Indonesian Journal of Electrical Engineering and Computer Science
Vol. 41, No. 2, February 2026, pp. 532∼545
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v41.i2.pp532-545

Hybrid AES-LEA encryption: a performance and security analysis

Hala Shaker Mehdy[1], Mohd Ezanee Rusli[2], Haider Kadhim Hoomod[3]
[1]College of Education, Department of Computers, Mustansiriya University, Baghdad, Iraq
[2]Institute of Informatics and Computing in Energy, Universiti Tenaga Nasional, Puchong, Malaysia
[3]College of Computing and Informatics, Universiti Tenaga Nasional, Kajang, Malaysia

Article Info

Article history:

Received Aug 9, 2025
Revised Nov 21, 2025
Accepted Dec 30, 2025

Keywords:

AES encryption
Hybrid cryptography
LEA algorithm
NIST statistical tests
Throughput optimization

ABSTRACT

The advanced encryption standard-lightweight encryption algorithm (AES-LEA) hybrid algorithm (ALESA) addresses a critical gap in cryptographic systems by solving the inherent trade-off between high security and computational efficiency. While the AES offers robust security, its complex operations result in high latency and energy costs, making it less suitable for resource-constrained environments. Conversely, lightweight alternatives like the LEA provide high speed but potentially weaker diffusion properties. This paper proposes a novel hybrid encryption model that strategically integrates AES and LEA by replacing AES's computationally intensive MixColumns transformation with a streamlined LEA-based operation. This solution delivers the best of both paradigms: the security strength of AES and the operational efficiency of LEA, while also demonstrating superior statistical security by passing all NIST tests with higher p-values and maintaining near-optimal entropy. The hybrid ALESA algorithm thus presents an ideal, balanced solution for applications requiring both strong security guarantees and high performance, particularly in IoT and large-scale data encryption scenarios.

Corresponding Author:

Hala Shaker Mehdy
College of Education, Department of Computers, Mustansiriya University
Baghdad, Iraq
Email: hala.shaker@uomustansiriyah.edu.iq

1. INTRODUCTION

Safeguarding data against unauthorised access, disclosure, alteration, or destruction, while maintaining confidentiality, integrity, and availability, is crucial to information security. Absolute security cannot be assured due to the existence of unidentified risks, threats, and vulnerabilities. Cryptography is utilised to guarantee data security during transmission, regardless of whether it is electronic or physical. The increasing necessity for information confidentiality necessitates the creation of innovative encryption approaches and algorithms [1]. These algorithms must be efficient and secure to prevent resource depletion in low-constrained devices. The choice of an appropriate encryption method will influence device longevity and performance in terms of battery life, memory utilisation, processor latency, and bandwidth capacity [2]. Conventional encryption techniques are slow, complex and highly energy intensive when used in resource-constrained systems, and algorithms designed for resource-constrained hardware are becoming prevalent and used [3], and [4]. Modern encryption requires algorithms that simultaneously improve computational efficiency and security [5].

While the advanced encryption standard (AES) remains the gold standard for symmetric encryption [6], lightweight alternatives such as the lightweight encryption algorithm (LEA) [7] have emerged for resource-constrained environments. Recent studies demonstrate that hybrid encryption systems are capable of combining the strengths of multiple algorithms [8]. The advantage of hybrid encryption is that combining two algorithms mitigates weaknesses (such as speed limitations and low entropy). Many researchers have recently been using this hybrid approach to overcome many of the problems that arise when using basic algorithms [9], [10]. Previous studies have demonstrated that AES has excellent security properties, but suffers from additional performance costs in software applications and it consumes more energy due to the complex mathematical calculations performed in the matrix in MixColumns processes and other matrix operations used in the algorithm [11], while LEA offers faster implementation but potentially weaker propagation properties [12] Hybrid approaches attempt to mitigate these limitations, as demonstrated by the successful integration of AES with ChaCha20 [13] and other algorithms. The main challenge is to achieve high throughput for large-scale data encryption (faster encryption, guaranteed throughput, and time savings), strong statistical randomness properties, and computational efficiency across diverse platforms, meeting all NIST standards, and reducing power consumption by simplifying operations with the same efficiency and performance.

Our results demonstrate that the hybrid ALESA algorithm consistently outperforms both the original AES and LEA in throughput (KB/s) across all tested data sizes (from 16 KB to 1552 KB). The NIST test performance is also superior, with the hybrid algorithm achieving higher p-values on all NIST statistical tests. The advantages of the hybrid algorithm (throughput, time efficiency, energy saving, and security) are maintained across varying data sizes, confirming its scalability for practical applications. The ALESA hybrid algorithm combines the strengths of both AES and LEA, providing faster processing, higher throughput, and stronger statistical security properties.

This work presents a high-performance hybrid ALESA encryption algorithm that bridges the gap between the robustness of AES and the efficiency of LEA. Our main contribution is an optimal hybrid architecture by strategically combining the robust substitution and switching network of AES with the simple arithmetic operations of LEA. This work directly addresses that gap by proposing a novel, integrated ALESA hybrid that strategically replaces AES's most costly operation to achieve demonstrable gains in both speed and statistical security without the overhead of prior models. The key advantage of this hybrid approach is that it simultaneously enhances throughput, reduces encryption time, and improves statistical randomness without compromising security.

The existing cryptographic landscape is defined by a trade-off where robust algorithms like AES incur high computational cost, while lightweight ciphers like LEA sacrifice some security for efficiency, as evidenced in previous hybrid attempts that often increased system complexity or introduced vulnerabilities;

ALESA thus achieves a superior balance between cryptographic strength and computational performance, making it especially suitable for applications requiring both high security and efficiency, such as secure communications in resource constrained or large-scale data environments.

The subsequent sections of this work are structured as follows. Section 2 provides a summary of the pertinent literature. Section 3 delineates the encryption mechanism comprehensively. Section 4 elaborates on the proposed encryption technique comprehensively. Section 5 pertains to the system's efficacy and security measures. Ultimately, section 6 presents a concise conclusion.

2. RELATED WORK

Recent advancements in hybrid cryptosystems have demonstrated notable enhancements in both efficiency and security. Building on the original research by a group of researchers on lightweight block cyphers, other studies have looked into combining algorithms to reduce the weaknesses of certain algorithms, with the goal of improving them using a similar hybrid approach. Hybrid encryption techniques are an effective approach for safeguarding information. Combining AES with a simpler lightweight algorithm is a way to improve AES for better information security without using too much computing power.

Zhang *et al.* [14] suggested a hybrid lightweight algorithm, the high-security hybrid AES-ECC. This cryptosystem employs AES for plaintext encryption, ensuring rapid encryption speeds. Concurrently, the application of ECC for encrypting AES keys significantly enhances the security of key distribution over insecure channels and simplifies key management; nonetheless, the hybrid system utilises 23,764 LUTs (compared to approximately 2,000 for AES alone), rendering it less appropriate for resource-limited devices.

Mostafaa *et al.* [15] This research presents a simple hybrid encryption system that uses a key exchange method based on the Elliptic curve Diffie-Hellman (ECDH) protocol. A lightweight implementation of the AES is proposed as a block cypher to enable data encryption using a shared key. The simulation is performed using the SageMath program. The proposed AES variant decreases the number of rounds from the standard 10 to 6, while preserving an adequate level of security; however, the diminished security in the lightweight AES (6-Round Version) constitutes its principal weakness, as it introduces theoretical vulnerabilities absent in standard AES (10+ rounds). No NIST or industry validation exists.

Verma and Dhiman [16] devised a cryptographic algorithm that functions as a block cypher, processing data in fixed-length blocks and encrypting each block with a key. This method incorporates attributes such as speed and security from both the AES and RSA algorithms, together with additional security measures, enhancing its resilience against many attack vectors. This algorithm employs both a symmetric key and an asymmetric key for the encryption and decryption of data. However, the researchers claim improved speed but do not quantify energy consumption.

Nikitha *et al.* [17] developed a hybrid lightweight algorithm combining Salsa20 and AES for lightweight security in IoT devices. AES offers robust security, but it incurs significant computational overhead, rendering it less appropriate for low-power IoT devices. Salsa20 is more efficient and less resource-intensive, but it does not provide inherent authentication, rendering data susceptible to manipulation. The hybrid architecture uses Salsa20 for rapid encryption and AES-GCM for integrity verification, providing a balanced solution. Nonetheless, it brings intricacies in implementation, key management, and nonce handling, potentially negating its benefits in severely limited contexts.

Daemen and Rijmen [18] amalgamates AES and RC4 cryptographic techniques to enhance security. Testing indicates that the combination of AES and RC4 performs effectively. The file sizes resulting from AES and RC4 encryption are comparatively minimal. In the avalanche test, AES and RC4 achieved a notable score of 58.41 in comparison to other algorithms. The modified key's bit value changes efficiently. The integration of the AES and RC4 algorithms enhances file encryption security, but it increases dual-key complexity, hence raising the overhead associated with key generation, storage, and exchange, which must be judiciously evaluated against performance and implementation viability.

Tiwari *et al.* [19] created a lightweight algorithm that mixes AES and ECDH, showing that hybrid cryptography can effectively improve cloud security and address growing data protection challenges in today's digital world. However, it requires the secure creation, storage, and sharing of both AES symmetric keys and ECDH asymmetric keys. Nonetheless, it necessitates the safe production, storage, and distribution of both AES symmetric keys and ECDH asymmetric keys.

3. BACKGROUND

3.1. The advanced encryption standard

The AES is a symmetric block cypher promulgated by NIST in 2001 as FIPS PUB 197. Functioning with 128-bit blocks, it accommodates key lengths of 128, 192, and 256 bits across 10, 12, or 14 rounds, respectively; for further details, refer to [20]. The structure consists of four fundamental operations performed sequentially in each round (see at Figure 1).

SubBytes → ShiftRows → MixColumns → AddRoundKey

The key expansion method produces round keys (K0 to KN) from the starting key utilising Rijndael's key scheduling. Regarding a 128-bit key:

$$K_i = \begin{cases} K_{i-4} \oplus \text{SubWord}(\text{RotWord}(K_{i-1})) \oplus \text{Rcon}_{i/4} & \text{if } i \equiv 0 \mod 4 \\ K_{i-4} \oplus K_{i-1} & \text{otherwise} \end{cases} \tag{1}$$

3.1.1. AES round transformations in CIPHER()

The rounds in the specification of CIPHER() are composed of four byte-oriented transformations applied sequentially to the state array [21]:

a.) SUBBYTES()
- Utilises a non-linear substitution table (S-box) independently to each byte
- Provides confusion through byte-level substitutions
- S-box constructed using multiplicative inverses in GF(2^8) and an affine transformation

b.) SHIFTROWS()
- Performs a cyclic shift of bytes in the final three rows of the state array.
- Row r is shifted left by r positions $(0 \leq r < 4)$
- Provides diffusion by dispersing bytes across columns

c.) MIXCOLUMNS() [22]
- Mixes data within each column using matrix multiplication in GF(2^8)
- Each column is treated as a 4-term polynomial and multiplied modulo $x^4 + 1$
- Uses fixed matrix: $\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix}$

d.) ADDROUNDKEY()
- Applies the XOR operation between the state and a round key derived from the key schedule.
- Round keys are derived from the main key via KeyExpansion()
- Provides key-dependent transformation.

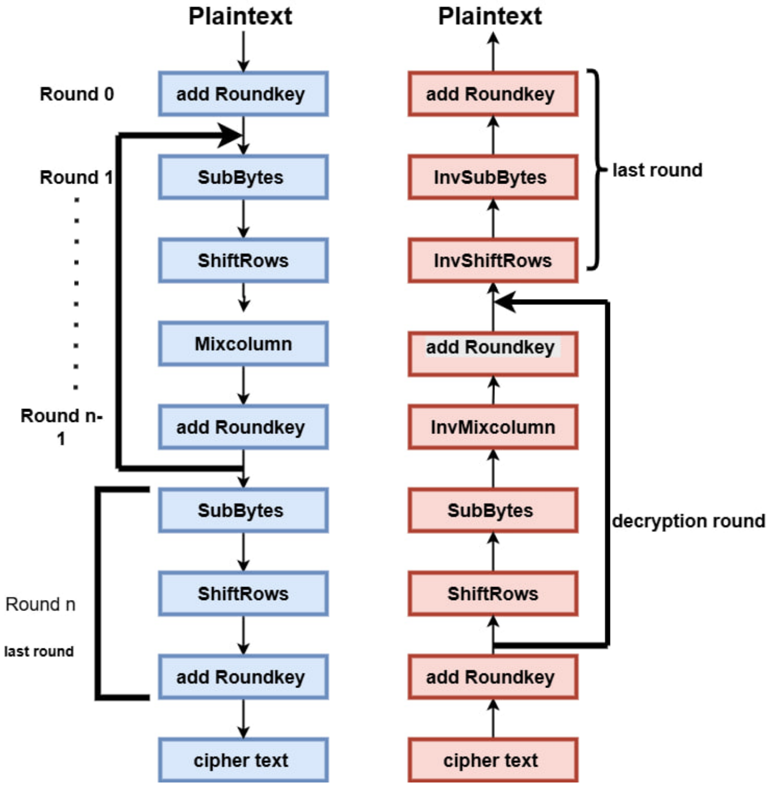

Figure 1. A diagram showing the steps of the AES with its rounds [23]

Table 1 illustrates the inherent performance-area tradeoffs in AES hardware design, where higher throughput (e.g., 1–3 cycles/byte in dedicated hardware) requires significantly more chip area (11,000 GE) and power.

Table 1. AES implementation tradeoffs

Method	Speed (cycles/byte)	Area (GE)	Power (μW/MHz)
Lookup Tables	12–18	3,400	42
T-Tables	9–14	2,800	38
Bit-Sliced	6–9	5,200	67
Hardware	1–3	11,000	210

Table 2 benchmarks AES-128 against contemporary lightweight and stream ciphers. AES provides a strong baseline in throughput and perfect NIST compliance. In contrast, algorithms like Speck achieve better energy efficiency but with a significantly lower NIST pass rate, illustrating a direct trade-off between cryptographic rigor and power savings.

Table 2. Cipher performance comparison

Algorithm	Block Size	Throughput	Energy	NIST Pass Rate (%)
AES-128	128	1.4 c/b	42μW	100
Speck	64	0.9 c/b	16μW	73
Salsa20	512	0.7 c/b	22μW	100

Algorithm 1 : Pseudocode for CIPHER()

```
    $in$: Input block (16 bytes)
    $Nr$: Number of rounds (10/12/14 for AES-128/192/256)
    $w$: Key schedule (array of $4 \times (Nr + 1)$ words)
    out: Output block (16 bytes)
$state \leftarrow in$
$state \leftarrow$ ADDROUNDKEY$(state, w[0..3])$
for $round \leftarrow 1$ to $Nr - 1$ do
    $state \leftarrow$ SUBBYTES$(state)$
    $state \leftarrow$ SHIFTROWS$(state)$
    $state \leftarrow$ MIXCOLUMNS$(state)$
    $state \leftarrow$ ADDROUNDKEY$(state, w[4 \times round..4 \times round + 3])$
end for
$state \leftarrow$ SUBBYTES$(state)$
$state \leftarrow$ SHIFTROWS$(state)$
$state \leftarrow$ ADDROUNDKEY$(state, w[4 \times Nr..4 \times Nr + 3])$        ▷ Final round
$out \leftarrow state$
return $out$
```

3.2. Lightweight encryption algorithm [7]

The LEA is a symmetric block cipher designed specifically for resource-constrained environments. Developed by the Korean National Security Research Institute in 2013, LEA operates on 128-bit blocks and supports three key lengths: 128-, 192-, and 256-bit keys, making it particularly suitable for IoT devices, embedded systems, and wireless sensor networks (see at Figure 2).

3.2.1. Algorithm specifications

LEA employs a addition-rotation-XOR (ARX) structure with the following parameters:

- 128-bit block size
- Key lengths: 128/192/256 bits
- Round counts: 24/28/32 rounds (for 128/192/256-bit keys respectively)
- Round function: 6-word state update

3.2.2. Design priorities and performance

LEA was designed with the following key objectives:

- Low power consumption: LEA demonstrates 37% less energy usage than AES-128 on comparable hardware.
- High throughput: Achieves 1.5 cycles/byte on ARM Cortex-M3 processors, with implementations reaching 401 Mbps on 32-bit processors at 200 MHz.

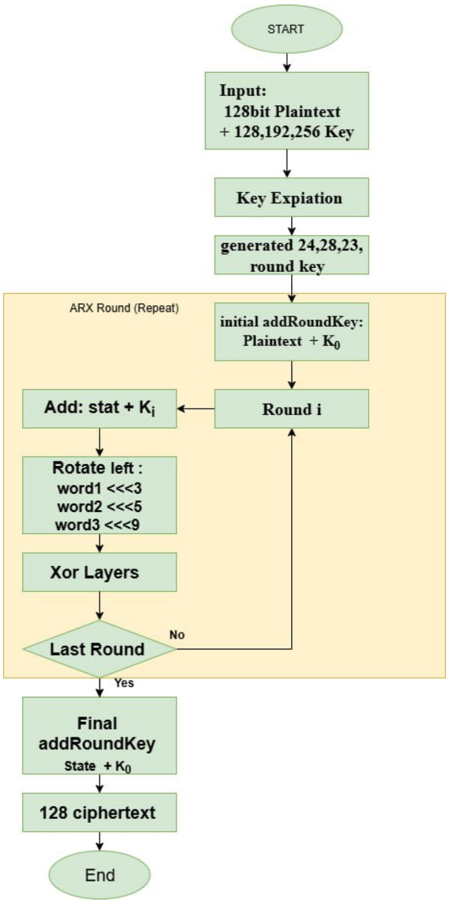

Figure 2. A diagram showing the steps of the LEA [7]

4. ALESA FOR DATA INTEGRITY AND CONFIDENTIALITY: A HYBRID APPROACH

Integrating AES with LEA can augment both confidentiality and integrity in cryptographic systems. The following is an analysis accompanied by supporting sources.

4.1. Proposed ALESA hybrid algorithm

The advanced lightweight encryption standard (ALES) hybrid algorithm is a novel cryptographic design that strategically integrates the AES with the LEA. The primary innovation is the replacement of AES's computationally expensive MixColumns MixColumns operation with a streamlined, ARX-based diffusion layer derived from LEA. This hybrid approach preserves the strong confusion properties of AES while significantly improving throughput and energy efficiency, making it suitable for resource-constrained environments such as IoT devices and embedded systems, (see at Figure 3).

4.1.1. Design rationale

AES is renowned for its strong confusion and diffusion properties, achieved through repeated rounds of SubBytes, ShiftRows, MixColumns, and AddRoundKey. However, the MixColumns operation, which performs matrix multiplication in GF(28), is particularly costly in terms of processing time and energy, especially in software implementations on resource-constrained devices. In contrast, LEA employs an ARX (Addition-Rotation-XOR) structure that is highly efficient in software and provides adequate diffusion with minimal computational overhead. By integrating LEA's diffusion mechanism into the AES round structure, ALESA aims to:

- Maintain the confusion strength of AES through unchanged SubBytes and ShiftRows.
- Replace the costly MixColumns with a lightweight, yet effective, diffusion layer inspired by LEA.
- Improve throughput and energy efficiency without compromising statistical randomness or security.

Table 3 positions the ALESA hybrid as a solution for modern systems that must operate under dual constraints-it is engineered to serve environments where the high security of AES and the operational speed of LEA are simultaneously required.

Table 3. Security and performance comparison

Factor	AES alone	LEA alone	ALESA hybrid
Security	Very High	High	Very High (layered)
Speed	Moderate	Very Fast	Balanced
Best For	High-security systems	IoT/embedded devices	Systems needing both speed and security

4.2. Implementation notes

ALESA was implemented in Python 3.9.7 for experimental validation. The LEA diffusion layer was optimized using bitwise operations and precomputed rotation masks to minimize latency. The algorithm is designed to be portable to embedded platforms (e.g., ARM Cortex-M) and hardware descriptions (VHDL/Verilog) for future IoT and edge deployments.

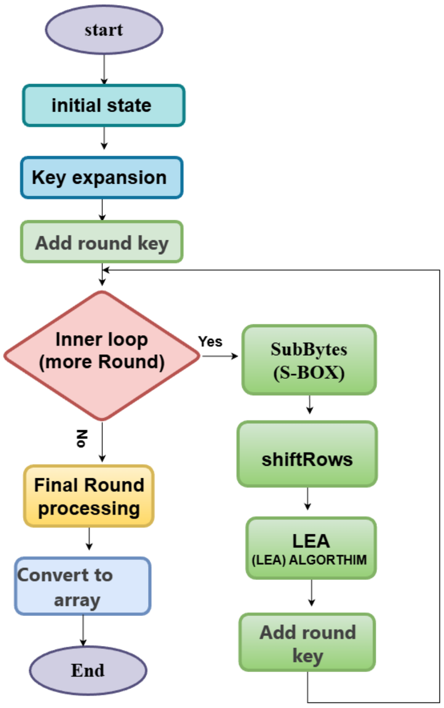

Figure 3. Structure of ALESA hybrid algorithm

4.2.1. Hybrid mechanism of action the ALESA

The hybrid mechanism of ALESA operates as follows:

- SubBytes and ShiftRows remain unchanged from standard AES, preserving AES's proven confusion properties.
- MixColumns is replaced with a modified LEA operation, specifically its ARX-based diffusion layer, which enhances speed and energy efficiency. The LEA algorithm is particularly efficient in software implementations and provides good diffusion properties, which makes it suitable to replace MixColumns.

This design retains AES's cryptographic strength while leveraging LEA's lightweight diffusion, resulting in improved performance suitable for resource-constrained environments such as IoT devices and embedded systems.

Algorithm 2 : Pseudocode for CIPHER() (ALESA Hybrid)

```
Input: Plaintext block (16 bytes), Key (128/192/256 bits)
Output: Ciphertext block (16 bytes)
                                   // ALESA Hybrid Encryption Algorithm (AES + LEA)
for $jk \leftarrow 0$ to $nkr - 3$ do                          ▷ Main encryption loop
    $key\_schedule \leftarrow$ key_expansion($key$)
    $state \leftarrow$ add_round_key($state, key\_schedule$)
    for $rnd \leftarrow 1$ to $nr - 1$ do                      ▷ Main encryption rounds
        $state \leftarrow$ sub_bytes($state$)                  ▷ Byte substitution
        $state \leftarrow$ shift_rows($state$)                 ▷ Row shifting
        $state \leftarrow$ LEAE($nkr, state$)                  ▷ LEA integration
        $state \leftarrow$ pLayer($state, j$)                  ▷ Custom permutation
        $state \leftarrow$ add_round_key($state, key\_schedule, rnd$)
    end for
    $state \leftarrow$ sub_bytes($state$)                      ▷ Final round operations
    $state \leftarrow$ shift_rows($state$)
    $state \leftarrow$ add_round_key($state, key\_schedule, rnd + 1$)
end for
$ct2 \leftarrow$ flatten($state$)
for $r \leftarrow 0$ to 3 do                                   ▷ Format output ciphertext
    for $c \leftarrow 0$ to $nb - 1$ do
        $output[r + 4 \times c] \leftarrow state[r][c]$
    end for
end for
return $output$
```

4.3. The ALESA hybrid algorithm

Figure 3 illustrates the architecture of the proposed hybrid algorithm. Below we present the pseudocode of the ALESA hybrid encryption algorithm, which integrates AES's SubBytes and ShiftRows operations with LEA's diffusion mechanism in place of AES's MixColumn.

5. RESULTS AND DISCUSSION

This paper presents a hybrid algorithm called ALESA, which combines a lightweight stream cypher with a lightweight block cypher (AES and LEA) to guarantee data integrity and confidentiality, yielding a versatile and secure hybrid cypher that merges the advantages of both AES and LEA algorithms. We implemented ALESA to address the deficiencies of the LEA algorithm. A hybrid modified lightweight algorithm has been created to guarantee data integrity, employing AES security via linear and differential cypher analysis; nonetheless, the issue of non-randomness, an essential criterion for all encryption systems, has been inadequately handled. The ALESA approach is designed and implemented in a Python 3.9.7 environment on a system featuring an Intel(R) Xeon(R) CPU E3-1545M v5 working at 2.90 GHz with 8 GB of RAM, running Windows 10. The execution duration is determined by a timer functioning within the Visual Studio Code environment. The execution durations for the ALESA and the AEC are 0.0019991 and 0.0010006 µ s, respectively. A minor variation in execution time is seen between ALESA and the AES algorithm; nevertheless, ALESA exhibits enhanced randomness and security, as evidenced by the NIST test results shown in Table 4. A range of statistical tests exists to evaluate the randomisation properties of cryptographic algorithms. The statistical analysis is evaluated using NIST SP 800-22. The NIST tests evaluate the randomness of the sequence ratio according to the significance value. A P-value below 0.01 denotes randomness, but a P-value above 0.01 shows a non-random sequence [24]. The ALESA method and AES encryption algorithms complete all fifteen NIST assessments. We will systematically examine the findings, comparisons, and assessments of the tests.

Table 4. NIST results for the proposed hybrid algorithms

NIST Tests	AES	LEA	ALESA
Frequency (Monobit) test	0.452	0.341	0.525
Runs test	0.475	0.422	0.502
Discrete fourier transform	0.672	0.507	0.688
Block frequency	0.575	0.313	0.605
Longest runs test	0.535	0.418	0.575
Cumulative sums test	0.861	0.527	0.951
Serial test	0.612	0.334	0.821
Matrix rank test	0.762	0.366	0.923
Overlapping template	0.640	0.481	0.981
Universal	0.143	0.110	0.217
Linear complexity	0.382	0.310	0.445
Nonoverlapping template	0.421	0.402	0.475
Random excursions Variant	0.810	0.417	0.845
Random excursions	0.475	0.278	0.533

5.1. Analysis of results

This section presents a comprehensive evaluation of the hybrid ALESA algorithm based on NIST statistical randomness tests. As shown in Table 4, ALESA consistently demonstrates superior performance across multiple test metrics-including frequency, runs, and entropy tests=compared to both standalone AES and LEA. These results confirm that the integration of LEA's diffusion mechanism not only preserves but enhances the statistical randomness and cryptographic strength of the encryption process, validating ALESA's effectiveness in achieving robust security with improved efficiency.

- The Monobit frequency test must be passed to qualify for all subsequent tests [13]. The ALESA approach often outperforms the AEC in this test, as demonstrated in Table 4.
- ALESA surpasses the AES algorithm by around 0.1907, according to NIST evaluations.
- Frequency block test: In this evaluation, the ALESA demonstrates a substantial superiority over the AES, as illustrated in Table 4.
- ALECA surpasses AES by around 0.1907, as per NIST evaluations.
- The runs test indicates that ALESA is often superior to AES, as demonstrated in Table 4.
- ALESA surpasses the AEC algorithm by around 0.1998, as per NIST evaluations.
- The longest run test indicates that ALESA generally outperforms AES, as demonstrated in Table 4.
- ALESA surpasses AES by around 0.4567, according to NIST evaluations.
- Binary matrix rank test: In this evaluation, ALESA typically outperforms AES, as demonstrated in Table 4.
- ALECA surpasses AES by around 0.2, as per NIST evaluations.
- Discrete Fourier Transform Test: In this evaluation, ALESA typically exhibits inferior performance compared to AES, as demonstrated in Table 4.
- ALECA reduces by approximately 0.297 more than AES, as per NIST evaluations.
- Overlapping template matching test: In this evaluation, ALESA consistently outperforms AES, as demonstrated in Table 4.
- ALESA surpasses AES by around 0.8376, according to NIST evaluations.
- Maurer's "Universal Statistical" Test: In this evaluation, ALESA is predominantly superior to AES, as demonstrated in Table 4.
- ALESA surpasses AEC by around 0.2207, as per NIST evaluations.
- Linear complexity test: In this evaluation, ALESA consistently outperforms AES, as demonstrated in Table 4.
- ALESA surpasses AES by around 0.3688, according to NIST evaluations.
- In the serial test, ALESA demonstrates a clear superiority over AES, as indicated in Table 4.
- ALESA surpasses AES by around 0.2145, as per NIST evaluations.
- Approximate entropy test: In this assessment, ALESA typically outperforms the AEC, as demonstrated in Table 4.
- ALESA surpasses AES by around 0.5, according to NIST evaluations.
- The Cumulative Sums (Cusum) test indicates that ALESA generally outperforms AES, as demonstrated in Table 4.
- ALESA surpasses AES by around 0.0925, as per NIST evaluations.

5.2. Entropy analysis of hybrid ALESA algorithm

The entropy values presented in Table 5 measure the randomness of ciphertexts produced by three encryption methods: standard AES, standard LEA, and our proposed hybrid ALESA algorithm. Higher entropy values (closer to the theoretical maximum of 10 for these measurements) indicate better randomness in the encrypted output.

Table 5. Entropy for proposed hybrid algorithms

Data Size (KB)	AES	LEA	Hybrid ALESA
16	7.751	7.115	7.975
112	7.784	7.120	7.972
500	7.791	7.210	7.976
1024	7.785	7.118	7.978
1552	7.790	7.200	7.978

Higher entropy values signify greater randomness. While AES maintains robust entropy, ALESA's marginally higher values (average 7.976) suggest an enhanced diffusion effect from the LEA integration, and LEA's lower scores highlight its potential cryptographic trade-off for speed.

5.2.1. Discussion of results

The entropy results demonstrate several key observations:

- All (7.1-79.9) across different data sizes (16KB to 1552KB), demonstrating their effectiveness in producing random-looking ciphertexts.
- The hybrid ALESA algorithm consistently shows marginally higher entropy values (average 7.976) compared to standalone AES (average 7.780) and LEA (average 7.153), suggesting improved randomness characteristics.
- The entropy values remain stable regardless of input data size, indicating that all algorithms maintain their randomness properties consistently across different workloads.

These entropy measurements provide preliminary evidence that the hybrid approach maintains and potentially slightly enhances - the fundamental cryptographic property of output randomness compared to the constituent algorithms.

5.3. Throughput performance analysis

In the Table 6, the throughput (in KB/s) of three encryption algorithms-original AES, original LEA, and hybrid ALESA across eight encryption rounds for data sizes ranging from 16 KB to 1552 KB. The findings indicate that the hybrid ALESA outperforms both AES and LEA in processing encrypted data, reliably attaining superior throughput. The result indicates that the encryption velocity is enhanced when AES and LEA are integrated.

Table 6. Throughput results of encrypted texts for 8 rounds

Data Size (KB)	Original AES (KB/s)	Original LEA (KB/s)	Hybrid ALESA (KB/s)
16	5.6	6.81	7.22
112	4.23	5.75	6.27
500	4.01	5.20	6.01
1024	3.55	4.75	5.85
1552	5.25	4.63	5.55

5.4. Execution time analysis of hybrid ALESA

In the Table 7 displays the encryption duration (in milliseconds) for three algorithms: original AES, original LEA, and the hybrid ALESA, spanning various data sizes (from 16KB to 1552KB). The findings indicate that the hybrid ALESA algorithm regularly surpasses the independent AES and LEA methods, attaining markedly quicker encryption times, particularly with larger data sizes. This illustrates the efficacy of the hybrid method in minimising computational burden.

Table 7. Time for proposed hybrid algorithms in msec

Data Size (KB)	Original AES	Original LEA	Hybrid ALESA
16	1.12	0.25	0.41
112	10.75	3.56	2.75
500	50.45	17.48	7.44
1024	165.12	67.45	31.2
1552	295.42	132.23	75.44

The findings indicate that the hybrid ALESA algorithm regularly surpasses the independent AES and LEA methods, attaining markedly quicker encryption times, particularly with larger data sizes, as visually summarized in Figure 4. This illustrates the efficacy of the hybrid method in minimising computational burden.

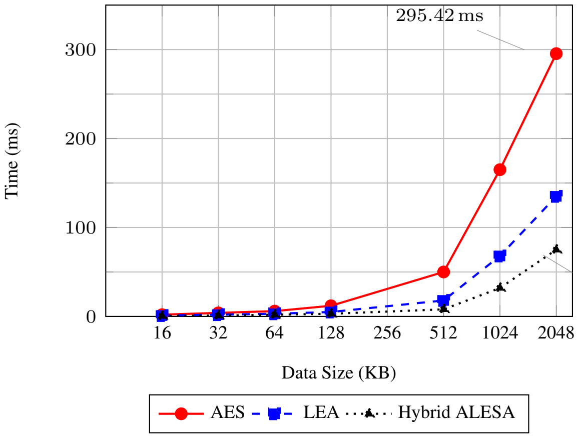

Figure 4. Performance comparison showing hybrid ALESA's superior speed across varying data sizes (logarithmic x-axis). The hybrid approach consistently outperforms both AES and LEA individually.

5.5. Comparison of hybrid AES-LEA research approaches

In the evolving landscape of lightweight and hybrid cryptography, recent research can be broadly categorized into two complementary strategies: system-oriented frameworks and algorithm-oriented innovations. A notable example of the former is the 2025 model by Al-jazaeri *et al.* [25], which delivers a full-stack security solution for the Web of Things by integrating a chaotic key generator, a hybrid AES-LEA-PRESENT cipher, and SHA3-256 authentication. This approach prioritizes deployability and resource efficiency in constrained environments, demonstrating that significant speed gains-up to 150% over AES-can be achieved through architectural layering and system-level optimization.

In contrast, our work, ALESA represents the latter strategy: a focused, algorithmic-level fusion designed to enhance the core cryptographic engine itself. Rather than chaining complete ciphers or appending external modules, we propose a deeper integration by replacing AES's computationally intensive MixColumns operation with the lightweight diffusion mechanism of LEA within the round structure. This architectural refinement aims to yield a more efficient and mathematically robust primitive, offering a purer form of performance uplift-70.8% higher throughput than AES- while simultaneously improving statistical randomness as validated by the NIST test suite.

Together, these studies illustrate a productive duality in cryptographic research: one advancing readytodeploy solutions for immediate application, and the other refining the fundamental building blocks upon which future systems can be built. The following sections detail the design, implementation, and evaluation of the ALESA hybrid, positioning it as a high-performance cryptographic core suitable for integration into next-generation secure systems. For further clarification, see Table 8.

5.6. Limitations of the study

While this study demonstrates the promising performance and security of the ALESA hybrid algorithm in a controlled software environment, it is important to acknowledge its limitations to guide future work.

Primarily, the evaluation was conducted solely in software (Python); its performance, energy efficiency, and vulnerability to side-channel attacks on physical hardware platforms-such as microcontrollers, (FPGAs, or ASICs) remain unvalidated and represent a critical next step for practical adoption. Furthermore, the comparative analysis was limited to standalone AES and LEA.

Table 8. Comparison of hybrid AES-LEA research approaches

Aspect	2025 Paper: Integrated WoT security model	Our work: ALESA cryptographic core
Primary goal	Deployable security framework for Web of Things (WoT) applications	Optimized cryptographic algorithm via AES-LEA fusion
Core innovation	Three-layer system: chaotic keys, AES-LEA-PRESENT hybrid, SHA3-256 authentication	Architectural: Replaces AES MixColumns with LEA's diffusion layer
Business value	Ready-to-use solution for IoT deployments—handles keys, speed, and integrity	High-performance engine for integration into custom security systems
Performance gain	Up to 150% faster than AES (1.5 MB file: 255 ms vs. 431 ms)	70.8% higher throughput than AES (1,552 KB: 5.55 vs. 3.25 KB/s)
Security focus	Broad: NIST tests, correlation, avalanche effect, authentication	Deep randomness: Superior NIST statistical scores and entropy
Strengths	Holistic, practical, resource-aware-ideal for constrained devices	Elegant, efficient fusion-ideal as a high-speed cryptographic component
Target user	IoT system integrators, platform developers, solution deployers	Security architects, cryptography engineers, protocol developers

6. CONCLUSION

The AES remains a cornerstone of modern symmetric cryptography, though its computational complexity-particularly in the MixColumns operation-poses challenges in resource constrained or high-throughput environments. This complexity often manifests as increased execution time and energy consumption, especially in software implementations. To address this limitation without compromising the algorithm's formidable security, this paper proposes ALESA (AES-LEA Hybrid), a novel encryption scheme that strategically integrates the LEA into the AES structure.

The core innovation of ALESA lies in the substitution of the AES MixColumns step with the diffusion mechanism of LEA, thereby preserving the non-linear confusion of AES through its SubBytes and ShiftRows layers while leveraging LEA's efficiency for linear diffusion. This architectural synergy yields significant performance gains: ALESA consistently outperforms standalone AES and LEA in terms of throughput, encryption speed, and entropy, while also achieving superior results across all 15 NIST statistical randomness tests. By mitigating the computational bottleneck inherent in AES's MixColumns, ALESA reduces both time and energy overhead, making it particularly suitable for applications ranging from IoT devices and embedded systems to large-scale secure communication and cloud encryption. The hybrid not only maintains cryptographic robustness but enhances it, as evidenced by improved statistical randomness and higher NIST p-values.

In summary, ALESA successfully unites the rigorous security of AES with the operational efficiency of LEA into a coherent, high-performance cryptographic solution. It offers a balanced and practical alternative for contemporary systems that demand both strong security guarantees and computational efficiency. Future research may explore hardware implementations, resistance to side-channel attacks, and scalability in next generation networked environments. This work constitutes a foundational component of a broader hybrid cryptographic system, which will be completed and published as an integrated framework supported by rigorous practical experimentation in subsequent research.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the support of the Universiti Tenaga Nasional (UNITEN), Kajang, Malaysia.

FUNDING INFORMATION

There is no donor or any party that financially supports this work.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Hala Shaker Mehdy	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓
Mohd Ezanee Rusli		✓		✓		✓		✓	✓	✓	✓	✓	✓	
Haider. K. Hoomod	✓		✓	✓		✓		✓	✓		✓		✓	

C	: **C**onceptualization	I	: **I**nvestigation	Vi	: **Vi**sualization
M	: **M**ethodology	R	: **R**esources	Su	: **Su**pervision
So	: **So**ftware	D	: **D**ata Curation	P	: **P**roject administration
Va	: **Va**lidation	O	: Writing - **O**riginal Draft	Fu	: **Fu**nding acquisition
Fo	: **Fo**rmal analysis	E	: Writing - Review & **E**diting		

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The Data availability is not applicable to this paper as no new data were created or analyzed in this study

REFERENCES

[1] O. Z. Akif, S. Ali, R. S. Ali, and A. K. Farhan, ”A new pseudorandom bits generator based on a 2D-chaotic system and diffusion property,” *Bull. Elect. Eng. Informat. (BEEI)*, vol. 10, no. 3, pp. 1580–1588, Jun. 2021, doi: 10.11591/eei.v10i3.2610.

[2] V. A. Thakor, M. A. Razzaque, and M. R. A. Khandaker, ”Lightweight cryptography algorithms for resource-constrained IoT devices: A review, comparison and research opportunities,” *IEEE Access*, vol. 9, pp. 28177–28193, 2021, doi: 10.1109/ACCESS.2021.3052867.

[3] A. Sevin and A. A. Mohammed, ”A survey on software implementation of lightweight block ciphers for IoT devices,” *Journal of Ambient Intelligence and Humanized Computing*, 2021, doi: 10.1007/s12652-021-03395-3.

[4] H. H. Al-Badrei and I. S. Alshawi, ”Improvement of RC4 security algorithm,” *Advances in Mechanics*, vol. 9, no. 3, pp. 1467–1476, 2021.

[5] C. Paar and J. Pelzl, ”*Understanding Cryptography*,” Springer-Verlag, 2010, doi: 10.1007/978-3-642-04101-3.

[6] J. Daemen and V. Rijmen, ”*The Design of Rijndael: AES – The Advanced Encryption Standard*,” Springer, 2002.

[7] D. Hong *et al.*, ”LEA: A 128-bit block cipher for fast encryption on common processors,” in *International Workshop on Information Security Applications*, 2013, pp. 3–27, doi: 10.1007/978-3-319-05149-9-1.

[8] K. S. Patil, I. Mandal, and C. Rangaswamy, ”Hybrid and Adaptive Cryptographic-based secure authentication approach in IoT based applications using hybrid encryption,” *Pervasive and Mobile Computing*, vol. 82, 2022, Art. no. 101552, doi: 10.1016/j.pmcj.2022.101552.

[9] Z. M. J. Kubba and H. K. Hoomod, ”*A hybrid modified lightweight algorithm combined of two cryptography algorithms PRESENT and Salsa20 using chaotic system*,” 2019 First International Conference of Computer and Applied Sciences (CAS), 2019, pp. 1–6, doi: 10.1109/CAS47993.2019.9075531.

[10] R. H. Altaie and H. K. Hoomod, ”An intrusion detection system using a hybrid lightweight deep learning algorithm,” *Engineering, Technology & Applied Science Research*, vol. 14, no. 5, pp. 16740–16743, 2024, doi: 10.48084/etasr.7657.

[11] P. Schwabe and K. Stoffelen, ”*All the AES you need on Cortex-M3 and M4*,” in International Conference on Selected Areas in Cryptography, 2016, pp. 1–15.

[12] D. Lee *et al.*, ”Efficient hardware implementation of the lightweight block encryption algorithm LEA,” *Sensors*, vol. 14, no. 1, pp. 975–994, 2014, doi: 10.3390/s140100975.

[13] Y. Nir and A. Langley, ”*ChaCha20 and Poly1305 for IETF Protocols*,” RFC 7539, 2015. [Online]. Available: https://tools.ietf.org/html/rfc8439.

[14] J. Zhang, W. Gao, J. Li, X. Tian, and H. Dang, ”*High-Speed and High-Security Hybrid AES-ECC Cryptosystem Based on FPGA*,” in 2019 IEEE International Conference on Signal, Information and Data Processing (ICSIDP), 2019, pp. 1–6, doi: 10.1109/ICSIDP47821.2019.9173457.

[15] H. Mostafaa, S. M. Eisaa, H. H. Issaa, and N. H. Shaker, ”*Lightweight Hybrid Encryption System with FPGA Design Proposal*,” IOP Conference Series: Materials Science and Engineering, vol. 1051, 2021, Art. no. 012345, doi: 10.1088/1757-899X/1051/1/012023.

[16] R. Verma and J. Dhiman, ”*Implementation of Improved Cryptography Algorithm*,” Int. J. Inf. Technol. Comput. Sci., vol. 14, no. 2, pp. 45–53, Mar.-Apr. 2022, doi: 10.5815/ijitcs.2022.02.04.

[17] G. A. Nikitha *et al.*, ”*Hybrid cryptographic algorithm to secure internet of things*,” in 2023 7th International Conference on Intelligent Computing and Control Systems (ICICCS), 2023, pp. 1556–1562, doi: 10.1109/ICICCS56967.2023.10142709.

[18] J. Daemen and V. Rijmen, *"Rijndael,"* in Encyclopedia of Cryptography, Security and Privacy. Springer, 2025, pp. 2110–2114. doi: 10.1007/978-3-662-04722-4.

[19] A. Tiwari *et al.*, "*Hybrid Cryptography Algorithms for Cloud Data Security,*" in 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), 2024, pp. 1–5, doi: 10.1109/ICCCNT61001.2024.10725233.

[20] A. Sevin, E. Savas, and A. Aysu, "Area-Efficient Hardware Implementations of Lightweight Cryptographic Algorithms," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems)*vol. 29, no. 5, pp. 1025–1038, 2021, doi: 10.1109/TVLSI.2021.3059812.

[21] E. Barker and Q. Dang, "Recommendation for Key Management, Part 1: General," *Nat. Inst. Stand. Technol.*, Gaithersburg, MD, USA, NIST SP 800-57, Pt. 1, Rev. 5, May 2020. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r5.pdf, doi: 10.6028/NIST.SP.800-57pt1r5.

[22] K. Agyei, K. Owa, T. Al-Hadhrami, "Analysis and Improvement of MixColumn Operations in the Advanced Encryption Standard Algorithm". In: *World Congress in Computer Science, Computer Engineering and Applied Computing.* Cham: Springer Nature Switzerland, p. 394-408, 2024.

[23] H. V. Gamido, A. M. Sison, and R. P. Medina, "Modified AES for text and image encryption," *Indonesian Journal of Electrica l Engineering and Computer Science (IJEECS)*, vol. 11, no. 3, pp. 942–948, 2018, doi: 10.11591/ijeecs.v11.i3.pp942-948.

[24] A. Rukhin *et al.*, A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications, *Nat. Inst. Stand. Technol.*, Gaithersburg, MD, USA, NIST SP 800-22, Rev. 1a, Apr. 2010. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-22r1a.pdf.

[25] Z. A. Al-Jazaeri, J. R. Naif, and A. M. Ghandour, "Data security model using (AES-LEA) algorithms for WoT environment," *Iraqi J. Comput. Inform.*, vol. 51, no. 1, pp. 94-107, 2025, doi: doi: 10.25195/ijci.v51i1.559.

BIOGRAPHIES OF AUTHORS

Hala Shaker Mehdy have a Bachelor's degree in Computer Science from Anbar University in Iraq, 2003-2004 and a Master's degree from the Southern Federal University in Russia. I am currently working as a lecturer at Al-Mustansiriya University in Baghdad. I am currently a Ph.D. student at the College of Computing and Informatics, Universiti Tenaga Nasional, Malaysia. My research interests include information technology, artificial intelligence, data security, and internet of things. She can be contact at email: hala.shaker@uomustansiriyah.edu.iq.

Mohd Ezanee Rusli He received the M.Sc. degree in computer systems design from the University of Manchester Institute of Science and Technology (UMIST), U.K., in 2001, and the Ph.D. degree in network engineering from Massey University, New Zealand, in 2012. He is currently an Associate Professor with the College of Computing and Informatics, Institute of Informatics and Computing in Energy, Universiti Tenaga Nasional (UNITEN), Malaysia, where he is also the University's Chief Operating Officer. His research interests include cybersecurity, cyber-physical systems, artificial intelligence, machine learning, and data mining. He can be contacted at email: ezanee@uniten.edu.my.

Haider. K. Hoomod I hold a Bachelor's degree in Electrical Engineering from the Department of Electrical Engineering at the University of Technology in Baghdad, Iraq. I also hold a Master's and Ph.D. degrees in Computer Networks from the Department of Communications Engineering at the University of Technology in Baghdad. I am currently a Professor of Computer Science at the College of Education, Al-Mustansiriya University. My research interests include: smart security, smart grid security, network security, Internet of Things (IoT) and World Wide Web (WoT) security, hashing and encryption techniques, data encryption, information security, chaotic systems, image and video processing, cloud computing security, wireless networks, wireless sensor networks, embedded systems, and operating systems. He can be contacted at email: drjnew@gmail.com.