International Journal of Inventive Engineering and Sciences (IJIES)
ISSN: 2319-9598 (Online), Volume-13 Issue-1, January 2026

An Extended AES Scheme for Increasing Randomness in Ciphertext

Ushawu Ibrahim, Edem Kwedzo Bankas, Callistus Ireneous Nakpih

Abstract: ***Technological advancements, such as high-speed internet, have transformed the world into a global village, raising concerns about privacy and secrecy amid cyberattacks and the disclosure of sensitive data. Cryptography and steganography are two well-known methods of secret communication. The former distorts the message, whilst the latter hides the very existence of the information within seemingly innocent carriers. Steganography faces challenges of steganalysis, whilst cryptography faces challenges of cryptanalysis. The extensive approval of Advanced Encryption Standard (AES) as an efficient symmetric cryptographic technique and other state- of-the-art data protection techniques has exposed them to increased attacks, prompting researchers to enhance AES's strength. To contribute to the line of research, a novel matrix-based diffusion layer for the AES (MDLAES) scheme is proposed. The proposed scheme combines matrix data manipulation with the AES algorithm, adding an extra layer of security. This extended scheme produces a data scrambling algorithm that reconstructs plain text and secret keys before performing AES encryption on the result. The approach, first and foremost, ensures that knowledge of the initial key is insufficient to break the system; it also introduces a higher degree of randomness than the traditional AES cryptosystem. The study examined the performance of encryption and decryption operations using key sizes from 128 to 256 bits. As key size increases, CPU time and memory usage increase. It is also observed that AES encryption with matrix operations requires more CPU time and memory than the traditional AES algorithm. The research improves the diffusion rate by 3.04 when a single simulation is matched with the orthodox AES algorithm, and by 1.62 on average when 10 simulations are run with different keys. It is worth noting that a high diffusion rate and a double key make it more difficult for a plain-text attack.***

Keywords: Cryptography, AES, Randomness, Diffusion, Plain Text Attack.

Nomenclature:
AES: Advanced Encryption Standard
CP: Constraint Programming
IWT: Integer Wavelet Transform
IT: Initial Text
KI: Key Inverse
DES: Data Encryption Standard

Manuscript received on 03 December 2025 | Revised Manuscript received on 19 December 2025 | Manuscript Accepted on 15 January 2026 | Manuscript published on 30 January 2026.

*Correspondence Author(s)

Ushawu Ibrahim*, University for Development Studies, Tamale, Ghana. Email ID: ushawu.ibrahim@uds.edu.gh, ORCID ID: 0009-0008-4539-1961

Edem Kwedzo Bankas, Associate Professor, Department of Business Computing, C. K. Tedam University of Technology and Applied Sciences, Navrongo, Ghana; School of Computing and Information Sciences. Email ID: ebankas@cktutas.edu.gh, ORCID ID: 0000-0001-7596-9682

Callistus Ireneous Nakpih, Department of Business Computing, C. K. Tedam University of Technology and Applied Sciences, Navrongo, Ghana; School of Computing and Information Sciences. Email ID: cnakpih@cktutas.edu.gh, ORCID ID: 0000-0002-7490-5166

I. INTRODUCTION

Technological progress, especially the development of fast internet for long-distance communication, has made it possible for information to roam the world. As a result, the globe is now truly a global village. But at the same time, people and businesses are worried about privacy and secrecy due to cyberattacks and the disclosure of sensitive data [1], which is where cryptography comes into play. Cryptography is the foundation of contemporary privacy technologies and has expanded the realm of data minimisation, a fundamental tenet of privacy engineering and also privacy by design [2]. Cryptography plays a key role in implementing data minimisation techniques, including minimal data exposure and minimal data collection, which reduces the need to trust end users [3]. Theoretically, cryptography enables the development of privacy-reliant systems that do not rely on the generosity or good behaviour of service providers or systems administrators, minimising the urge to entrust them with the fortification of users' privacy. This is achieved by strategy and implemented over code rather than through prescribed arrangements or confidentiality policies. Even though the Advanced Encryption Standard (AES) remains a widely used symmetric cryptography method, it still suffers from side-channel attacks, such as cache-timing, which can recover keys in just a matter of minutes [4]. Information security measures intended to be implemented with AES continue to face threats, including brute force and fault injection; these actions of attackers have led scientists to come up with mitigating factors such as internal structural modifications, including secret key generation, as well as randomisation of key-independent transformation to increase AES's effectiveness [5]. In cryptography, related-key differential attack refers to a situation where the cryptanalyst probes block cyphers using plaintext pair(s) to infer the secret private key that was used for encryption [6]. Additionally, the strength analysis of the ciphertext transmitted out by cypher experts disclosed that, in accordance with the contemporary development of cumulative computational supremacy, eight of the ten. rounds in AES can be effectively and quickly attacked by brute force, leaving only two rounds that could be readily cracked [7]. One well-known approach to breaking traditional simple substitution or transposition cyphers is the probable word method, also known as a cipherkey-plaintext pair, which exemplifies a known-plaintext attack; these attacks allow the intruder to compromise systems using a related key [8]. From the ongoing discussion, AES is a robust encryption standard that encrypts plaintext into ciphertext, making it harder to break or decrypt without

knowledge of the secret key. However, hackers continually test their skills on standard schemes, and AES is no exception. Plaintext is a method that can break a system if the hacker has access to several plaintexts and their corresponding ciphertexts; however, generating more random plaintext with a high avalanche effect reduces the probability of guessing the key through a plaintext attack. This has necessitated exploring and developing experimental techniques that could further strengthen the conventional AES algorithm. The paper extends the AES algorithm by applying byte-level manipulations to the secret key and the data before encryption. Making the final text more random, thereby increasing the avalanche effect, is the justification for the manipulation method used in this paper. This means that altering any piece of the key or character in the secret text will significantly change the ciphertext. The remaining portions of the manuscript are organised as follows: II includes an explanation of cryptography and Encryption, a review of related work detailing the existing literature, and key findings. The conceptual framework, suggested algorithms, and suggested methodology are all explained in the Proposed Method. Results and discussions include the experimental outcomes of the extended AES algorithm, performance analysis and comparison based on avalanche effects, and process execution time.

II. CRYPTOGRAPHY

The term Cryptography refers to a method that ensures message confidentiality. It is a Greek word with a translation of" secret writing." It ensures that the information delivered is safe enough that only the authorised recipient can access it, protecting the privacy of people and organisations through various complementary tools [9]. Cryptography has a long history and is still being researched as an ancient method. The field of cryptography dates back to ancient Egypt, from 2000 BC.

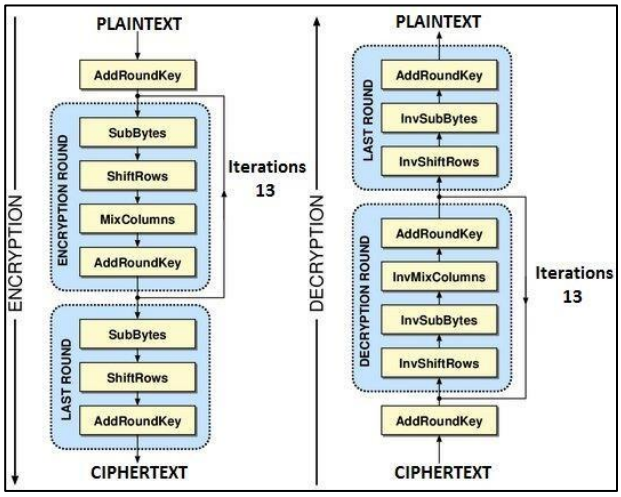
B.C. Since hieroglyphic writing was a coded language used for communication, it was another type of cryptography. Ancient Rome employed the Caesar cypher, a different kind of substitution cypher system in which every alphabetic letter in the communicated plaintext is shifted a definite quantity of places along the alphabet order [10].

A. The 3 forms of Cryptography are:

i. Symmetric key cryptography, which utilises a unique key for mutual decryption and encryption and is shared by both the receiver and sender. This is speedier than the asymmetric.

ii. Asymmetric cryptography, also known as public key cryptography, uses a private key, which is known only to the receiver, to decode messages and sign signatures. In contrast, a public key available to everyone is used to encrypt transmitted messages and to verify authenticated signatures. In this case, the data is encrypted with one key and decrypted with another.

iii. The Hash function uses a mathematical transformation that assists in irreversibly encrypting information. Sometimes termed as a no-key function or message digests, [11]



[Fig.1: AES Architecture (Source: [12]) Aes Algorithm]

The U.S. government adopted the popular encryption method known as AES (Advanced Encryption Standard) in 2002. It is a symmetric key encryption algorithm that encrypts and decrypts data using a block cypher. AES has key sizes of 128, 192, or 256 bits, making it safer than DES. The input plaintext is encrypted into ciphertext using AES with a round-based structure and a fixed block size of 128 bits. A round key derived from the initial encryption key is used to perform mathematical operations on the data throughout each round. AES is superior to other encryption algorithms in several ways, including flexibility, high efficiency, and excellent security. AES is now the accepted encryption algorithm for a wide range of applications, including government, financial, and military communications [13]. The structure of the AES encryption and decryption algorithm is outlined in Figure 1.

III. RELATED WORKS

This section produces a summary of existing works related to the research. It reviews works on AES cryptography that extend the traditional AES system, as well as works that increase randomness in ciphertexts. Some extensions of the AES system basically enhance processing time and security. In contrast, others seek to increase randomness in the ciphertext to reduce the likelihood of a plaintext attack.

A. Empirical Evidence

Offered an improved version of the AES algorithm by changing its SubBytes and ShiftRows configurations [5]. The AES algorithm was improved by modifying the SubBytes and ShiftRows transformations, yielding a round-key-dependent SubBytes transformation. The modified AES had an avalanche impact of 57.81%, slightly higher than that of traditional AES. However, the modified AES had slightly longer execution times, despite the enhanced encryption and decryption strength. Also, the simulation did not show how many secret keys were flipped; instead, the avalanche depended on the flipping of just one (1) byte location.

Detailed prototypes of Constraint Programming (CP) to address a cryptological issue, specifically the picked key differential attack, in contradiction to the standard block cypher AES [14]. The research showed that CP

solvers can resolve these difficulties more quickly than dedicated cryptanalyst tools. The study showed that the resolution considered best in the two most recent cryptanalysis studies is not optimal, as it yields a higher resolution. If the attacker offers pairs of plaintext bit blocks, x_1 and x_2, with known changes between them, the improved technique is also computationally costly and especially targets plaintext attacks.

Introduced a novel modification of the AES algorithm using the Butterfly Effect to enhance encryption and decryption processes [15]. The revised algorithm outperforms the original AES in diffusion, confusion, and integrity checks. The modified AES provides stronger ciphertext security and enhances the overall encryption and decryption process, yielding a significant increase in accuracy. The butterfly effect recurs across the 3 active stages of the traditional AES algorithm, increasing computational complexity.

To address low diffusion percentages in the first rounds, [16] altered the Advanced Encryption Standard (AES). Byte substitution, round constant addition, and primitive operations were added to the improved AES. Tests showed an average increase in diffusion and improved randomness of the ciphertext. The modified AES can successfully decrypt and recover the original plaintext, demonstrating improved diffusion and confusion properties.

For 1024 bytes of data, the combination of symmetric and asymmetric approaches takes 3.045 ms, rising to 3-4 ms for 2048 bytes of data, and so on, according to [17]. By altering the S-Box and Shift Row, the study proposes a novel approach to improve the AES algorithm's Mix Column transformation. The outcome demonstrates that optimisation decreased by 3 milliseconds and will continue to accelerate as the byte count rises. The approach uses more memory to hold two additional modified S-Box maps and an Array Shift Row map, and the optimisation's percentage average is 86.143%.

which suggests that a less capable machine might be able to breach the system after multiple trials.

Encrypted the cover image into 16 x 16 blocks in a separate study after encrypting the secret data using the AES encryption algorithm [18]. The Integer Wavelet Transform (IWT) is then applied to the cover image to use a neural network to locate the pixel for steganography. Lastly, the secret data bits are substituted for the LSB bits of the pixels in the array using the traditional LSB technique. As a result of IWT determining pixel locations, this approach offers an additional degree of security by making it impossible to retrieve hidden data.

The primary goal of [19] was to improve the security of the current AES algorithm by conducting an inclusive investigation into its security. The research effectively raised the Time Security and Strict Avalanche Criterion by altering the current AES algorithm through XORing an extra byte with the s-box value. To improve the security, they added a random extra key. The outcome of the security measurement can vary because this key is random. The avalanche effect's result is still minimal compared to the most recent works in the state-of-the-art comparison in Table 4.

The key security of the Playfair cypher, which used bit shifting, two's complement, the XOR operator, and a 16 × 16 matrix, was proposed in the article by [20]. They used the seven appropriate randomness tests in the NIST Test Suite. The chosen randomness tests that the suggested method passed several criteria, including the frequency (Monobit) test, the frequency test inside a block, the run test, the test for the lengthiest execution of ones in a block, the discrete Fourier transform, the approximation entropy test, and the cumulative sums test. The experiment's results demonstrate that the binary sequences generated are random. P-values ranged from 0.01 to 1.00 for the various key lengths (10, 20, 30, and 40).

Two of the main benefits of [21] work is increased security and user data privacy. This adds a double-round key component, which speeds up the encryption route by 1000 blocks per second when compared to the previous 128 AES standard technique. Nonetheless, a single round key with 800 blocks per second is typically used. Improved load balancing, reduced power consumption, and enhanced network resource management are all benefits of the proposed algorithm. The deployment of the standard AES with 128-, 64-, 32-, and 16-bit block sizes, exposing text bytes, is part of the proposed framework. The visualisation of simulation results illustrates the algorithm's usefulness in obtaining specific superiority properties. The proposed framework lessens energy utilisation by 14.43%, network usage by approximately 11.53%, and the delay by 15.67%, according to the results. Therefore, when establishing computational cloud services, the outlined architecture enhances security, minimises resource utilisation, and decreases delay.

B. Key Findings

From the empirical evidence, it can be deduced that some modifications of the standard AES algorithm increase computational complexity, as in the case of [15], while others alter sub-bytes and shift rows, specifically in the case of [17]. Also, symmetric and asymmetric cryptography techniques are combined by adding an S-box map, which is both computationally demanding and memory-intensive [19]. Also added a random extra key to increase diffusion and randomness, but the increment rate is not very good. Even though the work of [5] increases the diffusion rate by a large margin, the research has not provided enough simulations to confirm the consistency and robustness of the results. The work will produce a simple, less computationally complex extensible plugin that increases randomness in the final ciphertext. This will go a long way toward preventing plaintext attacks on the AES cryptosystem.

IV. PROPOSED METHOD

This section details all procedures, including algorithms and simulation tools, for achieving the stated objectives. The proposed scheme is well discussed in terms of algorithms, concepts, flowcharts, and both forward and reverse procedures, with mathematical illustrations. The discussions include pseudocode listings, algorithms, and flowcharts. The detailed implementation of the scheme using text is equally discussed. Evaluation of the proposed scheme is done. The plain text and cypher key are confused using matrix

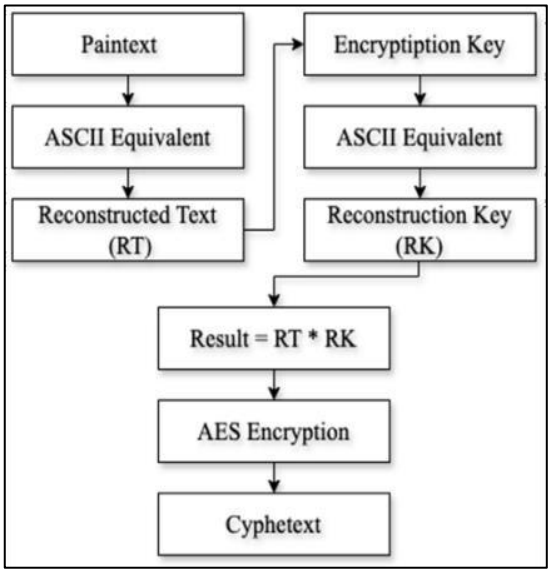
Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

multiplication; the resultant data is further encrypted using the conventional AES. The final result is then transferred to the recipient or embedded in the case of steganography.

Two levels of security are employed here: AES encryption and an extended matrix multiplication. The flowchart in Figure 2 outlines the general structure of the method used. The given plain text and secret key are both reconstructed and shaped into the required number of matrices. A result is calculated from the product of the reconstructed plain text and the secret key. The standard AES algorithm is then applied to the result to further randomise the ciphertext.

A. Conceptual Framework

The conceptual framework, as shown in Figure 2, illustrates how increasing randomness in the ciphertext amplifies the avalanche effect. The operational procedure of the framework is outlined below:



[Fig.2: Conceptual Framework]

B. Forward Conversion

The conceptual framework, as shown in Figure 2, depicts the forward procedure for adding randomness to the resultant ciphertext by increasing the avalanche effect, which is elaborated here. The operational procedure of the framework is outlined below:

i. The plain text, which refers to the secret message that is to be transmitted, is first converted into an ASCII equivalent

ii. The result from (1) is reconstructed into matrix form using the number of parts (N) as specified in Algorithm 1. Where the result from (1) above is an odd number, an arbitrary constant is added to ensure the matrix is well-formed.

iii. The secret key, which refers to the key used for both encryption and decryption, is also converted into an ASCII equivalent.

iv. A matrix is formed out of the result from (3) using the procedure as outlined in Algorithm 1.

v. Multiply the output from (2) and the output from (4) to get the Result.

vi. Pass the Result, which is now the new plain text as well as the original secret key, into the standard AES algorithm.

vii. Encrypt and produce the ciphertext for transmission to the recipient.

C. Reverse Conversion

The reverse process of the concept is outlined below.

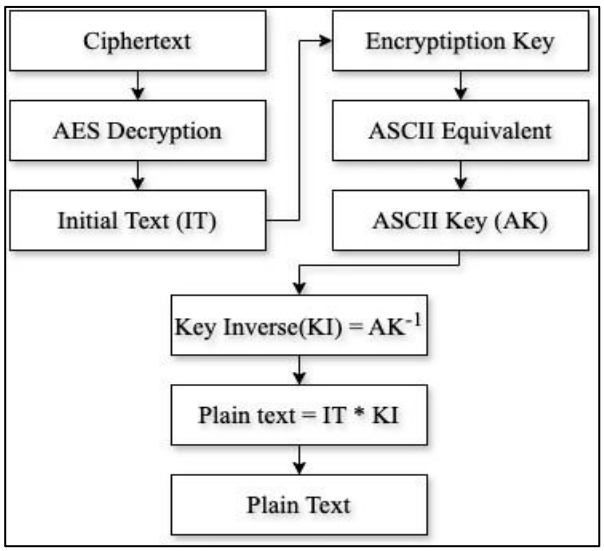
i. The ciphertext text, which contains the covert communication, is first decrypted through AES using the secret key to produce the Initial Text (IT).

ii. The ASCII equivalent of the secret key is computed.

iii. Compute the Key Inverse (KI) of the ASCII equivalent of the secret key as computed in (ii) above using equation 9

iv. Multiply the inverse computed in (3) above by the plain text extracted in (1) to produce the original plain text that has been communicated.

The logical flow of the reverse conversion process is outlined in Figure 3, a similar figure to the conceptual framework, but for the reverse process.



[Fig.3: Reverse Conversion Process]

D. Matrix Multiplication

Consider N to be the required number of data parts into which we wish to split the plain text and secret data. N is chosen such that it can be reconstructed in matrix form for multiplication.

i. Data Illustration

If N = 4, using arbitrary constants, then we end up with the following 2 x 2 matrix, such as Equations 1 and 2, whose resultant equation is Equation 3 [22]:

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \quad \ldots \quad (1)$$

$$B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \quad \ldots \quad (2)$$

$$C = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \quad \ldots \quad (3)$$

Where the variables are expanded into Equations 4, 5, 6 and 7 below [22]:

$$c_{11} = a_{11} \cdot b_{11} + a_{12} \cdot b_{21} \quad \ldots \quad (4)$$
$$c_{12} = a_{11} \cdot b_{12} + a_{12} \cdot b_{22} \quad \ldots \quad (5)$$
$$c_{21} = a_{21} \cdot b_{11} + a_{22} \cdot b_{21} \quad \ldots \quad (6)$$
$$c_{22} = a_{21} \cdot b_{21} + a_{22} \cdot b_{22} \quad \ldots \quad (7)$$

E. Matrix Inversion

Matrix inversion is required to recover the original values. In order to find matrix B where A is given, you will need to find the inverse of A in Equation 8 and multiply by resultant C to get back B, as shown in Equation 10 [22].

Matrix inversion is required to recover the original values

Let's find B given

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad \ldots \quad (8)$$

$$A^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix} \quad \ldots \quad (9)$$

$$B = A^{-1} \cdot C \quad \ldots \quad (10)$$

Where $ad - bc$ is the determinant of matrix A = $| A |$

Therefore

Algorithm 1: Algorithm For Splitting the Secret Key

Inputs: **k** = ACII equivalent of Secret data, **N**= Number of parts to divide secret key into

Output: $x_1, x_2, \ldots, x_n$

```
l ← length (k)
Pn ← floor (1/N)
R ← l mod N
for (i = 0, i++, l – 1) do
| xi ← str(k) [i * Pn:(i + 1) * Pn]
endfor
If R ≠0
    xn ← str (k)[* Pn:]
endif
```

Where **R** = remainder, $\mathbf{P_n}$ = length of data in a part, $\boldsymbol{x_i}$ = data in parts of size $\mathbf{P_n}$

F. Splitting The Key / Plain Text

This section describes the process of splitting the data and the secret key into the number of columns suitable for matrix multiplication. Data must be converted to an ASCII equivalent for byte splitting, as demonstrated in Algorithm 1; this procedure could also be used for bit splitting. Splitting, one only has to convert the secret message further into its binary equivalent.

i. Data Illustration:

Consider the ASCII equivalent of a data string as: 12345678901234567892

Parts to divide data string to (N) = 4

Length of string (l) = len|12345678901234567892| = 20

Calculating part length P_n = 5

R = 20 mod 4 = 0 Initialise an array = [],

Start index of the current = i×5,

End index of current part, start index of the current +5.

Concatenate the start and end strings to form the complete string. Computing the parts:

For i=0, start=0, end=5, = "12345" that is (index 0 – 4 inclusive).

For i=1, start=5, end=10, = "67890" that is (index 5 – 9 inclusive).

For i=2, start=10, end=15, = "12345" that is (index 10 – 14 inclusive).

For i=3, start=15, end=20, = "67892" that is (index 15 – 19

inclusive). R = 0 = "67892"

Considering the data string: "12345678901234567892", The Split data = "12345", "67890", "12345", "67892".

If, in any case, there is a remainder when computing the part number, the remainder is appended to the last part.

G. Reconstructing the Key

The split key used for multiplication on the plain text can be reconstructed using Algorithm 2. The algorithm takes the data parts and loops through them to reconstruct them into 1 part representing the ASCII equivalent of the original secret data or key. It converts plain text and keys between ASCII and binary to enable seamless algebraic operations and linear transformations in cryptography.

Algorithm 2: Algorithm for

Inputs: x_1, x_2, x_n

Output: K

```
1 for (i =0, i++, x1 − 1) do
2 | k ← x1
3  endfor
4 return k
```

Where k = ACII equivalent of string x_i = data in parts

H. Evaluation Using Avalanche Effect

Horst Feistel originally introduced the phrase" avalanche effect" in his 1973 work; subsequently, the idea was recognised as Shannon's confusion property. The avalanche effect measures the level of instability (nonlinearity) in cryptographic algorithms, including hash functions, especially block cyphers such as the Advanced Encryption Standard (AES) and the Data Encryption Standard (DES) [15].

The Avalanche Effect can be expressed mathematically as captured in [5] and summarized below:

let AE = Avalanche Effect

h = No. of bit difference in 2 cypher texts

z = Total number of bits in ciphertext

$$AE = \frac{h}{z} * 100 \quad \ldots \quad (11)$$

A cryptographic algorithm has inadequate randomisation if it does not show an appreciable amount of avalanche effect (at least 50 per cent). As a result, given only the output, cryptanalysts can guess the input. This could be sufficient to break the algorithm entirely, or worse, only partially.

Aside from the avalanche effect, encryption and decryption times were also recorded in [17]. The formula for finding the avalanche is thus shown in Equation 11.

V. RESULTS AND DISCUSSION

This sector presents the outcomes of the proposed scheme. The scheme was tested with a data set and compared to the orthodox AES algorithm. The proposed scheme was tested using Python version 3.9.13 on a MacBook Air (2020) with an Intel Core i3 processor at 1.1 GHz, 8 GB of memory (LPDDR4X), running macOS Ventura 13.4.1 ©. The matrix

size could be any, but a 2 x 2 is chosen for the initial demonstration. The results obtained when keys of different lengths, 128, 192, and 256 bits, are used with the standard plain text "Proposed Hybrid Chaos scheme using Cryptography and Steganography" are displayed in Tables 1, 2, and 3, along with their respective outputs. The table results include the average CPU time, memory usage, and processing time. These values were recorded after running the simulation 16 times and averaging 10 recordings.

Table I: Results for Conversion Using 128 Bits of Key

		Encryption			Decryption		
No.	Algorithm	CPU	Memory	Average Time	CPU	Memory	Average Time
1	Standard AES	21.22	72.4	1.031138	15.32	72.72	0.836108
2	Proposed MDLAES	31.98	73.76	1.323388	34.4	73.7	1.084088

A. Performance Analysis

It is important to note that as the key magnitude increases from 128 to 256 bits, CPU time and memory usage also increase for encryption and decryption operations. 256-bit AES encryption has slightly higher CPU time and memory usage. Still, it may vary by implementation and hardware configuration, since a machine with high resources will yield higher efficiency than a system with lower computational capabilities. In this work, AES encryption with matrix operations requires more CPU time and memory.

Table II: Results for Conversion using 192 Bits of Key

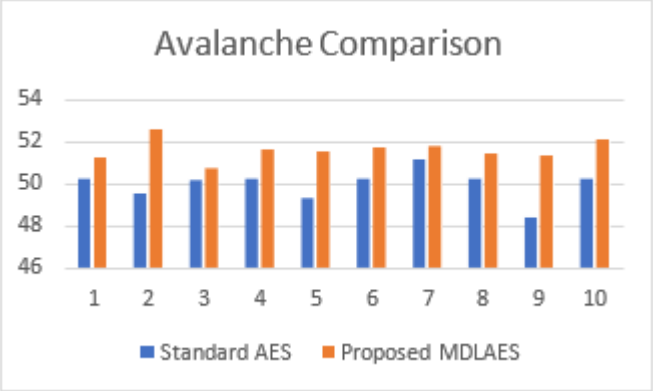
		Encryption			Decryption		
No.	Algorithm	CPU	Memory	Average Time	CPU	Memory	Average Time
1	Standard AES	48.08	73.74	1.005144	43.32	74.1	0.910842
2	Proposed MDLAES	38.98	74.46	1.553792	39.64	74.46	1.20007

Table III: Results for Conversion using 256 Bits of Key

		Encryption			Decryption		
No.	Algorithm	CPU	Memory	Average Time	CPU	Memory	Average Time
1	Standard AES	38.6	74.94	1.00675	41.74	75.48	0.913276
2	Proposed MDLAES	9.34	72.42	1.491654	16.52	72.36	1.36563

B. The Avalanche Effect

As shown in Figure 4, there has been a consistent increase in avalanche effect over 10 iterations using a 128-bit secret key. The results are obtained by maintaining the initial key (Key1) and changing one byte of the second key (Key2). Figure 4 presents a graphical view of the results and indicates that MBDLAES has a higher mean avalanche effect (51.58%) than traditional AES (49.96%), with a lower standard deviation (0.41%). Both algorithms show consistent results, with the matrix algorithm showing a slightly higher and more consistent avalanche effect. This suggests that the MDLAES has improved cryptographic properties as compared to the standard AES algorithm.



[Fig.4: Avalanche Test Result Between AES and the Proposed MDLAES]

C. State-of-the-Art Comparison

The performance of the scheme for similar works is shown in Table 4. Also, considering the existing literature, [5] shows that the results were obtained by flipping a single character, replacing an 'r' with an 's', which could explain the high value attributed to it. An algorithm that stands the test of time should be tested with varying data to confirm its robustness. Although the difference recorded in Figure 4 was the highest for the proposed MDLAES algorithm, the researchers repeated the test with different bytes at different locations. They found the average across 10 data sets of size 128 bits to be 16.2, as shown in Figure 4. From the results, it is apparent that flipping or substituting different bytes of the secret key will produce different results even when the same plaintext is used.

Table IV: State-of-the-Art Comparison with Respect to Avalanche Effect

No.	Author	Conventional AES (%)	Modified Algorithm (%)	Difference
1	[5]	49.973	56.3625	6.3895
2	[19]	50.78	52.34	1.56
3	Proposed MDLAES	49.51	52.55	3.04

VI. CONCLUSION AND RECOMMENDATION

The paper proposed an extended cryptographic Algorithm. The scheme extends AES encryption by reconstructing the secret key and the plaintext into two independent matrices. These two matrices are multiplied together to produce a result; an AES encryption is then applied to the result to create the final ciphertext. The ciphertext payload is sent to the recipient in an appropriate medium for decryption using the proposed decryption algorithm. The scheme's increased randomness in the ciphertext prevents a plaintext attack. It is recommended that subsequent results focus on enhancing the hybrid algorithm by using a Residue Number System to compress the resulting matrix data, thereby improving computational speed and minimising storage space.

DECLARATION STATEMENT

IJEAT and/or the editor or editors are not responsible for any of the clauses, words, views, opinions, or data used in experiments that are contained in any publication;

they are typically and exclusively owned by the individual author and supplied and recognised. IJEAT and/or the editor(s) hereby disclaim all duty for any loss or damage to public or private property resulting from any concepts, notions, techniques, products, or instructions used or discussed in any of the information supplied.

As the article's author, I must verify the accuracy of the following information after aggregating input from all authors.

- **Conflicts of Interest/ Competing Interests:** Based on my understanding, this article has no conflicts of interest.
- **Funding Support:** This article has not been funded by any organizations or agencies. This independence ensures that the research is conducted objectively and without external influence.
- **Ethical Approval and Consent to Participate:** The content of this article does not necessitate ethical approval or consent to participate with supporting documentation.
- **Data Access Statement and Material Availability:** The code used to produce the results is available at https://github.com/rajah20/RNSB_AES.
- **Author's Contributions:** Conceptualization; U.I., C. I. N and E. K. B., methodology; U.I. and C. I. N., software, U.I.; validation, U.I., E. K. B. and C. I. N.; formal analysis, U.I and and C. I. N..; investigation, U.I. and and E. K. B; resources, U.I. and E. K. B; data curing, U.I.; writing: Initial draft authoring, U.I.; writing: reviews and editing and formatting, U.I., E. K. B. and C. I. N.; visualization, U.I. and C. I. N.; supervision, E. K. B. and C. I. N.; project administration, C. I. N., The published version of the article has been read and approved by all writers; other areas are collaborative efforts at every step.

REFERENCES

1. M. A. Usman and M. R. Usman, "Using image steganography for providing enhanced medical data security," in *2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, IEEE, Jan. 2018, pp. 1–4. DOI: http://doi.org/10.1109/CCNC.2018.8319263.
2. E. Balsa, H. Nissenbaum, and S. Park, "Cryptography, Trust and Privacy: It's Complicated," in *Proceedings of the 2022* Symposium *on Computer Science and Law*, New York, NY, USA: ACM, Nov. 2022, pp. 167–179. DOI: https://doi.org/10.1145/3511265.3550443
3. S. Rathore, P. K. Sharma, V. Loia, Y.-S. Jeong, and J. H. Park, "Social network security: Issues, challenges, threats, and solutions," *Inf Sci (N Y)*, vol. 421, pp. 43–69, Dec. 2017, DOI: https://doi.org/10.1016/j.ins.2017.08.063
4. N. Mouha, "Review of the Advanced Encryption Standard," Jul. 2021. DOI: https://doi.org/10.6028/NIST.IR.8319
5. O. C. Abikoye, A. D. Haruna, A. Abubakar, N. O. Akande, and E. O. Asani, "Modified advanced encryption standard algorithm for information security," *Symmetry (Basel)*, vol. 11, no. 12, Dec. 2019, DOI: https://doi.org/10.3390/SYM11121484
6. D. Gérault, P. Lafourcade, M. Minier, and C. Solnon, "Revisiting. This write-up hasn't received any funding by way of support. To *the* best of our knowledge, there are no conflicts of interest. No, participation in the article with evidence does not require ethical approval or consent. AES related-key differential attacks with constraint programming," *Inf Process Lett*, vol. 139, pp. 24–29, Nov. 2018, DOI: https://doi.org/10.1016/j.ipl.2018.07.001
7. H. Zodpe and A. Sapkal, "An efficient AES implementation using FPGA with enhanced security features," *Journal of King Saud University - Engineering Sciences*, vol. 32, no. 2, pp. 115– 122, Feb. 2020, DOI: https://doi.org/10.1016/j.jksues.2018.07.002
8. N. Kolokotronis and S. *Cyber-Security Threats, and Dynamic Mitigation*. Press, 2021. Accessed: 06, 2025. [Online]. Available: https://www.routledge.com/Cyber-Security-Threats-Actors-and-Dynamic-Mitigation/Kolokotronis-Shiaeles/p/book/9780367745875
9. D. Mercadier et P. É. Dagand, "USUBA: High-throughput and constant-time cyphers, by construction," in *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI)*, Association for Computing Machinery, Jun. 2019, pp. 157–173. DOI: https://doi.org/10.1145/3314221.3314636
10. Abdalbasit Mohammed Qadir and Nurhayat Varol, "A Review Paper on Cryptography," *IEEE Xplore*, 2019, DOI: https://doi.org/10.1109/ISDFS.2019.8757514
11. E. A. Abanga, "Symmetric, Asymmetric and Hash Functions," *Advances in Multidisciplinary and Scientific Research Journal Publication*, vol. 10, no. 4, pp. 55–60, Nov. 2022, DOI: https://doi.org/10.22624/AIMS/DIGITAL/V10N4P7
12. A. Vishwanath, R. Peruri, and J. (Selena) He, "Security in Fog Computing through Encryption," *International Journal of Information Technology and Computer Science*, vol. 8, no. 5, pp. 28–36, May 2016, DOI: https://doi.org/10.5815/ijitcs.2016.05.03
13. A. B. L Nikhitha V S Arjun Naveen Chandra Gowda, "Survey of applications, advantages, and comparisons of AES *encryption* algorithm with other standards," *International Journal of Computational Learning and Intelligence*, vol. 2, no. 2, 2023, DOI: https://doi.org/10.5281/ZENODO.7921019
14. D. Gerault, M. Minier, and C. Solnon, "Using Constraint Programming to solve a Cryptanalytic Problem Using *Constraint* Programming to solve a Cryptanalytic Problem *," 2017. [Online]. Available: https://hal.science/hal-01528272
15. H. Talirongan, A. M. Sison, and R. P. Medina, "Modified advanced encryption standard using butterfly effect," in *2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management, HNICEM 2018*, Institute of Electrical and Electronics Engineers Inc., Jul. 2018. DOI: https://doi.org/10.1109/HNICEM.2018.8666368
16. E. M. De Los Reyes, A. M. Sison, and R. P. Medina, "Modified *AES* Cypher Round and Key Schedule," Indonesian Journal of Electrical Engineering and Informatics, vol. 7, no. 1, pp. 28–35, Dec. 2019, DOI: https://doi.org/10.1109/iciibms.2018.8549995
17. R. Riyaldhi, Rojali, and A. Kurniawan, "Improvement of Advanced Encryption Standard Algorithm with Shift Row and S.Box Modification Mapping in Mix Column," in *Procedia Computer Science*, Elsevier B.V., 2017, pp. 401–407. DOI: https://doi.org/10.1016/j.procs.2017.10.079
18. K. S. Seethalakshmi, Usha B A, and Sangeetha K N, "Security enhancement in image steganography using neural networks and visual cryptography," in *2016 International* Conference *on Computation System and Information Technology for Sustainable Solutions (CSITSS)*, IEEE, Oct. 2016, pp. 396–403. DOI: https://doi.org/10.1109/CSITSS.2016.7779393
19. A. Al-Mamun, S. S. M. Rahman, T. A. Shaon, and M. A. Hossain, "Security analysis of AES and enhancing its security by modifying s-box with an additional byte," *International Journal of* Computer *Networks and Communications*, vol. 9, no. 2, pp. 69–88, Mar. 2017, DOI: https://doi.org/10.5121/ijcnc.2017.9206
20. R. M. Marzan, A. M. Sison, and R. P. Medina, "Randomness analysis on enhanced key security of Playfair cypher algorithm," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 8, no. 4, pp. 1248–1253, Jul. 2019, DOI: https://doi.org/10.30534/ijatcse/2019/34842019
21. I. A. Awan, M. Shiraz, M. U. Hashmi, Q. Shaheen, R. Akhtar, and A. Ditta, "Secure Framework Enhancing AES Algorithm in Cloud Computing," *Security and Communication Networks*, vol. 2020, 2020, DOI: https://doi.org/10.1155/2020/8863345
22. M. Dan and R. Joseph, *Interactive Linear Algebra*. Georgia Institute of Technology, 2019. Accessed: Dec. 06, 2025. [Online]. Available: https://textbooks.math.gatech.edu/ila/ila.pdf

AUTHOR'S PROFILE

Ushawu Ibrahim is an Assistant Registrar at the University for Development Studies and a former Assistant lecturer at the Bolgatanga Technical University. He Holds a BSc. Computer Science from the University for Development Studies, an MSc in Information Technology from the Kwame Nkrumah University for Science and Technology, Ghana, and currently pursuing a PhD at the C. K. Tedam University for Technology and Applied Sciences. His master's research delved into a comparative

analysis of PHP frameworks, which was duly published. He has also taught and managed IT infrastructure in the Nurses Training College under the Ministry of Health, Ghana. He also serves as a consultant for many NGO's and government institutions in Ghana and abroad. He is also an expert in both mobile and web applications. His research interests include software engineering and cybersecurity.

Edem Kwedzo Bankas, PhD, is an Associate Professor of Computational Mathematics at the Department of Business Computing, C.K. Tedam University of Technology and Applied Sciences (CKT-UTAS), School of Computing and Information Sciences, Navrongo, Ghana. He holds a Doctor of Philosophy (PhD) in Computational Mathematics from the University for Development Studies (UDS), Tamale, Ghana. Prof. Bankas's academic and research career spans the fields of Computer Arithmetic, Digital Logic Design, and Very Large- Scale Integration (VLSI) architectures, with a particular emphasis on redundant and other unconventional number representation systems. His scholarly work is firmly rooted in Computer Arithmetic and Digital Logic Design, with extensive research contributions in Cryptography and High-Performance and Parallel Computing. In these areas, his focus includes evaluating arithmetic algorithms, analysing computational complexity, and designing fault-tolerant computing systems capable of supporting efficient and reliable digital architectures. In addition to his core specialisation, Prof. Bankas also engages in research and project initiatives in the broader domain of Information Technology, particularly the application of computer technologies in education, capacity building, and community development. His work is driven by a commitment to advancing computational methods, strengthening research-based teaching, and leveraging technology to address societal and developmental challenges.

Callistus Irenaeus Nakpih, PhD, is a faculty member at C. K. Tedam University for Technology and Applied Sciences. He is also a former lecturer at St. John Bosco's College of Education. He holds a PhD in Computer Science from the University of Cape Coast, Ghana. His research interests include Artificial Intelligence and Theoretical Computing. A modified Vector Space Model for semantic information retrieval and a Novel Proximity-Based Sorting Algorithm for Real-Time Numerical Data Streams and Big Data Applications are two of his most recent publications.