of five different encryption methods: AES, ARC4, RSA, Logistic Map, and SHA-512 with twenty tests. Featuring a user-friendly interface and effortless encryption algorithm evaluation, the proposed software can guide you in making the optimal choice to assess the performance of encryption algorithms.**

***Index Terms*—NITS suite, Security, AES, RC4, Logistic map, SHA-512, RSA**

I. Introduction

With the rise of sophisticated penetration techniques, businesses prioritize using effective methods and algorithms to protect their customer's sensitive data and financial transactions. New cryptography algorithms are constantly being invented to meet the demands of diverse applications. However, any proposed algorithm must be trustworthy enough to meet the requisite standards. Various analyses have been conducted to assess the efficacy of encryption methods [1]–[3]. In reference [4], the effectiveness of four encryption schemes (AES, Blowfish, RC6, and 3DES) was compared. Keyspace, CPU workload, encryption time, and power consumption were all considered. The results of these assessments indicated that, in comparison to the other encryption algorithms, AES emerged as the most efficient and effective. Reference [5] presents an evaluation of the performance of two encryption techniques, symmetric encryption (including AES, DES, and Blowfish) and asymmetric encryption (such as RSA), considering both encryption and decryption time and throughput. This assessment was done using the Java programming language, utilizing image, binary, and text files. The results demonstrate that AES outperforms other encryption algorithms in all metrics. In [6], the present study examines and evaluates the randomness of lightweight ciphers within the context of the Internet of Things. To accomplish this, the research team employed five NIST tests (namely, the monobit test, the block test, the run test, the approximation entropy test, and the cumulative sums test) in addition to encryption time and microcontroller utilization unit assessments to evaluate the cipher algorithms. The results revealed that, concerning the randomness tests, Simeck and Kasumi performed comparably to AES and DES. However, in terms of encryption time and CPU usage, they outperformed AES and DES, demonstrating their potential as effective lightweight cyphers for IoT devices. Reference [7] presents a novel LabVIEW simulation designed to compare the encryption times of several symmetric encryption algorithms, including AES, DES, 3DES, and RC2, with those of the Advanced Encryption Package (AEP) program. Text files of varying lengths and key sizes were utilized to conduct this assessment. The findings indicate that the LabVIEW simulation improved the throughput and speed of cipher algorithms by an impressive 67%, underscoring its potential as a valuable tool for assessing and optimizing the performance of cryptographic algorithms. In [8], NIST developed a statistical suite comprising fifteen tests to evaluate encryption algorithms' randomness and key generation. While this battery of tests is widely used, its administration is notoriously ineffective, and its use involves some complexity. However, the suite's comprehensive nature ensures high reliability in assessing the randomness of encryption algorithms and key generation. It serves as a useful tool for assessing the cryptographic strength of algorithms and key generation mechanisms, enabling the identification of vulnerabilities and the improve-

[1]https://github.com/AREEG94FAHAD/22test

ment of security

The previous studies mentioned above only encompass some of the essential evaluations required to determine the effectiveness of encryption algorithms. To fill this gap, a novel software is proposed that employs over twenty tests to measure the performance of encryption algorithms, including hashing technology and key generation processes. The program comprises all NIST test suites, in addition to evaluations of other crucial factors, such as encryption and decryption times, mean square errors, correlations between the original and encrypted texts, single-to-noise ratios, peak signal-to-noise ratios, password spaces, and text entropies. By incorporating these comprehensive tests, the proposed software can provide a more robust assessment of encryption algorithm efficacy.

The outline of this paper is organized as follows: Section II presents a brief explanation of all the tests provided by the proposed software. Section III describes the main software files, the packets used to implement the proposed software, and how it operates. Section IV presents the software interface and main components. Section V provides a brief comparative analysis of the encryption algorithms used in the testing program. Finally, the conclusion of the study and its results are presented.

Contributions:

- Developed the evaluation software using Python and implemented twenty-two tests for assessing encryption algorithms.
- Conducted performance comparison among AES, Logistic Map, and ARC4 algorithms, identifying their strengths in security, efficiency, and effectiveness.
- Explored practical applications of the software for evaluating and comparing encryption algorithms in real-world scenarios.

II. Overview of Tests Applied in the Proposed Program

This section provides an overview of the numerous tests applied to the proposed program. The program is designed to evaluate the efficacy of encryption algorithms, hashing technology, and key generation processes, and more than twenty different tests were utilised to achieve this goal. In addition to incorporating all NIST test suites, the proposed program also evaluates various other factors such as encryption and decryption times, mean square errors, correlations between the original and encrypted texts, single-to-noise ratios, peak signal-to-noise ratios, password spaces, and text entropies. Each of these tests plays an essential role in ensuring the robustness and reliability of the encryption algorithms, and their comprehensive application makes the proposed program a potent tool for evaluating the efficacy of various encryption techniques. Table 1 contains a brief description of them.

III. Proposed System Architecture

The proposed software program architecture can be seen in Figure 1, which displays five primary files. The first file, ”test_a,” contains all the NIST test suites. The second file, named ”test_b,” contains the implemented code for the signal-to-noise ratio (SNR), peak signal-to-noise ratio (PSNR), and mean squared error (MSE) tests. The third file, ”test_c,” contains the histogram and correlation tests. For testing the performance and throughput of encryption and decryption, there is a file called ”test_d.” Finally, the fifth file, named ”examples,” includes the implementations of three encryption algorithms: symmetric types AES [17] and Chaotic Map (logistic) [18] and asymmetric types RSA in addition to SHA512. The proposed software serves to assess and compute the performance of each of the encryption methods. It utilizes several Python packages, and some tests require multiple Python packages. All the software packages used are freely available online and straightforward to use. Figure 2 illustrates all the packages used in each test. The software presented in this study computes the performance of certain tests, including MSE, PSNR, SNR, and correlation, only if the length of the original data is equal to the length of the encrypted data. If the lengths are unequal, the value will be labeled ”unequal size.” It is important to note that this limitation applies to some tests and not all. Nevertheless, it is essential to consider this limitation when interpreting the results obtained from the software, as it may affect the validity of the analysis. Overall, the proposed software architecture is well-organized and easy to navigate, with each test separated into files for easy access. Including several encryption methods and various tests provides a comprehensive way to evaluate the performance of encryption techniques. Additionally, freely available packages make the software accessible to a wider audience.

IV. Proposed System Interface and Algorithm Evaluation

Figure 3 illustrates the proposed software. When a user wishes to assess the encrypted data, they must fill out the two fields with the original and encrypted data, then click the ”run” button to show the test results. Moreover, the program lets users download the histogram findings as PNG files. The proposed software is versatile and can test many security algorithms. We tested the suggested program by applying tests to AES, Chaotic map (logistic), RSA, ARC4, and sHA-512. The data used was the message ”Hello world!”. All tests were implemented using an ASUS Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz 2.90 GHz with

TABLE I
DESCRIPTION OF STATISTICAL TESTS

Name	Description
Frequency (Monobit) Test	Used to determine the ratio of ones to zeros across an entire sequence. Passing this test is a prerequisite for all remaining tests [9].
Frequency Test within a Block	Use to evaluate the percentage of ones in blocks of M bits to determine if it is close to M/2, assuming randomness [8].
Run Test	Determines the total number of runs in a sequence and checks if the fluctuation between zeros and ones is within expectations for a random sequence [8].
The Longest Run of Ones in a Block test	Determines whether the longest run of ones in a sequence is comparable in length to what is expected in a random sequence [10].
The Binary Matrix Rank test	Determines if variable-length substrings of the sequence are linearly dependent on one another [8].
Discrete Fourier Transform (Spectral) test	Identifies periodic characteristics or repetitive patterns in the sequence using the discrete Fourier transform [11].
Non-overlapping Template Matching	Identifies generators that create an excessive number of instances of a particular periodic pattern [12].
The Overlapping Template Matching	Focuses on the number of times predetermined target strings appear in the sequence [8].
The Universal Statistical	Evaluates the compressibility of a sequence by measuring the number of bits between matching patterns [8].
The Linear Complexity	Measures the shortest length of the linear feedback register (LFSR) of each block to assess randomness in a sequence [8].
Serial	Assesses if the frequency of m-bit overlapping patterns in a sequence is comparable to what is expected in a random sequence [8].
Approximate Entropy	Determines if the observed frequency of overlapping blocks corresponds to the frequency predicted from a random sequence [8].
Cumulative Sums	Assesses whether the cumulative total of the partial sequences in a sequence is excessively high or low compared to expectations [8].
Random Excursions	Evaluates the duration of time spent in a given state during a random walk in the sequence [8].
Random Excursions Variant	Determines if the random walk follows the expected number of visits to each state [8].
Correlation	A statistical measure that indicates the degree of relationship or association between two variables. It quantifies how changes in one variable are related to changes in another variable [13].
Histogram analysis	A strong encryption algorithm is indicated by a histogram where the distribution appears uniformly random across the full range of possible value [14].
The Mean Square Error (MSE)	Measures how close the encrypted data is to the original, with lower values indicating higher similarity [15].
The Signal-to-Noise Ratio (SNR)	Detects the level of signal distortion caused by encryption, a lower SNR indicates the effectiveness of an encryption technique [15].
The Peak Signal-to-Noise Ratio (PSNR)	Measures the quality of an encryption algorithm by comparing the original and encrypted data, a lower PSNR indicates a more effective encryption algorithm [15].
Encryption and decryption speed	Evaluates the total time required to perform the encryption function [16].
Throughput	Measures the speed at which an algorithm can encrypt and decrypt data.

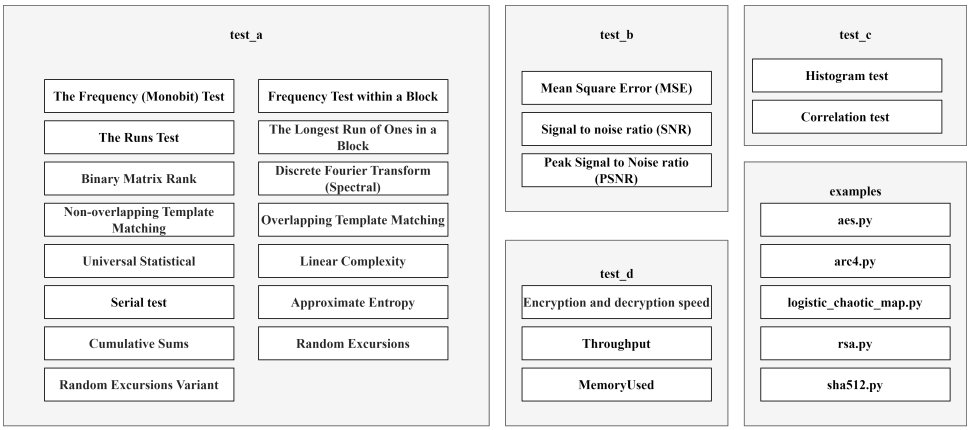

Fig. 1. Proposed software architecture

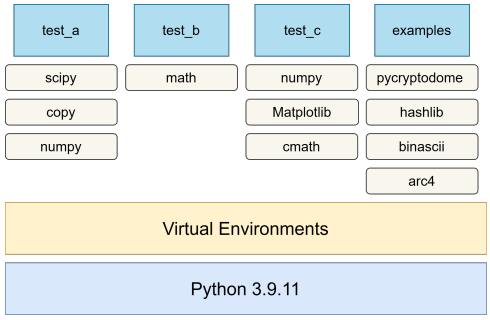

Fig. 2. Proposed Software Packets

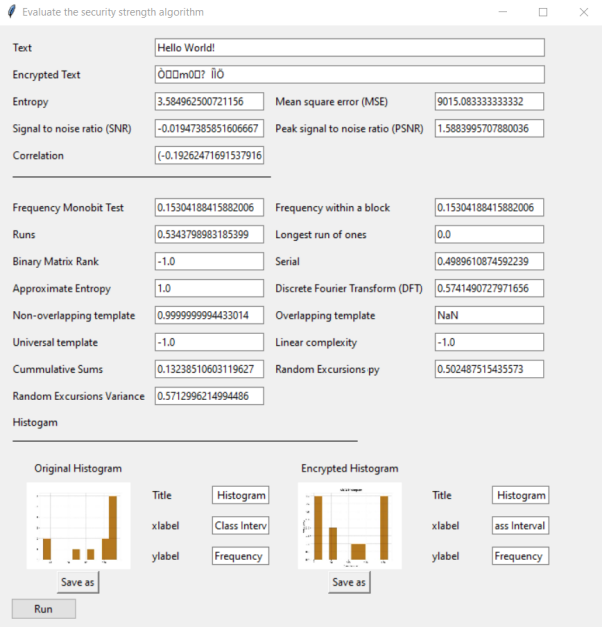

Fig. 3. Proposed System Interface

8 GB RAM and 256 SSD hard desk. It can be observed in Figure 4 that the symmetric algorithms, such as the logistic map, ARC4, and AES, outperform the asymmetric algorithm (RSA) in encryption/decryption speed. ARC4 and logistics were found to be more efficient in terms of throughput, as shown in Figure 5. In contrast, RSA showed less performance compared to the other algorithms. This is because RSA is an asymmetric encryption method that involves two keys, resulting in longer processing times. The superiority of symmetric algorithms, such as ARC4 and logistics, can be attributed to their simpler structure and the use of a single key for encryption and decryption, resulting in faster processing times. The proposed software was also used to measure the amount of memory required to encrypt the original data. Figure 6 shows that AES requires 24 bytes to encrypt the message, while RSA needs 256 bytes and SHA-512 uses 128 bytes. On the other hand, both ARC4 and logistic map achieve high performance with the same number of bytes produced for encryption and decryption. This highlights the advantage of using stream ciphers over symmetric and asymmetric algorithms in terms of memory usage. Table II displays the performance of the algorithms based on the randomness test. Logistics Map, AES, and ARC4 performed exceptionally well in the randomness test due to their design. They are engineered to generate sequences that closely resemble true randomness, making it hard for statistical tests to identify any patterns resulting in p-values above 0.01 across various tests.

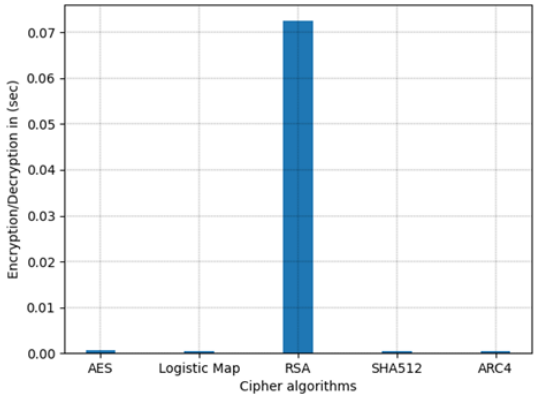

Fig. 4. Encryption/Decryption Speen in (sec)

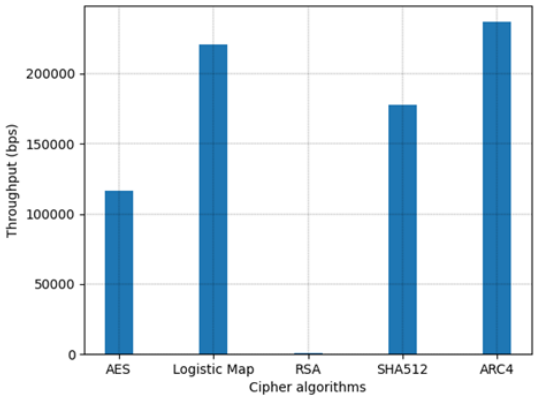

Fig. 5. Throughput in (bps)

V. Conclusion

This research presents a new software that uses over twenty-two checks to facilitate the assessment of cipher algorithms. Tinker, a Python-based framework, was used to create this application. Several cryptographic algorithms, including three stream ciphers (AES, Logistic Map, and ARC4) and one asymmetric cipher (RSA) in addition to SHA512, were analyzed

TABLE II
RANDOMNESS TESTS RESULTS

Method Name	AES	RSA	ARC4	Logistic Map	SHA512
Frequency (Monobit)	0.1939	1e-7	0.1530	0.6830	0.0005
Frequency Test within a Block	0.1572	0.0040	0.1530	0.6830	0.1218
Runs Test	0.0024	2.8e-51	0.5343	0.5283	0.4903
Longest-Run-of-Ones in a Block	0.0773	6.7504e-44	0.0	0.0	3.417e-26
Binary Matrix Rank	-1.0	0.0015	-1.0	-1.0	0.0391
Discrete Fourier Transform (Spectral)	0.5962	1.6e-11	0.5741	0.1897	0.0008
Non-overlapping Template Matching	0.9999	0.8675	0.9999	0.999	0.9860
Overlapping Template Matching	NaN	0.8865	NaN	NaN	NaN
Universal Statistical	-1.0	-1.0	-1.0	-1.0	-1.0
Linear Complexity	-1.0	0.0619	-1.0	-1.0	0.9856
Serial Test	-1.0	8.7e-143	0.4989	0.4985	9.606e-8
Approximate Entropy	1.0	1.607e-10	1.0	1.0	0.999
Cumulative Sums	0.1665	4.66e-7	0.1323	0.8864	0.0011
Random Excursions	0.84916	0.5494	0.5712	0.2872	0.9625
Random Excursions variant	0.9098	1.0	0.5712	0.3661	0.9098

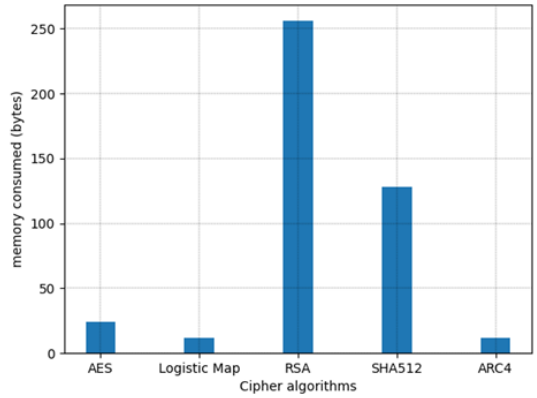

Fig. 6. Memory consumed

using the given software in order to determine the efficacy of the suggested software. The proposed software has been shown to be highly effective for testing and analyzing the performance of the system's security algorithms. In addition to providing the program, a simple comparison between the testing procedure's security algorithm and the results reveals that AES, Logistic, and ARC4 performed well in all tests. In addition, logistics offers lower complexity (O(n)) and memory consumption throughout the encryption and decryption process, leading them to promote its adoption for networks with constrained resources, such as the IoT and WSN.

REFERENCES

[1] M. Jangjou and M. K. Sohrabi, "A comprehensive survey on security challenges in different network layers in cloud computing," *Archives of Computational Methods in Engineering*, vol. 29, no. 6, pp. 3587–3608, 2022.

[2] E. Ukwandu, M. A. Ben-Farah, H. Hindy, M. Bures, R. Atkinson, C. Tachtatzis, I. Andonovic, and X. Bellekens, "Cybersecurity challenges in aviation industry: A review of current and future trends," *Information*, vol. 13, no. 3, p. 146, 2022.

[3] M. S. M. Shah, Y.-B. Leau, Z. Yan, and M. Anbar, "Hierarchical naming scheme in named data networking for internet of things: A review and future security challenges," *IEEE Access*, vol. 10, pp. 19 958–19 970, 2022.

[4] C. Peng, X. Du, K. Li, M. Li *et al.*, "An ultra-lightweight encryption scheme in underwater acoustic networks," *Journal of Sensors*, vol. 2016, 2016.

[5] J. Awotunde, A. Ameen, I. Oladipo, A. Tomori, and M. Abdulraheem, "Evaluation of four encryption algorithms for viability, reliability and performance estimation," *Nigerian Journal of Technological Development*, vol. 13, no. 2, pp. 74–82, 2016.

[6] M. Qasaimeh, R. S. Al-Qassas, and S. Tedmori, "Software randomness analysis and evaluation of lightweight ciphers: the prospective for iot security," *Multimedia Tools and Applications*, vol. 77, pp. 18 415–18 449, 2018.

[7] I. H. Latif, "Time evaluation of different cryptography algorithms using labview," in *IOP Conference Series: Materials Science and Engineering*, vol. 745, no. 1. IOP Publishing, 2020, p. 012039.

[8] A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert *et al.*, *A statistical test suite for random and pseudorandom number generators for cryptographic applications*. US Department of Commerce, Technology Administration, National Institute of …, 2001, vol. 22.

[9] J. Soto, "Statistical testing of random number generators," in *Proceedings of the 22nd national information systems security conference*, vol. 10, no. 99. Citeseer, 1999, p. 12.

[10] A. A. Zakaria, H. A. Rani, and N. A. Nik, "Enhanced statistical analysis evaluation using csm randomness test tool," *International Journal of Cryptology Research*, pp. 52–75, 2019.

[11] J. Borkowski, J. Mroczka, A. Matusiak, and D. Kania, "Frequency estimation in interpolated discrete fourier transform with generalized maximum sidelobe decay windows for the control of power," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 3, pp. 1614–1624, 2020.

[12] K. Li, J. Zhang, P. Li, A. Wang, and Y. Wang, "Parallel implementation of the non-overlapping template matching test using cuda," *China Communications*, vol. 17, no. 8, pp. 234–241, 2020.

[13] D. Dimov and Y. Tsonev, "Result oriented time correlation between security and risk assessments, and individual environment compliance framework," in *Information Systems and Technologies to Support Learning: Proceedings of EMENA-ISTL 2018 2*. Springer, 2019, pp. 373–383.

[14] A. Derhab, M. Belaoued, I. Mohiuddin, F. Kurniawan, and M. K. Khan, "Histogram-based intrusion detection and filtering framework for secure and safe in-vehicle networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 3, pp. 2366–2379, 2021.

[15] A. Mohanarathinam, S. Kamalraj, G. Prasanna Venkatesan, R. V. Ravi, and C. Manikandababu, "Digital watermarking

techniques for image security: a review," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, pp. 3221–3229, 2020.

[16] A. Al-Hyari, K. Aldebei, Z. A. Alqadi, B. Al-Ahmad *et al.*, "Rotation left digits to enhance the security level of message blocks cryptography," *IEEE Access*, vol. 10, pp. 69 388–69 397, 2022.

[17] K. G. Salim, S. M. K. Al-alak, and M. J. Jawad, "Improved image security in internet of thing (iot) using multiple key aes," *Baghdad Science Journal*, vol. 18, no. 2, pp. 0417–0417, 2021.

[18] E. A. Al-Bahrani and R. N. Kadhum, "A new cipher based on feistel structure and chaotic maps," *Baghdad Science Journal*, vol. 16, no. 1, pp. 270–280, 2019.