symmetry

Article

LAO-3D: A Symmetric Lightweight Block Cipher Based on 3D Permutation for Mobile Encryption Application

Abdul Alif Zakaria [1,2,*], Azni Haslizan Ab Halim [1,3,*], Farida Ridzuan [1,3], Nur Hafiza Zakaria [1,3] and Maslina Daud [2]

[1] Faculty of Science and Technology, Universiti Sains Islam Malaysia, Nilai 71800, Malaysia
[2] CyberSecurity Malaysia, Menara Cyber Axis, Cyberjaya 63000, Malaysia
[3] CyberSecurity and System Research Unit, Faculty of Science and Technology, Universiti Sains Islam Malaysia, Nilai 71800, Malaysia
* Correspondence: alif@cybersecurity.my (A.A.Z.); ahazni@usim.edu.my (A.H.A.H.)

Abstract: Data transmissions between smartphone users require security solutions to protect communications. Hence, encryption is an important tool that must be associated with smartphones to keep the user's data safe. One proven solution to enhance the security of encryption algorithms is by using 3D designs on symmetric block ciphers. Although a 3D cipher design could improve the algorithms, the existing methods enlarge the block sizes that will also expand the key sizes and encryption rounds, thus decreasing their efficiency. Therefore, we propose the LAO-3D block cipher using a 3D permutation that offers security by providing confusion and diffusion characteristics. Five security analyses were conducted to assess the strengths of LAO-3D. The findings suggest that LAO-3D achieves better results compared to other existing lightweight block ciphers, with 98.2% non-linearity, 50% bit error rates for both plaintext and key modifications, surpasses 100% of the randomness test, and is immune to differential and linear cryptanalysis attacks. Moreover, the block cipher obtains competitive performance results in software applications. From the security analyses and performance tests, it is proven that LAO-3D can provide sufficient security at low costs in mobile encryption applications.

Keywords: 3D permutation; block cipher; cryptanalysis; Internet of Things; lightweight cryptography; mobile application; security analysis

check for updates

Citation: Zakaria, A.A.; Ab Halim, A.H.; Ridzuan, F.; Zakaria, N.H.; Daud, M. LAO-3D: A Symmetric Lightweight Block Cipher Based on 3D Permutation for Mobile Encryption Application. *Symmetry* **2022**, *14*, 2042. https://doi.org/10.3390/sym14102042

Academic Editors: Debiao He and Kuo-Hui Yeh

Received: 29 June 2022
Accepted: 26 September 2022
Published: 30 September 2022

1. Introduction

Cyber security awareness has positively impacted the development of security products on the market. Most people have smartphones to communicate with, browse the Internet, read e-mails, and perform their jobs. According to predictions, the number of mobile users worldwide will likely increase to 7.26 billion by the end of 2022 [1]. Regarding the potential for cyberattacks, almost everyone in the world needs security protection [2]. Information security is crucial due to the interception and modification of data, which can lead to loss of availability, integrity, confidentiality, as well as other losses, such as loss of life, money, and assets [3]. Therefore, encryption is an important tool in security products that must be associated with the smartphone to keep the user's data safe.

Most mobile encryption applications available on the market rely on conventional standard encryption algorithms, such as the AES block cipher, which has been widely implemented in various industries for more than a decade due to its security strengths [4]. Since then, numerous block cipher algorithms have been developed, using AES as the design foundation and the security benchmark. Performance evaluations of encryption algorithms in mobile devices have been conducted by comparing cryptographic algorithms, such as AES, DES, TEA, RSA, and REA [5]. The experimentation results revealed that AES recorded the fastest execution speed for the encryption process. However, a conventional algorithm, such as AES, requires huge memory and high power consumption, which is not practical to be implemented in the software encryption application [6]. Therefore, there is an important need for a lightweight algorithm design that uses simple cryptographic operations to provide sufficient security to the devices while emphasizing efficiency.

Over the years, a range of lightweight algorithms have been proposed in the literature that include BRIGHT [7], CRAFT [8], ACT [9], LRBC [10], LWE [11], PriPresent [12], T-TWINE [13], LBC-IoT [14], and improved SM4 [15]. Although lightweight algorithm research has been intensively explored using multiple techniques that aim for better efficiency, their security features cannot be ignored [16]. An algorithm should offer confusion and diffusion characteristics to provide sufficient security [17]. Confusion obscures the relationship between the plaintext and ciphertext with the substitution method, while diffusion spreads the plaintext statistics through the ciphertext using the permutation method.

One solution to enhance the confusion and diffusion characteristics of a cryptographic algorithm is by implementing the three-dimensional (3D) design on the block cipher. The 3D design invention was introduced by Nakahara [18], who adopted the model on the AES algorithm. It has been experimentally demonstrated that the deployment of the 3D cipher design can strengthen the security of the AES with provided confusion and diffusion to the algorithm. An alternative 3D design approach has similarly been suggested by Suri and Deora [19] using a 3D rotation block cipher that produced random outputs. Later on, Ariffin et al. [20] constructed a 3D−AES using an immune-inspired approach that passed all statistical tests. In another work, Mala [21] improved the encryption acceleration of the 3D block cipher using a unified byte permutation. Next, Wang and Jin [22] developed a non-alternate 3D model that was immune to differential and linear attacks. In 2019, Mushtaq et al. [23] employed 3D hybrid cubes that did not correlate the input and output. Recently, Zakaria et al. [24] increased the security strength of the RECTANGLE cipher with a 3D bit rotation function.

While the security strength of the block cipher could be improved by adopting the 3D design, there is an issue that needs to be resolved for the method to be applicable. The 3D design forces the algorithm to increase the block size and enlarge the key bits, thus decreasing its efficiency. An algorithm with large blocks and key sizes is not the best option for mobile encryption applications. Due to the above justifications, we propose a new algorithm called LAO-3D, which was developed using an enhanced 3D cipher design to balance its efficiency and security strength. Three contributions can be highlighted in this research. Firstly, the proposed algorithm does not require increasing the block and key sizes of the cipher, which solved the scalability issue highlighted in the previous 3D algorithms. Secondly, the new algorithm provides a better bit permutation for the algorithm. Thirdly, the analyses conducted on the algorithm prove its security strength and performance, which meets the criteria of a lightweight algorithm that require sufficient security strength and is efficient for mobile encryption application.

The research presented in this paper is divided into five major sections. Section 1 introduces the scope of this research. Section 2 specifies the 3D cipher design architecture. Section 3 details the proposed LAO-3D block cipher. Section 4 performs the security analysis, performance test, and implementation of the proposed algorithm in a mobile encryption application. Section 5 concludes the overall research work.

2. 3D Cipher Design Architecture

This section highlights the foundation of the 3D cipher that has been adopted since 2008. The 3D design has the potential to be implemented in various block ciphers. It has been shown that the 3D cipher can boost the randomness, confusion, and diffusion characteristics, enhance the non-correlation between the input and output data, and increase the security strength against cryptanalysis attacks of block ciphers [18–24].

2.1. 3-Dimensional Cipher

A cube is a three-dimensional (3D) plaintext bit array on a block cipher. Figure 1 displays a cube that contains three independent vectors involving n-bits length (x-axis),

n-bits width (y-axis), and n-bits depth (z-axis), in which the block length is divisible by n^2. The cube is made of n-Slices, which represent a portion of a plaintext array containing n^2 bits of data.

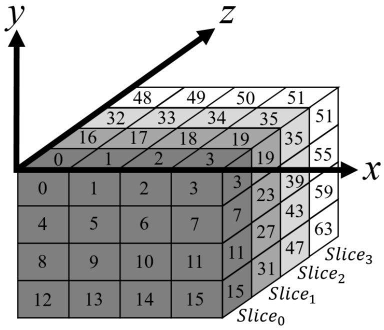


Figure 1. Cube.

A 3D cipher is composed of four plaintext Slices that are constructed by arranging input bits into quadruple 4 by 4 matrices. Let $W = \{(w_{63}, \ldots, w_0) : w_{j-1} \in W \text{ for } j = (64, \ldots, 1)\}$ demonstrate the plaintext. The initial 16 bits $(w_{15}, \ldots, w_0)$ are located in $Slice_0$ and the following 16 bits $(w_{31}, \ldots, w_{16})$ are assigned as $Slice_1$. The subsequent 16 cipher State bits are ordered as $Slice_2$ $(w_{47}, \ldots, w_{32})$ and $Slice_3$ $(w_{63}, \ldots, w_{48})$ as presented in Figure 2.

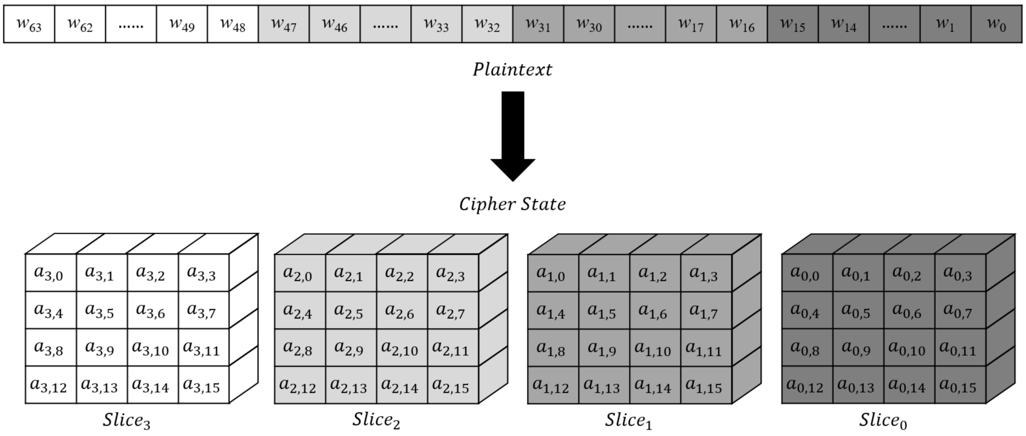
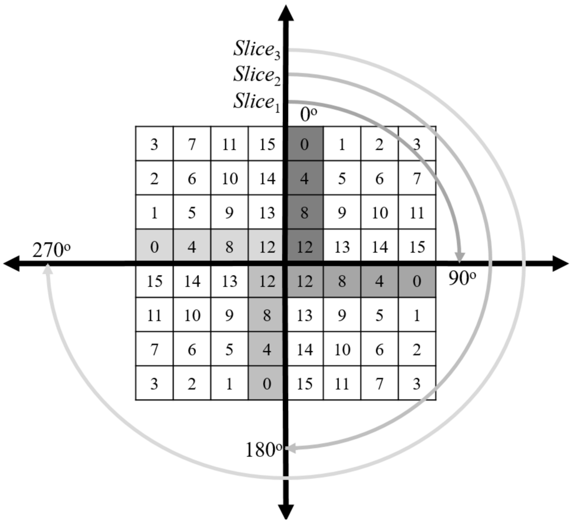

w_{63} w_{62} ...... w_{49} w_{48} w_{47} w_{46} ...... w_{33} w_{32} w_{31} w_{30} ...... w_{17} w_{16} w_{15} w_{14} ...... w_1 w_0
$Plaintext$
$Cipher\ State$
$a_{3,0}$ $a_{3,1}$ $a_{3,2}$ $a_{3,3}$
$a_{3,4}$ $a_{3,5}$ $a_{3,6}$ $a_{3,7}$
$a_{3,8}$ $a_{3,9}$ $a_{3,10}$ $a_{3,11}$
$a_{3,12}$ $a_{3,13}$ $a_{3,14}$ $a_{3,15}$
$Slice_3$
$a_{2,0}$ $a_{2,1}$ $a_{2,2}$ $a_{2,3}$
$a_{2,4}$ $a_{2,5}$ $a_{2,6}$ $a_{2,7}$
$a_{2,8}$ $a_{2,9}$ $a_{2,10}$ $a_{2,11}$
$a_{2,12}$ $a_{2,13}$ $a_{2,14}$ $a_{2,15}$
$Slice_2$
$a_{1,0}$ $a_{1,1}$ $a_{1,2}$ $a_{1,3}$
$a_{1,4}$ $a_{1,5}$ $a_{1,6}$ $a_{1,7}$
$a_{1,8}$ $a_{1,9}$ $a_{1,10}$ $a_{1,11}$
$a_{1,12}$ $a_{1,13}$ $a_{1,14}$ $a_{1,15}$
$Slice_1$
$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
$a_{0,4}$ $a_{0,5}$ $a_{0,6}$ $a_{0,7}$
$a_{0,8}$ $a_{0,9}$ $a_{0,10}$ $a_{0,11}$
$a_{0,12}$ $a_{0,13}$ $a_{0,14}$ $a_{0,15}$
$Slice_0$

Figure 2. 3D Cipher State.

2.2. 3DBitRotation Function

The 3DBitRotation function is proposed to strengthen the security of the block cipher by improving its confusion and diffusion characteristics. This function performs bit permutation on the quadruple 4 by 4 cipher State. The block cipher requires four Slices of 16 plaintext bits. Figure 3 displays the rotation formation of the 3DBitRotation function that follows a particular clockwise direction. $Slice_0$ remains unchanged; meanwhile, $Slice_1$, $Slice_2$, and $Slice_3$ undergo 90°, 180°, and 270° rotations, correspondingly.



3	7	11	15	0	1	2	3
2	6	10	14	4	5	6	7
1	5	9	13	8	9	10	11
0	4	8	12	12	13	14	15
15	14	13	12	12	8	4	0
11	10	9	8	13	9	5	1
7	6	5	4	14	10	6	2
3	2	1	0	15	11	7	3

Figure 3. The 3DBitRotation Formation.

Compared with the standard rotation method, there is a notable improvement in the 3DBitRotation function because it permutes the cipher bits in various orientations rather than in a one-way direction. The function offers better confusion and diffusion properties to the block cipher. Therefore, the 3DBitRotation function is used as the basic construction of the proposed block cipher to provide a better solution than the previous works.

3. Proposed LAO-3D Block Cipher

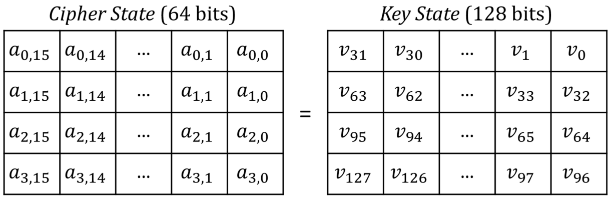
In this section, the primary contribution of our work is highlighted. We propose a lightweight algorithm designed using 3D cryptography, called the 3D light algorithm operation or LAO-3D. The core objective of LAO-3D development is to present a lightweight algorithm with sufficient security strength that can be used in mobile encryption applications. The step-by-step structure of the proposed algorithm will be discussed in the following sections.

3.1. Algorithm Specification

LAO-3D is a 20-round symmetric lightweight block cipher with a 64-bit block and a 128-bit key length. The block cipher realizes the substitution–permutation network (SPN) structure. SPN is chosen over the Feistel network because its execution time is shorter, the energy consumption is lower, and it has fewer round functions.

Each round of LAO-3D contains three main characteristics. First, the linear layer ensures multiple diffusion rounds. Second, the non-linear layer has parallel applications of S-boxes with sufficient confusion characteristics. Third, the key addition layer applies the XOR operation on the RoundKey and the cipher State.

Different transformations on the cipher State are represented as four rows of rectangular bit arrays. Meanwhile, the total number of cipher State columns is computed by dividing the block size (64 bits) by the number of rows. Identically, the secret key is depicted as four rows of rectangular arrays. The total number of secret key columns is equal to the key size (128 bits) divided by four rows. These representations of the cipher State and the key are shown in Figure 4.



Cipher State (64 bits)

$a_{0,15}$	$a_{0,14}$	...	$a_{0,1}$	$a_{0,0}$
$a_{1,15}$	$a_{1,14}$	...	$a_{1,1}$	$a_{1,0}$
$a_{2,15}$	$a_{2,14}$	...	$a_{2,1}$	$a_{2,0}$
$a_{3,15}$	$a_{3,14}$	...	$a_{3,1}$	$a_{3,0}$

=

Key State (128 bits)

v_{31}	v_{30}	...	v_1	v_0
v_{63}	v_{62}	...	v_{33}	v_{32}
v_{95}	v_{94}	...	v_{65}	v_{64}
v_{127}	v_{126}	...	v_{97}	v_{96}

Figure 4. Cipher State and Key State.

3.2. Encryption Algorithm

The round transformation of the proposed LAO-3D block cipher consists of three different functions as shown in Algorithm 1.

Algorithm 1 Pseudocode of the LAO-3D block cipher (encryption).

```
$RoundKeysGeneration(Key)$
$AddRoundKey(State, RoundKey_0)$
for $i = 1$ to 20 do
    $SubColumn(State)$
    $Double3DRotation(State, RoundKey_i)$
end for
```

In this notation, all three functions that include the AddRoundKey, SubColumn, and Double3DRotation are operated on arrays where the inputs are the cipher State and RoundKey. The overall encryption and decryption processes of the LAO-3D algorithm are displayed in Figure 5.

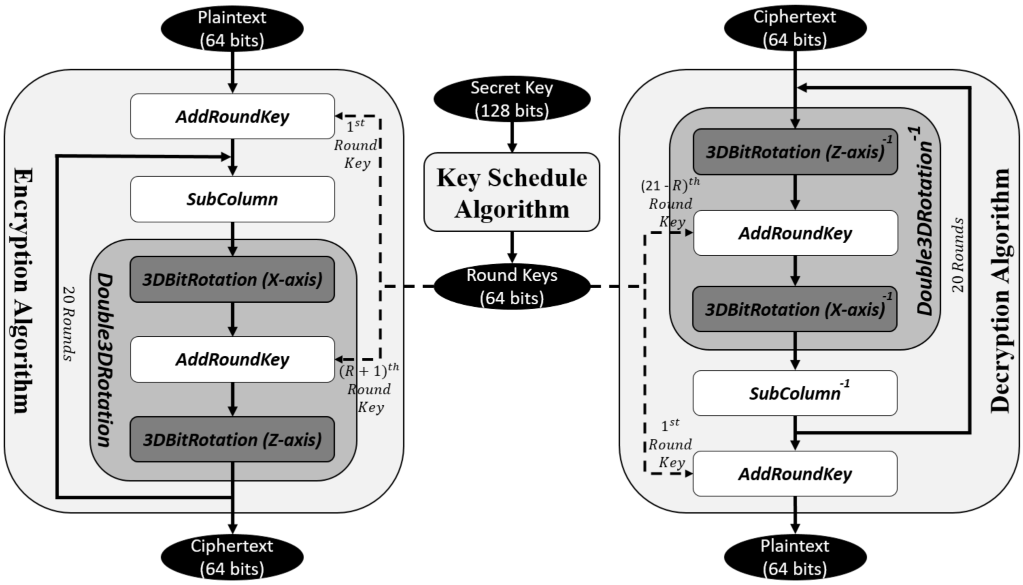


Figure 5. LAO-3D Block Cipher Process.

For the decryption process, the functions are to be operated on in reverse order with 20 rounds of the Double3DRotation and SubColumn functions followed by AddRoundKey. Similarly, all RoundKeys are applied in reverse order in the course of the decryption. Algorithm 2 displays all functional transformations of the algorithm.

Algorithm 2 Pseudocode of the LAO-3D block cipher (decryption).

```
$RoundKeysGeneration(Key)$
for $i = 1$ to 20 do
    $Double3DRotation^{-1}(State, RoundKey_{21-i})$
    $SubColumn^{-1}(State)$
end for
$AddRoundKey(State, RoundKey_0)$
```

3.2.1. AddRoundKey

In the AddRoundKey function, the cipher State $A = \{(a_{3,15},\ldots,a_{3,0}),(a_{2,15},\ldots,a_{2,0}),(a_{1,15},\ldots,a_{1,0}),(a_{0,15},\ldots,a_{0,0}) : a_{i,j} \in A$ for $i = (3,\ldots,0)$ and $j = (15,\ldots,0)\}$ applied a bitwise XOR with the RoundKey $K = \{(k_{3,15},\ldots,k_{3,0}),(k_{2,15},\ldots,k_{2,0}),(k_{1,15},\ldots,k_{1,0}),(k_{0,15},\ldots,k_{0,0}) : k_{i,j} \in K$ for $i = (3,\ldots,0)$ and $j = (15,\ldots,0)\}$ to produce the cipher State $B = \{(b_{3,15},\ldots,b_{3,0}),(b_{2,15},\ldots,b_{2,0}),(b_{1,15},\ldots,b_{1,0}),(b_{0,15},\ldots,b_{0,0}) : b_{i,j} \in B$ for $i = (3,\ldots,0)$ and $j = (15,\ldots,0)\}$ as derived in equation (1). The 64 bits RoundKey are generated using the key schedule algorithm.

$$B = A \oplus K \tag{1}$$

3.2.2. Sub-Column

SubColumn is a non-linear bit substitution that runs independently on every column of the State. This function implements the PRESENT S-box that applies the 4-by-4-bit substitution operation as shown in Table 1 [25]. The S-box is chosen because of its competitive cryptographic characteristics and small hardware footprint.

Table 1. S-box.

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S(x)	5	E	F	8	C	1	2	D	B	4	6	3	0	7	9	A

The input for the S-box used in the SubColumn function is represented by $Col_j = \{(b_{3,j},\ldots,b_{0,j}) : b_{i,j} \in Col_j$ for $i = (3,\ldots,0)$ and $j = (15,\ldots,0)\}$ and the output is $S(Col_j)$, which can be represented by $\{(c_{3,j},\ldots,c_{0,j}) : c_{i,j} \in S(Col_j)$ for $i = (3,\ldots,0)$ and $j = (15,\ldots,0)\}$ as depicted in Figure 6.

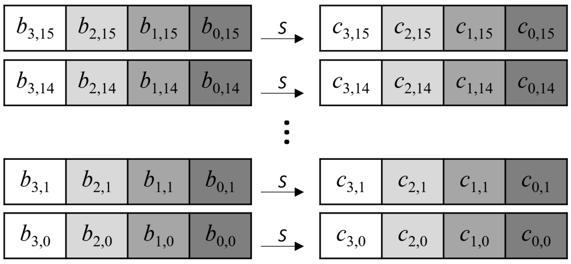
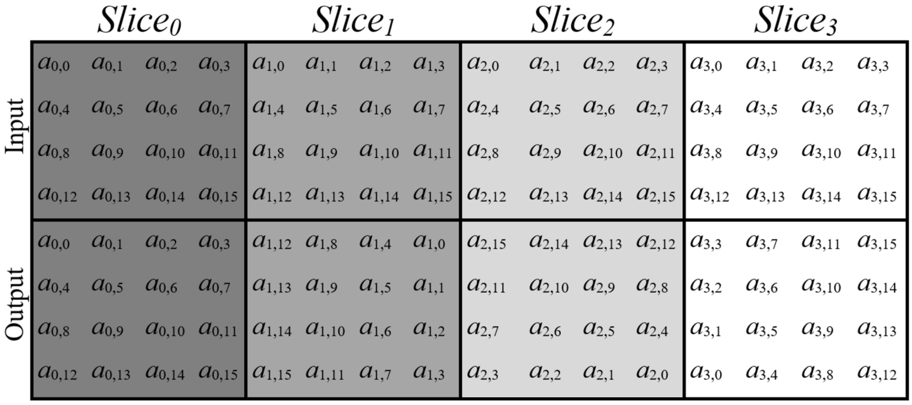

$b_{3,15}$ $b_{2,15}$ $b_{1,15}$ $b_{0,15}$ $\xrightarrow{S}$ $c_{3,15}$ $c_{2,15}$ $c_{1,15}$ $c_{0,15}$
$b_{3,14}$ $b_{2,14}$ $b_{1,14}$ $b_{0,14}$ $\xrightarrow{S}$ $c_{3,14}$ $c_{2,14}$ $c_{1,14}$ $c_{0,14}$
$b_{3,1}$ $b_{2,1}$ $b_{1,1}$ $b_{0,1}$ $\xrightarrow{S}$ $c_{3,1}$ $c_{2,1}$ $c_{1,1}$ $c_{0,1}$
$b_{3,0}$ $b_{2,0}$ $b_{1,0}$ $b_{0,0}$ $\xrightarrow{S}$ $c_{3,0}$ $c_{2,0}$ $c_{1,0}$ $c_{0,0}$

Figure 6. SubColumn Function.

3.2.3. Double3DRotation

The Double3DRotation is constructed by combining three functions that include the 3DBitRotation_*X*-axis, AddRoundKey, and 3DBitRotation_*Z*-axis to provide better permutation to the proposed block cipher.

1. 3DBitRotation_*X*-axis: A specific rotation formation that follows a clockwise direction at the *X*-axis. The cipher State of every Slice before and after executing the 3DBitRotation_*X*-axis function is shown in Figure 7. Values of the 3DBitRotation_*X* -axis function can be presented in the form of a permutation table as shown in Table 2.



	$Slice_0$				$Slice_1$				$Slice_2$				$Slice_3$			
Input	$a_{0,0}$	$a_{0,1}$	$a_{0,2}$	$a_{0,3}$	$a_{1,0}$	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$	$a_{2,0}$	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$	$a_{3,0}$	$a_{3,1}$	$a_{3,2}$	$a_{3,3}$
	$a_{0,4}$	$a_{0,5}$	$a_{0,6}$	$a_{0,7}$	$a_{1,4}$	$a_{1,5}$	$a_{1,6}$	$a_{1,7}$	$a_{2,4}$	$a_{2,5}$	$a_{2,6}$	$a_{2,7}$	$a_{3,4}$	$a_{3,5}$	$a_{3,6}$	$a_{3,7}$
	$a_{0,8}$	$a_{0,9}$	$a_{0,10}$	$a_{0,11}$	$a_{1,8}$	$a_{1,9}$	$a_{1,10}$	$a_{1,11}$	$a_{2,8}$	$a_{2,9}$	$a_{2,10}$	$a_{2,11}$	$a_{3,8}$	$a_{3,9}$	$a_{3,10}$	$a_{3,11}$
	$a_{0,12}$	$a_{0,13}$	$a_{0,14}$	$a_{0,15}$	$a_{1,12}$	$a_{1,13}$	$a_{1,14}$	$a_{1,15}$	$a_{2,12}$	$a_{2,13}$	$a_{2,14}$	$a_{2,15}$	$a_{3,12}$	$a_{3,13}$	$a_{3,14}$	$a_{3,15}$
Output	$a_{0,0}$	$a_{0,1}$	$a_{0,2}$	$a_{0,3}$	$a_{1,12}$	$a_{1,8}$	$a_{1,4}$	$a_{1,0}$	$a_{2,15}$	$a_{2,14}$	$a_{2,13}$	$a_{2,12}$	$a_{3,3}$	$a_{3,7}$	$a_{3,11}$	$a_{3,15}$
	$a_{0,4}$	$a_{0,5}$	$a_{0,6}$	$a_{0,7}$	$a_{1,13}$	$a_{1,9}$	$a_{1,5}$	$a_{1,1}$	$a_{2,11}$	$a_{2,10}$	$a_{2,9}$	$a_{2,8}$	$a_{3,2}$	$a_{3,6}$	$a_{3,10}$	$a_{3,14}$
	$a_{0,8}$	$a_{0,9}$	$a_{0,10}$	$a_{0,11}$	$a_{1,14}$	$a_{1,10}$	$a_{1,6}$	$a_{1,2}$	$a_{2,7}$	$a_{2,6}$	$a_{2,5}$	$a_{2,4}$	$a_{3,1}$	$a_{3,5}$	$a_{3,9}$	$a_{3,13}$
	$a_{0,12}$	$a_{0,13}$	$a_{0,14}$	$a_{0,15}$	$a_{1,15}$	$a_{1,11}$	$a_{1,7}$	$a_{1,3}$	$a_{2,3}$	$a_{2,2}$	$a_{2,1}$	$a_{2,0}$	$a_{3,0}$	$a_{3,4}$	$a_{3,8}$	$a_{3,12}$

Figure 7. Cipher State of 3DBitRotation (*X*-axis).

Table 2. 3DBitRotation Permutation Table (*X*-axis).

Permutation Table (*X*-axis Rotation)							
0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
28	24	20	16	29	25	21	17
30	26	22	18	31	27	23	19
47	46	45	44	43	42	41	40
39	38	37	36	35	34	33	32
51	55	59	63	50	54	58	62
49	53	57	61	48	52	56	60

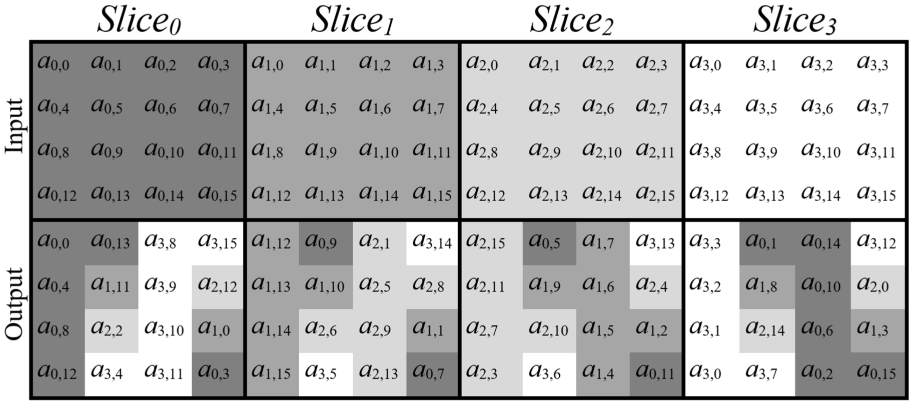
The input for the 3DBitRotation_*X*-axis is the output of the SubColumn function defined as $C = \{(c_{3,15}, \ldots, c_{3,0}), (c_{2,15}, \ldots, c_{2,0}), (c_{1,15}, \ldots, c_{1,0}), (c_{0,15}, \ldots, c_{0,0}) : c_{i,j} \in C$ for $i = (3, \ldots, 0)$ and $j = (15, \ldots, 0)\}$ is permutated using the permutation table and produced output $D = \{(d_{3,15}, \ldots, d_{3,0}), (d_{2,15}, \ldots, d_{2,0}), (d_{1,15}, \ldots, d_{1,0}), (d_{0,15}, \ldots, d_{0,0})$ $: d_{i,j} \in D$ for $i = (3, \ldots, 0)$ and $j = (15, \ldots, 0)\}$ as defined in the following equation.

$$D = P(C) \tag{2}$$

2. AddRoundKey: In this operation, the output from the 3DBitRotation_*X*-axis function $D = \{(d_{3,15}, \ldots, d_{3,0}), (d_{2,15}, \ldots, d_{2,0}), (d_{1,15}, \ldots, d_{1,0}), (d_{0,15}, \ldots, d_{0,0}) : d_{i,j} \in D$ for $i = (3, \ldots, 0)$ and $j = (15, \ldots, 0)\}$ is XOR with the RoundKey $K = \{(k_{3,15}, \ldots, k_{3,0}), (k_{2,15}, \ldots, k_{2,0}), (k_{1,15}, \ldots, k_{1,0}), (k_{0,15}, \ldots, k_{0,0}) : k_{i,j} \in K$ for $i = (3, \ldots, 0)$ and $j = (15, \ldots, 0)\}$. The result from the operation, cipher State $E = \{(e_{3,15}, \ldots, e_{3,0}), (e_{2,15}, \ldots, e_{2,0}), (e_{1,15}, \ldots, e_{1,0}), (e_{0,15}, \ldots, e_{0,0}) : e_{i,j} \in E$ for $i = (3, \ldots, 0)$ and $j = (15, \ldots, 0)\}$ is obtained from the following equation.

$$E = D \oplus K \tag{3}$$

3. 3DBitRotation_*Z*-axis: A specific rotation formation that follows a clockwise direction at the *Z*-axis. The position of each cipher Slice before and after executing the 3DBitRotation_*Z*-axis function is displayed in Figure 8. Permutation values of the function are shown in Table 3.



	$Slice_0$				$Slice_1$				$Slice_2$				$Slice_3$			
Input	$a_{0,0}$	$a_{0,1}$	$a_{0,2}$	$a_{0,3}$	$a_{1,0}$	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$	$a_{2,0}$	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$	$a_{3,0}$	$a_{3,1}$	$a_{3,2}$	$a_{3,3}$
	$a_{0,4}$	$a_{0,5}$	$a_{0,6}$	$a_{0,7}$	$a_{1,4}$	$a_{1,5}$	$a_{1,6}$	$a_{1,7}$	$a_{2,4}$	$a_{2,5}$	$a_{2,6}$	$a_{2,7}$	$a_{3,4}$	$a_{3,5}$	$a_{3,6}$	$a_{3,7}$
	$a_{0,8}$	$a_{0,9}$	$a_{0,10}$	$a_{0,11}$	$a_{1,8}$	$a_{1,9}$	$a_{1,10}$	$a_{1,11}$	$a_{2,8}$	$a_{2,9}$	$a_{2,10}$	$a_{2,11}$	$a_{3,8}$	$a_{3,9}$	$a_{3,10}$	$a_{3,11}$
	$a_{0,12}$	$a_{0,13}$	$a_{0,14}$	$a_{0,15}$	$a_{1,12}$	$a_{1,13}$	$a_{1,14}$	$a_{1,15}$	$a_{2,12}$	$a_{2,13}$	$a_{2,14}$	$a_{2,15}$	$a_{3,12}$	$a_{3,13}$	$a_{3,14}$	$a_{3,15}$
Output	$a_{0,0}$	$a_{0,13}$	$a_{3,8}$	$a_{3,15}$	$a_{1,12}$	$a_{0,9}$	$a_{2,1}$	$a_{3,14}$	$a_{2,15}$	$a_{0,5}$	$a_{1,7}$	$a_{3,13}$	$a_{3,3}$	$a_{0,1}$	$a_{0,14}$	$a_{3,12}$
	$a_{0,4}$	$a_{1,11}$	$a_{3,9}$	$a_{2,12}$	$a_{1,13}$	$a_{1,10}$	$a_{2,5}$	$a_{2,8}$	$a_{2,11}$	$a_{1,9}$	$a_{1,6}$	$a_{2,4}$	$a_{3,2}$	$a_{1,8}$	$a_{0,10}$	$a_{2,0}$
	$a_{0,8}$	$a_{2,2}$	$a_{3,10}$	$a_{1,0}$	$a_{1,14}$	$a_{2,6}$	$a_{2,9}$	$a_{1,1}$	$a_{2,7}$	$a_{2,10}$	$a_{1,5}$	$a_{1,2}$	$a_{3,1}$	$a_{2,14}$	$a_{0,6}$	$a_{1,3}$
	$a_{0,12}$	$a_{3,4}$	$a_{3,11}$	$a_{0,3}$	$a_{1,15}$	$a_{3,5}$	$a_{2,13}$	$a_{0,7}$	$a_{2,3}$	$a_{3,6}$	$a_{1,4}$	$a_{0,11}$	$a_{3,0}$	$a_{3,7}$	$a_{0,2}$	$a_{0,15}$

Figure 8. Cipher State of the 3D Bit Rotation (*Z*-axis).

Table 3. The 3D Bit Rotation Permutation Table (*Z*-axis).

Permutation Table (*Z*-axis Rotation)							
0	13	62	51	4	29	58	35
8	45	54	19	12	61	50	3
16	9	46	55	20	25	42	39
24	41	38	23	28	57	34	7
32	5	30	59	36	21	26	43
40	37	22	27	44	53	18	11
48	1	14	63	52	17	10	47
56	33	6	31	60	49	2	15

The 3DBitRotation_*Z*-axis permutes the output of the AddRoundKey function defined as $E = \{(e_{3,15}, \ldots, e_{3,0}), (e_{2,15}, \ldots, e_{2,0}), (e_{1,15}, \ldots, e_{1,0}), (e_{0,15}, \ldots, e_{0,0}) : e_{i,j} \in E$ for $i = (3, \ldots, 0)$ and $j = (15, \ldots, 0)\}$ using the permutation table. Equation (4) derived the results from the operation defined as $F = \{(f_{3,15}, \ldots, f_{3,0}), (f_{2,15}, \ldots, f_{2,0}), (f_{1,15}, \ldots, f_{1,0}), (f_{0,15}, \ldots, f_{0,0}) : f_{i,j} \in F$ for $i = (3, \ldots, 0)$ and $j = (15, \ldots, 0)\}$.

$$F = P(E) \tag{4}$$

3.3. Key Schedule Algorithm

The RoundKeys are derived using the key scheduling algorithm from a secret key. Overall, the total number of RoundKeys bits is equal to the round number plus one and multiplied by the block length. For an algorithm with a 64-bit block and 20 rounds, 1344 RoundKey bits are required. The RoundKeyExtraction process selected 64 bits RoundKey from the revised key after operating the NonceXOR, KeySubColumn, and RowTransformation functions, as shown in Figure 9.

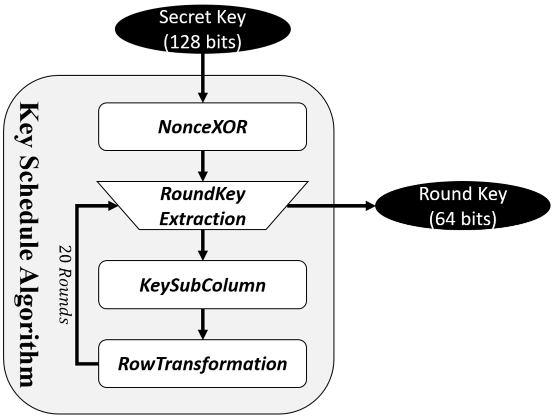


Figure 9. LAO-3D Key Schedule Process.

3.3.1. NonceXOR

In the initial stage of the key scheduling algorithm, the secret key $S = \{(s_{127}, \ldots, s_0) : s_{j-1} \in S \text{ for } j = (128, \ldots, 1)\}$ is XOR with a Nonce $N = \{(n_{127}, \ldots, n_0) : n_{j-1} \in N$ for $j = (128, \ldots, 1)\}$ to obtain the key state $V = \{(v_{127}, \ldots, v_0) : v_{j-1} \in V$ for $j = (128, \ldots, 1)\}$ as defined in Equation (5). The Nonce are random 128 bits of data that are stored in the algorithm.

$$V = S \oplus N \tag{5}$$

The introduction of the Nonce in the algorithm can improve the confusion of RoundKey generation. Additionally, the Nonce could remove the correlation between the secret key and the generated RoundKeys. By implementing different Nonce, a secret key will not be able to generate the same set of RoundKeys, which provide extra security to the users.

3.3.2. RoundKeyExtraction

LAO-3D cipher utilizes a 128-bit key that generates a set of 21 RoundKeys, including an initial RoundKey. The key schedule algorithm requires extracting 64 bits from the key state. Let $V = \{(v_0, \ldots, v_{127}) : v_{j-1} \in V$ for $j = (1, \ldots, 128)\}$, defining the key state. Every 32 bits are arranged in reverse order to obtain the RowKey as displayed in Figure 10.

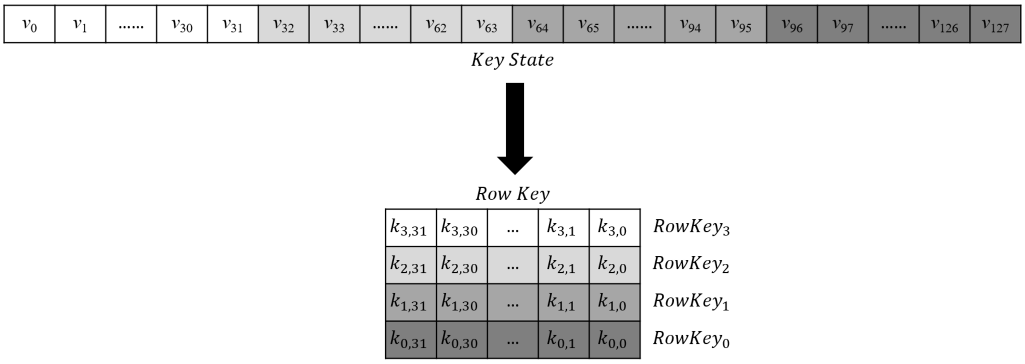

v_0 v_1 ...... v_{30} v_{31} v_{32} v_{33} ...... v_{62} v_{63} v_{64} v_{65} ...... v_{94} v_{95} v_{96} v_{97} ...... v_{126} v_{127}
Key State
Row Key
$k_{3,31}$ $k_{3,30}$ ... $k_{3,1}$ $k_{3,0}$ $RowKey_3$
$k_{2,31}$ $k_{2,30}$ ... $k_{2,1}$ $k_{2,0}$ $RowKey_2$
$k_{1,31}$ $k_{1,30}$ ... $k_{1,1}$ $k_{1,0}$ $RowKey_1$
$k_{0,31}$ $k_{0,30}$ ... $k_{0,1}$ $k_{0,0}$ $RowKey_0$

Figure 10. RowKey.

In order to generate 64 bits of the i^{th} round key K_i for $i = (0, \ldots, 20)$, 16 rightmost columns of $RowKey_3$, $RowKey_2$, $RowKey_1$, and $RowKey_0$ are appended, as shown in Figure 11. During the encryption and decryption processes, these round keys will be used in the AddRoundKey function.

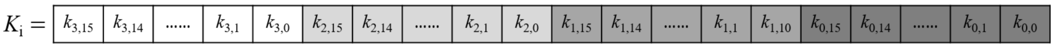

$K_i =$ $k_{3,15}$ $k_{3,14}$ ...... $k_{3,1}$ $k_{3,0}$ $k_{2,15}$ $k_{2,14}$ ...... $k_{2,1}$ $k_{2,0}$ $k_{1,15}$ $k_{1,14}$ ...... $k_{1,1}$ $k_{1,10}$ $k_{0,15}$ $k_{0,14}$ ...... $k_{0,1}$ $k_{0,0}$

Figure 11. RoundKey.

3.3.3. KeySubColumn

Upon completion of the RoundKeyExtraction function, the KeySubColumn will be used to modify the RoundKey value in each round. The S-box is used to reconstruct four uppermost rows and eight rightmost columns of the key state, i.e., $(k'_{3,j}, \ldots, k'_{0,j}) = S(k_{3,j}, \ldots, k_{0,j})$ for $j = (7, \ldots, 0)$ as depicted in Figure 12.

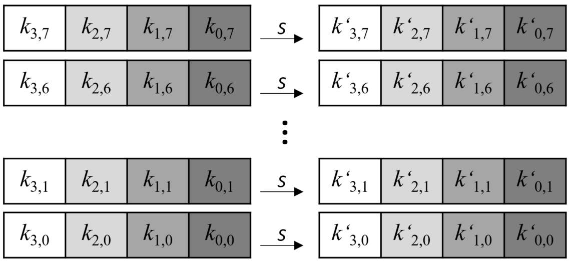

$k_{3,7}$ $k_{2,7}$ $k_{1,7}$ $k_{0,7}$ $\xrightarrow{S}$ $k'_{3,7}$ $k'_{2,7}$ $k'_{1,7}$ $k'_{0,7}$
$k_{3,6}$ $k_{2,6}$ $k_{1,6}$ $k_{0,6}$ $\xrightarrow{S}$ $k'_{3,6}$ $k'_{2,6}$ $k'_{1,6}$ $k'_{0,6}$
$k_{3,1}$ $k_{2,1}$ $k_{1,1}$ $k_{0,1}$ $\xrightarrow{S}$ $k'_{3,1}$ $k'_{2,1}$ $k'_{1,1}$ $k'_{0,1}$
$k_{3,0}$ $k_{2,0}$ $k_{1,0}$ $k_{0,0}$ $\xrightarrow{S}$ $k'_{3,0}$ $k'_{2,0}$ $k'_{1,0}$ $k'_{0,0}$

Figure 12. KeySubColumn Function.

3.3.4. RowTransformation

The new key state, known as the updated round key produced from the KeySubColumn output, is rearranged into RowKeys representation and revised using the following RowTransformation, as displayed in Figure 13.

1. $RowKey'_0 = (RowKey_0 <<< 8) \oplus RowKey_1$
2. $RowKey'_1 = RowKey_2$
3. $RowKey'_2 = (RowKey_2 <<< 16) \oplus RowKey_3$
4. $RowKey'_3 = RowKey_0$

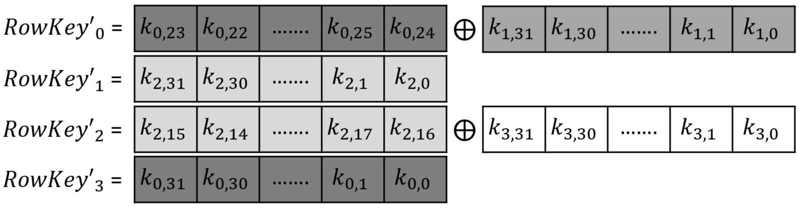

$RowKey'_0 =$ $k_{0,23}$ $k_{0,22}$ .... $k_{0,25}$ $k_{0,24}$ $\oplus$ $k_{1,31}$ $k_{1,30}$ .... $k_{1,1}$ $k_{1,0}$
$RowKey'_1 =$ $k_{2,31}$ $k_{2,30}$ .... $k_{2,1}$ $k_{2,0}$
$RowKey'_2 =$ $k_{2,15}$ $k_{2,14}$ .... $k_{2,17}$ $k_{2,16}$ $\oplus$ $k_{3,31}$ $k_{3,30}$ .... $k_{3,1}$ $k_{3,0}$
$RowKey'_3 =$ $k_{0,31}$ $k_{0,30}$ .... $k_{0,1}$ $k_{0,0}$

Figure 13. RowTransformation Function.

3.4. Test Vectors

Test vectors specify the output of a cryptographic algorithm from a given input to provide a copy of the verification data for developers and executors [26]. The input and its corresponding output data of the proposed LAO-3D block cipher are presented in Table 4.

Table 4. Test Vectors of the LAO-3D Block Cipher.

Key	Plaintext	Ciphertext
00000000000000000000000000000000	0000000000000000	5F07F85C4E5217E7
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	0000000000000000	11F40B91480C2776
65CA1E79B03D8F421A4C6F392DB7508E	0000000000000000	217B3379252F9476
00000000000000000000000000000000	FFFFFFFFFFFFFFFF	14741B345A2729D2
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	FFFFFFFFFFFFFFFF	69F746C6D6855E6C
65CA1E79B03D8F421A4C6F392DB7508E	FFFFFFFFFFFFFFFF	B3435F9A4DBB4EAA
00000000000000000000000000000000	C56B90AD3EF84712	E735E158FEA44714
FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	C56B90AD3EF84712	DA82F25137632062
65CA1E79B03D8F421A4C6F392DB7508E	C56B90AD3EF84712	4D00B854AD515FF8

4. Security Analysis, Performance Test, and Application

This section presents the security analyses performed in assessing the full 20 rounds of the proposed LAO-3D block cipher. The evaluation methods were implemented to distinguish the strength of the lightweight block cipher and possibly be used in various cryptographic primitives.

A decent algorithm must establish a complex relationship between the plaintext, ciphertext, and key, which is called confusion. The encryption method must also scatter the plaintext modifications over the entire ciphertext, which is called diffusion [27]. Any encryption algorithm should provide both confusion and diffusion properties to ensure information security. Confusion is obtained through substitution; meanwhile, diffusion is achieved with permutation operations. Confusion and diffusion will produce a certain randomness degree in the block cipher such that no ciphertext pattern can be recognized.

To evaluate the confusion and diffusion properties, three analyses were executed on the proposed block cipher—the correlation coefficient, bit error, and key sensitivity tests. In addition, a randomness evaluation was performed to examine the randomness characteristics of the algorithm. Finally, two cryptanalysis attacks were carried out to observe the strength of the LAO-3D block cipher.

Apart from the security analysis, a performance test was conducted to measure the speed, throughput, and required cycles per byte to execute the encryption. For benchmarking purposes, the experimental results were compared with existing lightweight block ciphers, such as GIFT, KATAN, KLEIN, LED, LRBC, PRESENT, PRINCE, PRINT, QTL, RECTANGLE, SIMECK, SPECK, and TEA. These algorithms were chosen because of their similarity in terms of block and key sizes that were no bigger than 64 bits and 128 bits, respectively, to distinguish their security strengths and performances within the same lightweight category.

In order to demonstrate the implementation of the LAO-3D lightweight block cipher, a mobile encryption application was developed to observe the functionality of the proposed algorithm. The application was built using C++ programming on Microsoft Visual Studio 2008 and Android Studio development software.

4.1. Correlation Coefficient Test

The avalanche effect examines the non-linear properties of a cipher. A non-linear transformation provides a confusion characteristic that relies on the generated output from a particular input data. The correlation coefficient, bit error, and key sensitivity experiments were carried out to monitor the avalanche effect of the block cipher. For the sample generation, a pseudorandom bit generator was used to produce the keys and plaintexts. The avalanche effect, AE, is defined in the following equation.

$$AE = \frac{1}{s}\sum_{i=1}^{s}|c_i - p_i| \tag{6}$$

where s is the plaintext/ciphertext length, while p_i and c_i are the i^{th} positions of the plaintext and ciphertext bits for $i = (0, \ldots, 63)$.

The correlation coefficient analyzes the non-linear relationship between plaintext and ciphertext [28]. The correlation coefficient, R, is able to discriminate the effect of confusion on the block cipher. The coefficient values range from −1 to +1, where the acceptable result ranges are shown in Table 5.

Table 5. Correlation Coefficient Result Indications.

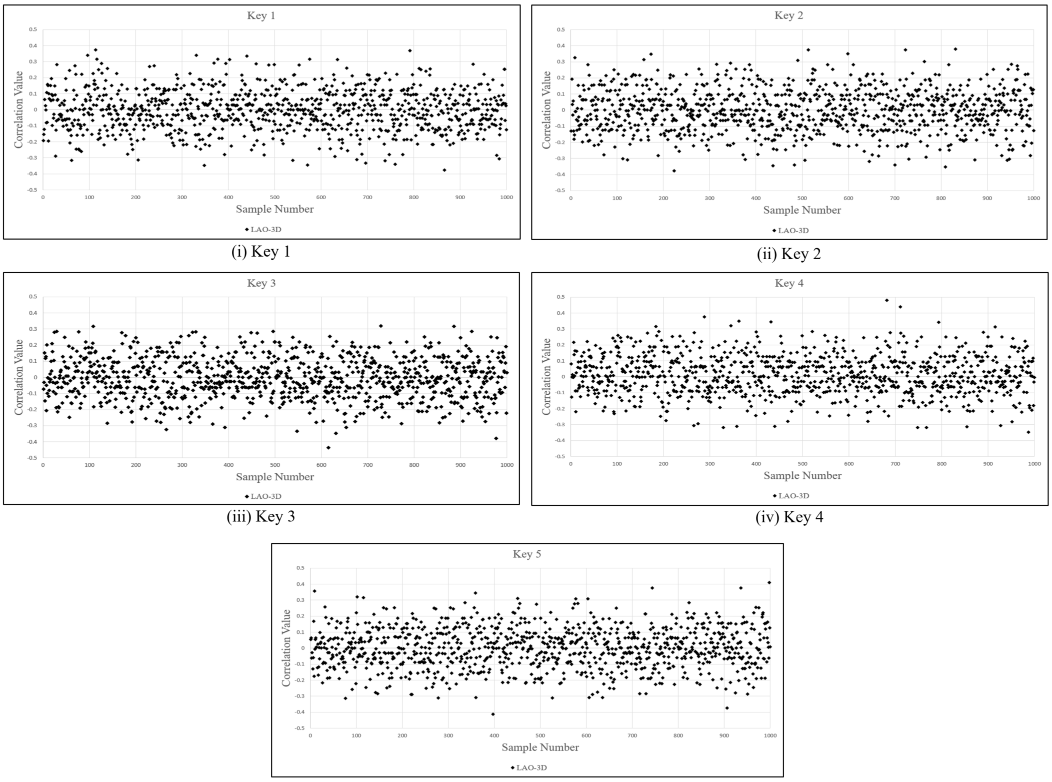
Result	Correlation Potential
$R = 0$	Non-linear relationship
$0 < R \leq 0.3$ or $-0.3 \leq R < 0$	Weak positive/negative linear relationship
$0.3 < R < 0.7$ or $-0.7 < R < -0.3$	Moderate positive/negative linear relationship
$0.7 \leq R < 1$ or $-1 < R \leq -0.7$	Strong positive/negative linear relationship
$R = 1$ or $R = -1$	Perfect positive/negative linear relationship

For the expression of the correlation coefficient, R, refer to the following equation:

$$R = \frac{\sum_{i=1}^{s}(p_i - AE)(c_i - AE)}{\sqrt{\sum_{i=1}^{s}(p_i - AE)^2}\sqrt{\sum_{i=1}^{s}(c_i - AE)^2}} \tag{7}$$

where AE represents the avalanche effect from (6), p_i indicates the i^{th} plaintext bit, and c_i defines the i^{th} ciphertext bit for $i = (0, \ldots, 63)$.

The correlation values from the analysis conducted on the plaintext and ciphertext were remarked. A total of 1000 random plaintexts and five random keys were examined on the algorithm. Figure 14 displays the scatter charts of the analysis results. The values in the charts are plotted according to the computed correlation coefficient based on the key and plaintext inputs.



(i) Key 1

(ii) Key 2

(iii) Key 3

(iv) Key 4

(v) Key 5

Figure 14. Scatter Charts of the Correlation Coefficient Results.

A summary of the results presented in the scatter charts is shown in Table 6. According to the experimental results, the LAO-3D block cipher offers performance advantages (98.2% of the correlation coefficients, in the range $0 < R \leq 0.3$ and $-0.3 \leq R < 0$), indicating a weak linear relationship. The results generated by all five keys show that the adoption of the 3D cipher design on the LAO-3D block cipher provided high correlation coefficient characteristics.

Table 6. Correlation Coefficient Results.

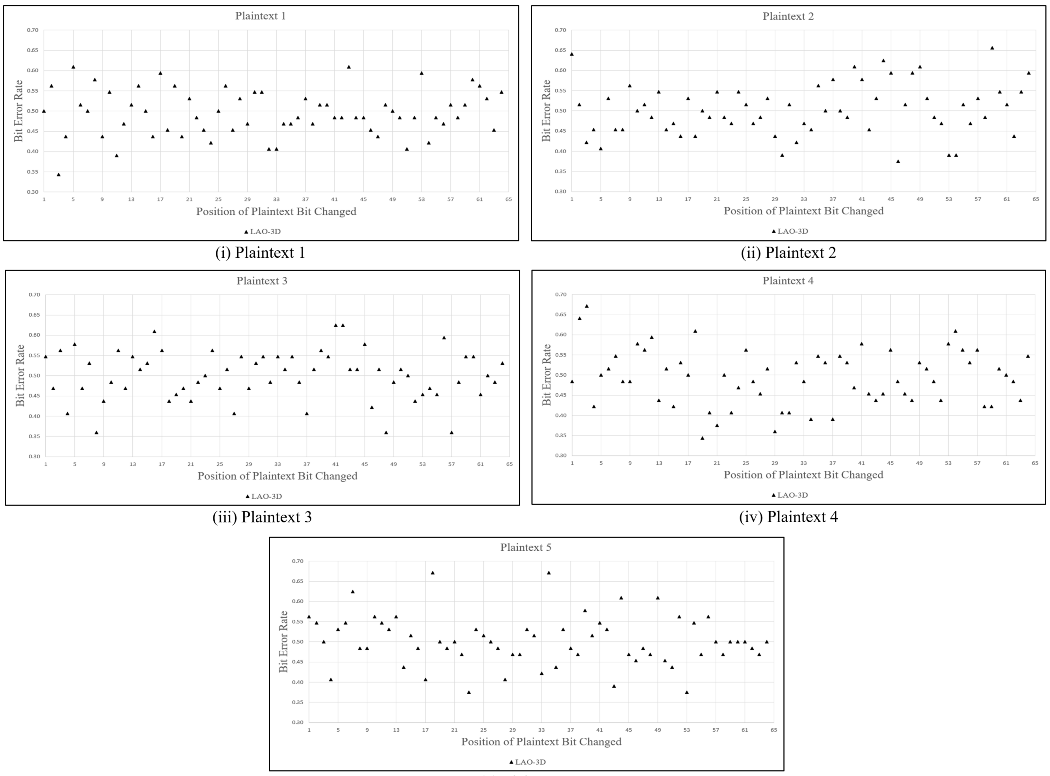
Result	Key 1	Key 2	Key 3	Key 4	Key 5	Average
$R = 0$	0	0	0	0	0	0%
$0 < R \leq 0.3$ and $-0.3 \leq R < 0$	979	977	990	982	982	98.2%
$0.3 < R < 0.7$ and $-0.7 < R < -0.3$	21	23	10	18	18	1.8%
$0.7 \leq R < 1$ and $-1 < R \leq -0.7$	0	0	0	0	0	0%
$R = -1$ and $R = 1$	0	0	0	0	0	0%

4.2. Bit Error Test

The bit error computes the differences in the ciphertext affected by the plaintext modifications. The number of changed ciphertext bits after modification of a plaintext bit is defined as the bit error. For the bit error test, the optimum result is 0.5 or 50% plaintext bit alterations [29]. The formulation of the bit error rate, BER, is defined in the following equation.

$$BER = \frac{\text{Number of ciphertext bit difference}}{\text{Total number of ciphertext bit}} \tag{8}$$

The purpose of the bit error test is to measure the relationship between plaintext and ciphertext. A random key and five random plaintexts were used to examine the LAO-3D block cipher. As seen in Figure 15, the experimental results are presented in the scatter charts. The data plots in each chart display the bit error values that represent the changes of ciphertext upon plaintext modifications.



(i) Plaintext 1

(ii) Plaintext 2

(iii) Plaintext 3

(iv) Plaintext 4

(v) Plaintext 5

Figure 15. Scatter Charts of Bit Error Rate Results.

An overall summarization of the bit error results presented in the scatter charts is shown in Table 7. After modifying the plaintext bits, the BER of the LAO-3D block cipher is more likely to be close to 0.5. The results generated by all five plaintexts prove that the LAO-3D block cipher achieved optimum bit error test results.

Table 7. Bit Error Results.

Input	Average Different Bits	Average Bit Error Rate
Plaintext 1	31.703125	0.495361
Plaintext 2	32.203125	0.503174
Plaintext 3	32.046875	0.500732
Plaintext 4	32.046875	0.500732
Plaintext 5	32.156250	0.502441
Average	50.05%	50.05%

By carefully examining the data, it is found that the LAO-3D block cipher obtains an average 50% bit error rate, which verifies that the ciphertext is entirely modified with a single alteration in the plaintext bit. Table 8 presents the comparison of the avalanche effect on the modification of the plaintext against existing block ciphers [10]. The results achieved by the proposed LAO-3D surpass the earlier works in terms of non-linearity between the plaintext and its corresponding ciphertext.

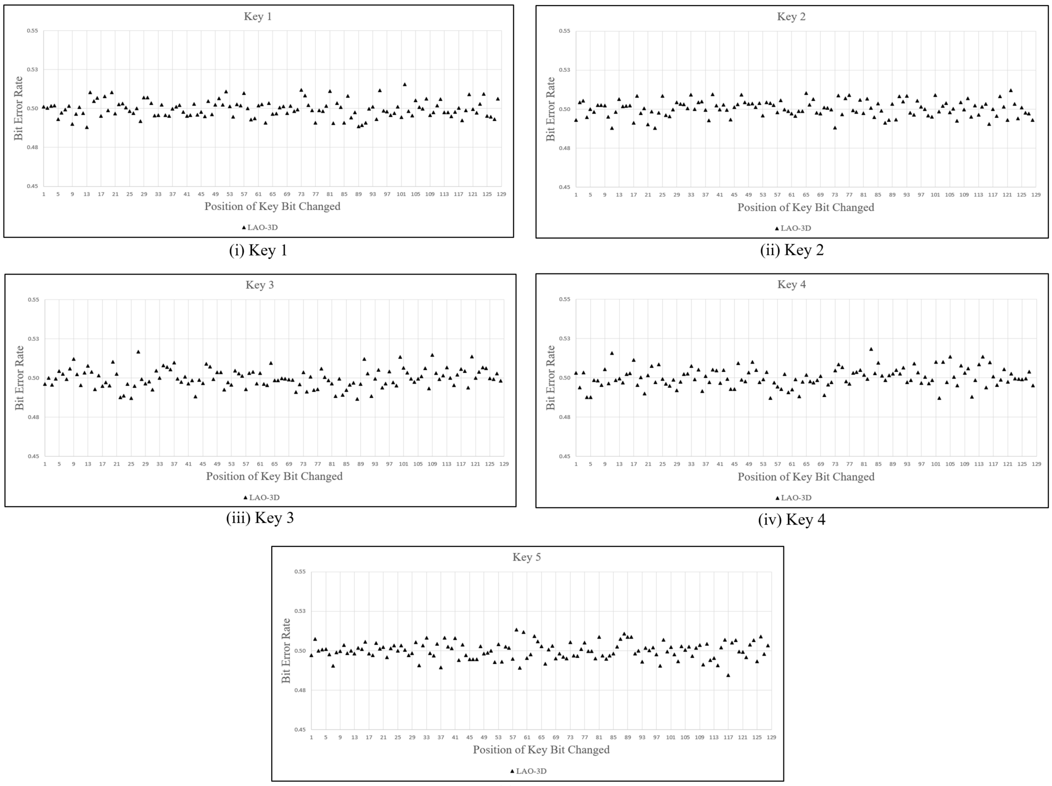
Table 8. Avalanche Effect on Plaintext Modifications.

Algorithm	Average Avalanche Effect
LAO-3D	50.05%
LED	52.83%
LRBC	58.00%
PRINCE	51.18%
PRINT	49.08%
QTL	52.56%
SIMECK	53.00%
TEA	49.12%

4.3. Key Sensitivity Test

The key sensitivity test examines the affected ciphertext upon key alteration [30]. An algorithm has a high-security level against the key sensitivity attack if changing a small portion of the key will cause major modifications on the ciphertext bits. During the observation of key sensitiveness, the key bit is manipulated by changing one bit from its first to the last bit position.

The bit error rate equation is applied to compute the experimental results. The result of a decent cryptographic algorithm must be close to 0.5 or 50% of the total ciphertext bit modification. In this experiment, a random plaintext and five random keys were included to examine the LAO-3D block cipher. The scatter charts of the experimental results are shown in Figure 16.



(i) Key 1

(ii) Key 2

(iii) Key 3

(iv) Key 4

(v) Key 5

Figure 16. Scatter Charts of the Key Sensitivity Results.

Table 9 summarizes the key sensitivity experimental results presented in the scatter charts. On average, the proposed LAO-3D block cipher obtained 32-bit differences with an approximate 50% bit error rate. The results suggest that LAO-3D has a non-linear relationship between the key and ciphertext that represents a high key sensitivity on the ciphertext.

Table 9. Key Sensitivity Results.

Input	Average Different Bits	Average Bit Error Rate
Key 1	32.000122	0.500002
Key 2	32.028317	0.500442
Key 3	32.007694	0.500120
Key 4	31.948364	0.499193
Key 5	32.013795	0.500216
Average	50.00%	50.00%

Similar observations were made on the key alteration while retaining a constant plaintext performed against other algorithms [10], presented in Table 10. The 50% key sensitivity result denotes that the entire key bits have an impact on every ciphertext bit. Based on the findings, the LAO-3D block cipher showed some improvements over the existing works on the avalanche effect of the key sensitivity.

Table 10. Avalanche Effect on Key Modifications.

Algorithm	Average Avalanche Effect
LAO-3D	50.00%
LED	50.37%
LRBC	55.75%
PRINCE	49.06%
PRINT	46.42%
QTL	50.31%
SIMECK	51.25%
TEA	47.12%

4.4. Randomness Test

The randomness of the proposed LAO-3D block cipher was tested using 15 statistical tests of the NIST Special Publication 800-22 [31]. The NIST accommodates an open-source randomness test application called NIST Statistical Suite. The randomness test application is aimed at varieties of non-random characteristics from the generated ciphertext.

Eight tests inclusive of random excursion (8 p-values), random excursion variant (18 p-values), cumulative sums (2 p-values), binary matrix rank (1 p-value), longest runs of ones (1 p-value), frequency (1 p-value), spectral DFT (1 p-value), and runs (1 p-value) are classified as the non-parameterized tests.

Seven other tests, i.e., non-overlapping (148 p-values), serial (2 p-values), linear complexity (1 p-value), maurer's universal (1 p-value), approximate entropy (1 p-value), overlapping templates (1 p-value), and block frequency (1 p-value) are labeled as parameterized tests that require parameter inputs.

A significant level, α, must be set within 0.1% (0.001) to 1% (0.01) before assessing the randomness test. Moreover, the sample size should at least be the inverse of the significant level ($1 \div 0.01 = 100$ samples). For the test results, the ciphertext is regarded as random if the p-value $\geq \alpha$. Conversely, if the p-value $< \alpha$, the ciphertext is determined as not random.

Equation (9) defines the proportion of test samples that will determine the randomness of an algorithm.

$$p_\alpha = (1-\alpha) - 3\sqrt{\frac{\alpha(1-\alpha)}{s}} \quad (9)$$

where s symbolizes the size of the test sample (1000 ciphertexts) and α indicates the significance level (0.01) that is used in this paper. If the total rejection exceeds the computed proportion p_α, the sample is regarded as not random.

Nine data categories of block cipher were implemented to generate the keys and plaintexts. Every data category generated unique 1000 samples where each sample consisted of 2081 to 15,744 ciphertexts that equaled 133,184 to 1,007,616 bits of data with a size range of 127 to 960 MB. The block and key lengths determine the total number of blocks for each sample [32]. The blocks were derived by appending the ciphertexts to establish large bit sequences.

Table 11 listed each of the data categories that generate multiple lengths of derived ciphertext bits based on the input data. All 15 statistical tests can be applied to evaluate CBC, PCC, RPRK, SPA, and SKA data categories [33]. Due to insufficient data lengths, only 11 tests could be used to examine HDK and LDK, and 10 tests for HDP and LDP.

Table 11. Input of Data Categories.

	Data Category	Key	Plaintext	Derived Blocks	Derived Bits
1.	Strict Key Avalanche (SKA)	123 128-bit keys	One all-zero 64-bit plaintext	15,744	1,007,616
2.	Strict Plaintext Avalanche (SPA)	One all-zero 128-bit key	245 64-bit plaintexts	15,680	1,003,520
3.	Plaintext/Ciphertext Correlation (PCC)	One 128-bit key	15,625 64-bit plaintexts	15,625	1,000,000
4.	Ciphertext Block Chaining Mode (CBCM)	One 128-bit key	One all-zero 64-bit plaintext	15,625	1,000,000
5.	Random Plaintext/Random Key (RPRK)	One 128-bit key	15,625 64-bit plaintexts	15,625	1,000,000
6.	Low-Density Key (LDK)	3241 128-bit keys	8257 64-bit plaintexts	8257	528,448
7.	High-Density Key (HDK)	3241 128-bit keys	8257 64-bit plaintexts	8257	528,448
8.	Low-Density Plaintext (LDP)	2081 128-bit keys	2081 64-bit plaintexts	2081	133,184
9.	High-Density Plaintext (HDP)	2081 128-bit keys	2081 64-bit plaintexts	2081	133,184

The experimental results demonstrate that the LAO-3D block cipher passed all data categories and statistical tests as shown in Tables 12 and 13. In addition, the p-values produced by the block cipher from the NIST Statistical Suite are uniformly distributed since the values are larger than 0.0001, as shown in Tables 14 and 15. The 100% passing rate achieved in the randomness and uniformity test results verified the randomness characteristics of the cipher output. Combinations of substitution and permutation components in LAO-3D have optimized the confusion and diffusion properties of the algorithm that contribute to the randomization of the ciphertext.

Table 12. Randomness Analysis Results (CBC, PCC, RPRK, SKA, and SPA).

		Data Category				
	Statistical Test	**CBC**	**PCC**	**RPRK**	**SKA**	**SPA**
		Range of Acceptable Rejection: [0, 20]				
1.	Runs	992/1000	987/1000	990/1000	990/1000	988/1000
2.	Frequency	989/1000	992/1000	993/1000	984/1000	989/1000
3.	Spectral DFT	991/1000	996/1000	988/1000	990/1000	985/1000
4.	Block Frequency	986/1000	986/1000	993/1000	994/1000	996/1000
5.	Binary Matrix Rank	987/1000	985/1000	992/1000	995/1000	989/1000
6.	Approximate Entropy	994/1000	995/1000	989/1000	988/1000	988/1000
7.	Longest Runs of Ones	989/1000	991/1000	994/1000	997/1000	985/1000
8.	Serial	992/1000	992/1000	990/1000	991/1000	990/1000
9.	Cumulative Sums	991/1000	992/1000	989/1000	989/1000	991/1000
10.	Non-Overlapping Templates	990/1000	990/1000	990/1000	990/1000	990/1000
11.	Maurer's Universal	992/1000	989/1000	988/1000	983/1000	986/1000
12.	Linear Complexity	986/1000	986/1000	992/1000	989/1000	991/1000
13.	Overlapping Template	983/1000	987/1000	992/1000	995/1000	987/1000
		Range of Acceptable Rejection: [0, 14]				
14.	Random Excursion	577/584	609/617	590/595	613/622	627/633
15.	Random Excursion Variant	578/584	611/617	590/595	616/622	627/633

Table 13. Randomness Analysis Results (LDK, HDK, LDP, and HDP).

		Data Category			
	Statistical Test	**LDK**	**HDK**	**LDP**	**HDP**
		Range of Acceptable Rejection: [0, 20]			
1.	Runs	990/1000	986/1000	992/1000	992/1000
2.	Frequency	987/1000	988/1000	992/1000	991/1000
3.	Spectral DFT	988/1000	984/1000	986/1000	984/1000
4.	Block Frequency	994/1000	991/1000	986/1000	990/1000
5.	Binary Matrix Rank	990/1000	989/1000	993/1000	990/1000
6.	Approximate Entropy	987/1000	992/1000	987/1000	989/1000
7.	Longest Runs of Ones	995/1000	989/1000	987/1000	989/1000
8.	Serial	992/1000	989/1000	986/1000	991/1000
9.	Cumulative Sums	985/1000	987/1000	993/1000	986/1000
10.	Non-Overlapping Templates	990/1000	990/1000	990/1000	989/1000
11.	Maurer's Universal	989/1000	992/1000	*	*
12.	Linear Complexity	*	*	*	*
13.	Overlapping Templates	*	*	*	*
		Range of Acceptable Rejection: Not Available			
14.	Random Excursion	*	*	*	*
15.	Random Excursion Variant	*	*	*	*

* no effective trail from the encryption round onward.

4.5. Differential and Linear Cryptanalysis

A cryptanalysis attack is an experimental method used to distinguish a cryptosystem from a random function [34]. To ensure that the confidentiality of the proposed LAO-3D output is robust, it is necessary to examine the security strength of the algorithm against a variety of attack types. Differential cryptanalysis (DC) and linear cryptanalysis (LC) are methods used to attack cryptographic algorithms, thereby, it is important to consider a secure cipher to withstand these attacks.

To apply differential cryptanalysis on an n-bit block cipher, a predictable difference has to be propagated in all (except for a few) rounds with probabilities greater than 2^{1-n}. For the LAO-3D block cipher to be immune to differential cryptanalysis, there should be no difference propagation with a probability higher than 2^{-63}. Table 16 represents the differential trails with the optimal probability for six rounds of LAO-3D that are depicted in Figure 17.

Table 14. The *p*-values of the Uniformity Test (CBC, PCC, RPRK, SKA, and SPA).

				Data Category		
	Statistical Test	CBC	PCC	RPRK	SKA	SPA
1.	Runs	0.870856	0.691081	0.061260	0.069430	0.402962
2.	Frequency	0.336111	0.530120	0.626709	0.641284	0.731886
3.	Spectral DFT	0.771469	0.765632	0.651693	0.225998	0.277082
4.	Block Frequency	0.092041	0.711601	0.116746	0.552383	0.719747
5.	Binary Matrix Rank	0.377007	0.122325	0.682823	0.944274	0.264901
6.	Approximate Entropy	0.431754	0.331408	0.881662	0.128874	0.739918
7.	Longest Runs of Ones	0.070737	0.302657	0.591409	0.771469	0.334538
8.	Serial	0.366038	0.391685	0.089581	0.624739	0.560316
9.	Cumulative Sums	0.741256	0.712763	0.082836	0.687575	0.379922
10.	Non-Overlapping Templates	0.540168	0.477106	0.538421	0.501121	0.500920
11.	Maurer's Universal	0.814724	0.164425	0.245490	0.277082	0.689019
12.	Linear Complexity	0.120207	0.471146	0.668321	0.298282	0.463512
13.	Overlapping Templates	0.102526	0.199045	0.554420	0.743915	0.228367
14.	Random Excursion	0.533205	0.399313	0.592636	0.556975	0.585012
15.	Random Excursion Variant	0.474550	0.414868	0.472390	0.577133	0.547103

Table 15. The *p*-values of the Uniformity Test (LDK, HDK, LDP, and HDP).

			Data Category		
	Statistical Test	LDK	HDK	LDP	HDP
1.	Runs	0.291091	0.177628	0.239266	0.496351
2.	Frequency	0.114712	0.566688	0.773405	0.639202
3.	Spectral DFT	0.274341	0.914025	0.684890	0.943242
4.	Block Frequency	0.500279	0.769527	0.361938	0.118812
5.	Binary Matrix Rank	0.370262	0.711601	0.001953	0.118120
6.	Approximate Entropy	0.889118	0.316052	0.072066	0.009535
7.	Longest Runs of Ones	0.757790	0.301194	0.749884	0.821937
8.	Serial	0.226422	0.576050	0.262395	0.804706
9.	Cumulative Sums	0.141130	0.181735	0.660295	0.418857
10.	Non-Overlapping Templates	0.503304	0.539874	0.425798	0.418952
11.	Maurer's Universal	0.570792	0.471146	*	*
12.	Linear Complexity	*	*	*	*
13.	Overlapping Templates	*	*	*	*
14.	Random Excursion	*	*	*	*
15.	Random Excursion Variant	*	*	*	*

* no test executed due to insufficient data length.

Table 16. Differential Trails with Optimal Probability.

Round	Input Difference of S-box	Output Difference of S-box	Probability	Probability of Differential Trails
1	0000000000000001	0000000000000003	2^{-2}	2^{-2}
2	0000000000000088	0000000000000039	2^{-6}	2^{-8}
3	000000004000C008	0000000050005003	2^{-7}	2^{-15}
4	0001200000090088	0003500000040099	2^{-12}	2^{-27}
5	500080820000E008	1000303500002009	2^{-17}	2^{-44}
6	0000890100229108	0000340300334303	2^{-20}	2^{-64}

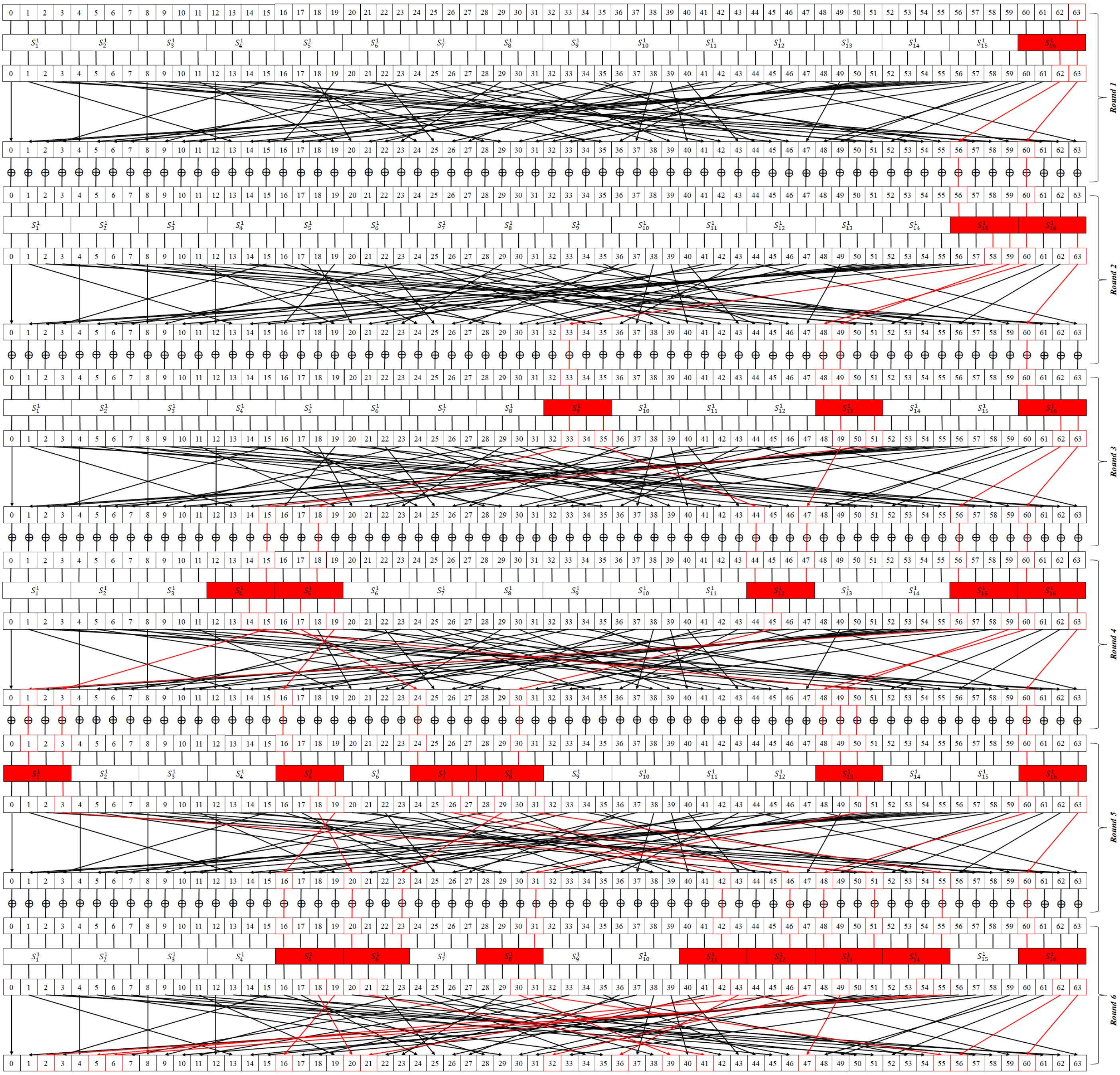


Figure 17. The 6-round Differential Characteristics of LAO-3D.

The highest probability of difference propagation found in the sixth round is 2^{-64}, which is lower than 2^{-63}. Differential cryptanalysis results of lightweight block ciphers are shown in Table 17. An effective differential distinguisher with more than five rounds for the algorithm was impossible to be constructed; the 20-round LAO-3D is sufficient to resist differential cryptanalysis.

Table 17. Probabilities of Differential Trails (DC) and Correlation Potentials of Linear Trails (LC).

	Algorithm							
	LAO-3D		**GIFT**		**PRESENT**		**RECTANGLE**	
	Attack							
Round	**DC**	**LC**	**DC**	**LC**	**DC**	**LC**	**DC**	**LC**
1	2^{-2}	2^{-2}	2^{-6}	2^{-1}	2^{-2}	2^{-1}	2^{-2}	2^{-1}
2	2^{-8}	2^{-8}	2^{-10}	2^{-2}	2^{-4}	2^{-2}	2^{-4}	2^{-2}
3	2^{-15}	2^{-14}	2^{-16}	2^{-3}	2^{-8}	2^{-4}	2^{-7}	2^{-4}
4	2^{-27}	2^{-20}	2^{-20}	2^{-5}	2^{-12}	2^{-6}	2^{-10}	2^{-6}
5	2^{-44}	2^{-26}	2^{-26}	2^{-7}	2^{-20}	2^{-8}	2^{-14}	2^{-8}
6	* 2^{-64}	2^{-32}	2^{-30}	2^{-10}	2^{-24}	2^{-10}	2^{-18}	2^{-10}
7		* 2^{-38}	2^{-36}	2^{-13}	2^{-28}	2^{-12}	2^{-25}	2^{-13}
8			2^{-40}	2^{-16}	2^{-32}	2^{-14}	2^{-31}	2^{-16}
9			2^{-46}	2^{-20}	2^{-36}	2^{-16}	2^{-36}	2^{-19}
10			2^{-50}	2^{-25}	2^{-41}	2^{-18}	2^{-41}	2^{-22}
11			2^{-56}	2^{-29}	2^{-46}	2^{-20}	2^{-46}	2^{-25}
12			2^{-60}	2^{-31}	2^{-52}	2^{-22}	2^{-51}	2^{-28}
13			* 2^{-64}	* 2^{-34}	2^{-56}	2^{-24}	2^{-56}	2^{-31}
14					2^{-62}	2^{-26}	2^{-61}	* 2^{-34}
15					* 2^{-66}	2^{-28}	* 2^{-66}	
16						2^{-30}		
17						2^{-32}		
18						*2^{-34}		

* No effective trail from the encryption round onward.

To apply linear cryptanalysis on an n-bit block cipher, there should be a predictable linear propagation overall but a few rounds with a significant amplitude greater than $2^{-\frac{n}{2}}$. For the LAO-3D block cipher to be immune to linear cryptanalysis, there should be no linear propagation with an amplitude greater than 2^{-32}. Table 18 represents the linear trails with the optimal bias for seven rounds of LAO-3D that are depicted in Figure 18.

Table 18. Linear Trails with Optimal Bias.

Round	**Input Mask of S-box**	**Output Mask of S-box**	**Bias**	**Correlation Potentials of Linear Trails**
1	0000000000000001	0000000000000005	2^{-2}	2^{-2}
2	0000000000000808	0000000000000202	2^{-6}	2^{-8}
3	0200000000000080	0200000000000020	2^{-6}	2^{-14}
4	0000000040000020	0000000020000020	2^{-6}	2^{-20}
5	0000000040000400	0000000020000200	2^{-6}	2^{-26}
6	0200000000000400	0200000000000200	2^{-6}	2^{-32}
7	0200000000000020	0200000000000020	2^{-6}	2^{-38}

The highest probability of linear propagation at the seventh round is 2^{-38}, which is lower than 2^{-32}. Linear cryptanalysis results of block ciphers are shown in Table 17. The clustering of linear trails of LAO-3D is limited and it is impossible to construct an effective linear propagation over six rounds. Hence, the 20-round LAO-3D is sufficient to resist linear cryptanalysis.

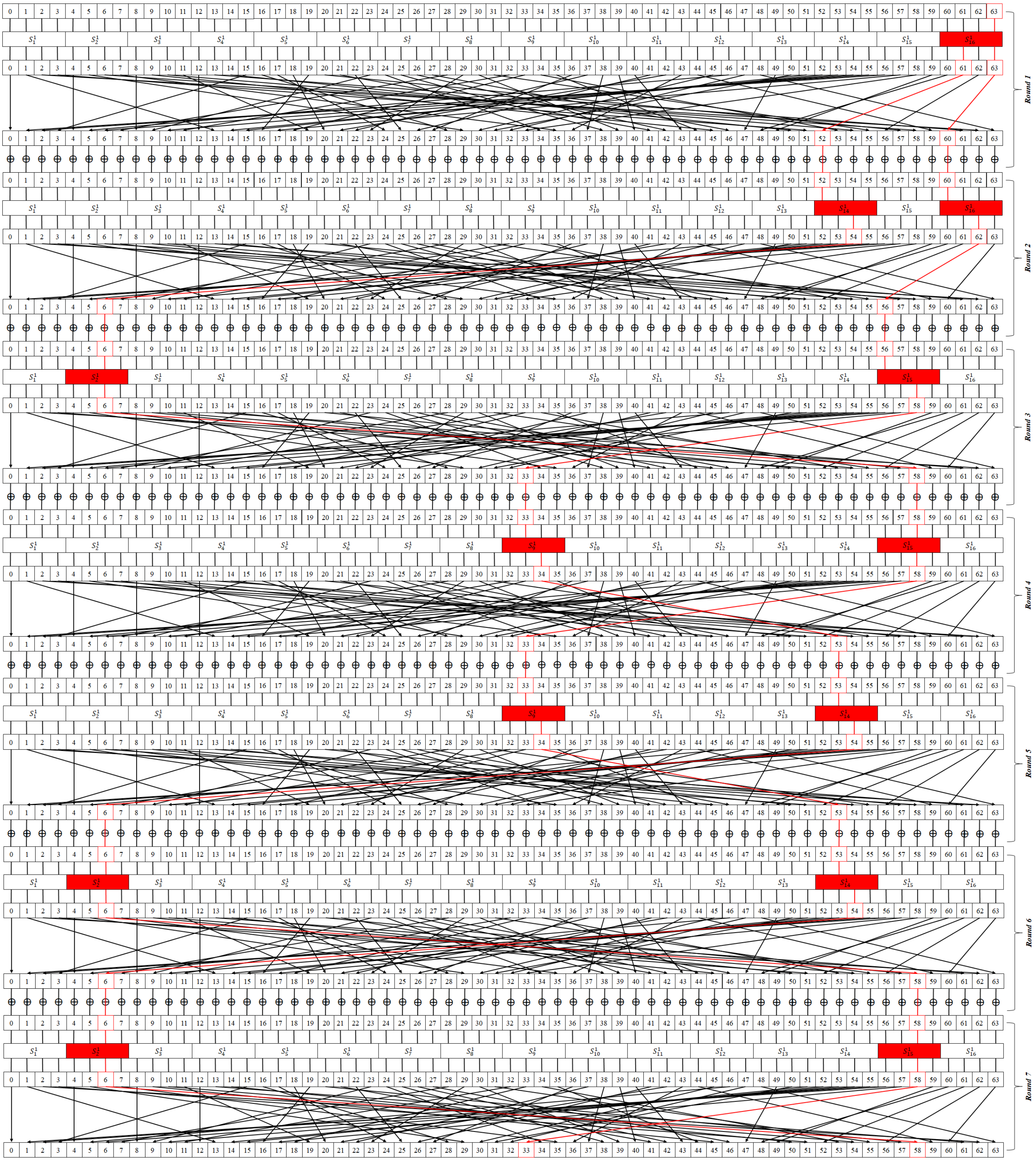


Figure 18. 7-Round Linear Characteristic of LAO-3D.

For a comparison with other existing lightweight algorithms, Table 19 shows the number of active differential and linear S-boxes of seven encryption rounds. The results show that the LAO-3D block cipher has higher active S-boxes from the second round onward. Referring to Table 17 [35,36] and Table 19 [37], LAO-3D recorded the lowest allowable probabilities of differential trails (2^{-44}) with 17 active S-boxes at the fifth round. In addition, LAO-3D achieved the lowest allowable correlation potentials of linear trails

(2^{-32}) at the sixth round with 11 active S-boxes. Therefore, LAO-3D has a high-security strength and is resistant to differential and linear attacks.

Table 19. Active S-Boxes of Differential Cryptanalysis (DC) and Linear Cryptanalysis (LC).

		Round								
Algorithm	**Attack**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
LAO-3D	DC	1	3	6	11	17	* 25			
	LC	1	3	5	7	9	11	* 13		
GIFT	DC	1	2	3	5	7	10	13	16	18
	LC	1	2	3	5	7	9	12	15	18
PRESENT	DC	1	2	4	6	10	12	14	16	18
	LC	1	2	3	4	5	6	7	8	9
RECTANGLE	DC	1	2	3	4	6	8	11	13	14
	LC	1	2	3	4	6	8	10	12	14

* no effective trail from the encryption round onward.

4.6. Performance Test

The proposed LAO-3D was implemented using C++ on an Intel(R) Core(TM) i7 2.70 GHz CPU with 8 GB RAM on Windows 10. Speed tests were conducted on the execution process of the proposed LAO-3D and compared with other existing algorithms. The speed tests were carried out on the full rounds of each encryption algorithm to observe the time required to process a ciphertext block that consisted of 64-bit data. In addition, the throughput tests evaluate the impact of cipher design such as the key size, block size, number of rounds, and encryption components on the algorithm throughput.

The performance of a cryptographic algorithm is determined by evaluating the running speed that can be measured by the average encryption time, encryption throughput, and the required number of cycles to encrypt one byte or block plaintext, which permits researchers to compare the running speeds of different algorithms working on different platforms. The encryption throughput and the number of cycles are defined in the following equations.

$$Encryption\ Throughput = \frac{\text{Message Size}}{\text{Encryption Speed}} \tag{10}$$

$$Cycles\ per\ Byte = \frac{\text{CPU Clock Speed}}{\text{Encryption Throughput}} \tag{11}$$

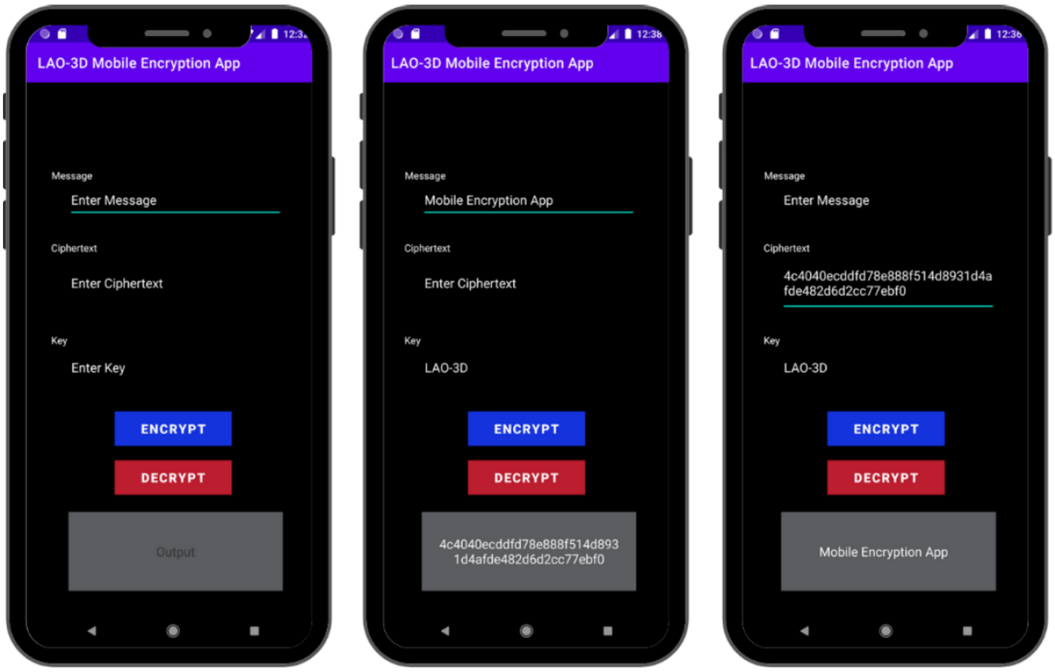
Table 20 shows that the LAO-3D block cipher achieved a better performance against the other competitors in terms of execution speed and throughput [38]. Factors that contribute to the results are the number of encryption rounds and the components of the algorithm. LAO-3D has low encryption rounds with optimum component operations. Substitution (SubColumn) and permutation (Double3DRotation) functions of the proposed block cipher ensure security without the need for implementing a high number of encryption rounds. These findings are supported by the security analysis results presented in previous sections. Therefore, these observations justify that the performance of the proposed algorithm is competitive and suitable to be applied in mobile encryption applications.

Table 20. Performance Test Results.

Algorithm	LAO-3D	KATAN	KLEIN	PRESENT	SPECK
Block Size (bit)	64	64	64	64	64
Key Size (bit)	128	80	64	128	128
Round	20	254	12	31	27
Encryption Algorithm Component	1. Add Round Key 2. Sub Column 3. 3D Rotation	1. LFSR	1. Sub Nibble 2. Rotate Nibble 3. Mix Nibble	1. Add Round Key 2. Substitution 3. Permutation	1. XOR 2. Modulo Addition 3. Rotation
Encryption Speed (millisecond)	1.4569	2.5498	2.0712	5.5895	4.6377
Encryption Throughput (byte per second)	5,491.25	3,137.50	3,862.50	1,431.25	1,725.00
Encryption Throughput (block per second)	686.41	392.19	482.81	178.91	215.63
Cycles per Byte	491,691	860,558	699,029	1,886,463	1,565,217
Cycles per Block	3,933,531	6,884,462	5,592,233	15,091,703	12,521,739

4.7. Mobile Encryption Application

This section presents the implementation of the LAO-3D lightweight block cipher on the mobile encryption application that was built using Android Studio (Bumblebee 2021.1.1) development software. Three steps are required to use the application—entering the encryption key, entering the message, and executing the data encryption in the mobile encryption application, as shown in Figure 19. Meanwhile, the decryption process requires entering the decryption key, entering the ciphertext, and executing the data decryption. Apart from the security analysis and performance test presented in Section 4, the development of the mobile encryption application proved the functionality of the proposed LAO-3D lightweight block cipher in a real application.



LAO-3D Mobile Encryption App

Message: Enter Message

Ciphertext: Enter Ciphertext

Key: Enter Key

Output

LAO-3D Mobile Encryption App

Message: Mobile Encryption App

Ciphertext: Enter Ciphertext

Key: LAO-3D

4c4040ecddfd78e888f514d893
1d4afde482d6d2cc77ebf0

LAO-3D Mobile Encryption App

Message: Enter Message

Ciphertext: 4c4040ecddfd78e888f514d8931d4a
fde482d6d2cc77ebf0

Key: LAO-3D

Mobile Encryption App

(i) Homepage (ii) Encryption (iii) Decryption

Figure 19. LAO-3D Mobile Encryption Application.

5. Conclusions

In conclusion, the proposed 3D light algorithm operation (LAO-3D) achieves confusion and diffusion characteristics with the implementation of a 3D design. Five security analyses were conducted—the correlation coefficient, bit error, key sensitivity, randomness tests, and cryptanalysis—to prove the security strength of the algorithm. From the findings of the research, the following conclusions are drawn. LAO-3D block cipher surpasses the security capabilities of existing algorithms and is immune to differential and linear attacks. The security provided by the proposed algorithm is an essential feature for mobile encryption applications. Apart from that, the performance test results justify the efficiency of LAO-3D, which has been verified with its implementation in a mobile encryption application. For future research, it is suggested to implement LAO-3D block cipher on hardware devices to observe its implementation performance.

Author Contributions: Conceptualization, A.A.Z. and A.H.A.H.; methodology, A.A.Z. and A.H.A.H.; software, A.A.Z.; validation, A.H.A.H., F.R. and N.H.Z.; formal analysis, A.A.Z.; investigation, F.R. and N.H.Z.; resources, A.H.A.H.; data curation, A.A.Z.; writing—original draft preparation, A.A.Z.; writing—review and editing, F.R. and M.D.; visualization, N.H.Z. and M.D.; supervision, A.H.A.H., F.R., N.H.Z. and M.D.; project administration, A.H.A.H. and F.R.; funding acquisition, A.H.A.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by the Ministry of Higher Education (MOHE) Malaysia under research grant FRGS/1/2019/ICT03/USIM/02/1. The authors would like to express their gratitude to Universiti Sains Islam Malaysia (USIM), MOHE, and CyberSecurity Malaysia (CSM) for the support and facilities provided.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank all reviewers for their helpful comments.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. O'Dea, S. Forecast Number of Mobile Users Worldwide 2020–2025. Available online: https://www.statista.com/statistics/218984/number-of-global-mobile-users-since-2010/ (accessed on 15 March 2022).
2. Daud, M.; Rasiah, R.; George, M.; Asirvatham, D.; Thangiah, G. Bridging the gap between organisational practices and cyber security compliance: Can cooperation promote compliance in organisations? *Int. J. Bus. Soc.* **2018**, *19*, 161–180.
3. Dhanda, S.S.; Singh, B.; Jindal, P. Lightweight cryptography: A solution to secure IoT. *Wirel. Pers. Commun.* **2020**, *112*, 1947–1980. [CrossRef]
4. Chew, L.C.N; Ismail, E.S. S-box construction based on linear fractional transformation and permutation function. *Symmetry* **2020**, *12*, 826. [CrossRef]
5. Rouaf, M.T.; Yousif, A. Design and implementation of a lightweight encryption scheme for wireless sensor nodes. In *International Conference on Computer, Control, Electrical, and Electronics Engineering*; Springer: Cham, Switzerland, 2021; pp. 1–5.
6. Salunke, R.; Bansod, G.; Naidu, P. Design and implementation of a lightweight encryption scheme for wireless sensor nodes. In *Advances in Intelligent Systems and Computing*; Springer: Cham, Switzerland, 2019; pp. 566–581.
7. Sehrawat, D.; Gill, N.S. BRIGHT: A small and fast lightweight block cipher for 32-bit processor. *Int. J. Eng. Adv. Technol.* **2019**, *8*, 1549–1556.
8. Beierle, C.; Leander, G.; Moradi, A.; Rasoolzadeh, S. CRAFT: Lightweight tweakable block cipher with efficient protection against DFA attacks. *IACR Trans. Symmetric Cryptol.* **2019**, *1*, 5–45. [CrossRef]
9. Jithendra, K.B.; Kassim, S.T. ACT: An ultra-light weight block cipher for Internet of Things. *Int. J. Comput. Digit. Syst.* **2020**, *9*, 921–929.
10. Biswas, A.; Majumdar, A.; Nath, S.; Dutta, A.; Baishnab, K.L. LRBC: A lightweight block cipher design for resource constrained IoT devices. *J. Ambient. Intell. Humaniz. Comput.* **2020**, 1–15. [CrossRef]
11. Toprak, S.; Akbulut, A.; Aydın, M.A.; Zaim, A.H. LWE: An energy-efficient lightweight encryption algorithm for medical sensors and IoT devices. *Electrica* **2020**, *20*, 71–81. [CrossRef]
12. Girija, M.; Manickam, P.; Ramaswami, M. PriPresent: An embedded prime lightweight block cipher for smart devices. *Peer-to-Peer Netw. Appl.* **2020**, *14*, 1–11. [CrossRef]
13. Sakamoto, K.; Minematsu, K.; Shibata, N.; Shigeri, M.; Kubo, H.; Funabiki, Y.; Bogdanov, A.; Morioka, S.; Isobe, T. Tweakable TWINE: Building a tweakable block cipher on generalized feistel structure. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* **2020**, *103*, 1629–1639. [CrossRef]

14. Ramadan, R.A.; Aboshosha, B.W.; Yadav, K.; Alseadoon, I.M.; Kashout, M.J.; Elhoseny, M. LBC-IoT: Lightweight block cipher for IoT constraint devices. *Comput. Mater. Contin.* **2021**, *67*, 3563–3579. [CrossRef]
15. Chen, B.W.; Xia, X.; Liang, Q.M.; Zhong, W.D. Lightweight design of SM4 algorithm and realization of threshold scheme. *J. Phys. Conf. Ser.* **2021**, *1871*, 012124 [CrossRef]
16. Nawaz, Y.; Wang, L. Block cipher in the ideal cipher model: A dedicated permutation modeled as a black-box public random permutation. *Symmetry* **2019**, *11*, 1485. [CrossRef]
17. Sakalauskas, E.; Dindienė, L.; Kilčiauskas, A.; Lukšys, K. Perfectly secure Shannon cipher construction based on the matrix power function. *Symmetry* **2020**, *12*, 860. [CrossRef]
18. Nakahara, J. 3D: A three-dimensional block cipher. In *Proceedings of the International Conference on Cryptology and Network Security*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 252–267.
19. Suri, P.R.; Deora, S.S. 3D array block rotation cipher: An improvement using lateral shift. *Glob. J. Comput. Sci. Technol.* **2011**, *11*, 17–23.
20. Ariffin, S.; Mahmod, R.; Jaafar, A.; Ariffin, M.R.K. Immune systems approaches for cryptographic algorithm. In Proceedings of the International Conference on Bio-Inspired Computing: Theories and Applications, Penang, Malaysia, 27–29 September 2011; pp. 231–235.
21. Mala, H. Unified byte permutations for the block cipher 3D. *J. Comput. Secur.* **2014**, *1*, 15–22.
22. Wang, Q.; Jin, C. A non-alternate 3D structure and its practical security evaluation against differential and linear cryptanalysis. *Sci. China Inf. Sci.* **2018**, *61*, 1–3. [CrossRef]
23. Mushtaq, M.F.; Jamel, S.; Megat, S.R.B.; Akram, U.; Deris, M.M. Key schedule algorithm using 3-dimensional hybrid cubes for block cipher. *Int. J. Adv. Comput. Sci. Appl.* **2019**, *10*, 427–442. [CrossRef]
24. Zakaria, A.A.; Azni, A.H.; Ridzuan, F.; Zakaria, N.H.; Daud, M. Extended RECTANGLE algorithm using 3D bit rotation to propose a new lightweight block cipher for IoT. *IEEE Access* **2020**, *8*, 198646–198658. [CrossRef]
25. Bogdanov, A.; Knudsen, L.R.; Leander, G.; Paar, C.; Poschmann, A.; Robshaw, M.J.B.; Seurin, Y.; Vikkelsoe, C. PRESENT: An ultra-lightweight block cipher. In *Cryptographic Hardware and Embedded Systems-CHES 2007*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 450–466.
26. Saha, S.; Jap, D.; Roy, D.B.; Chakraborty, A.; Bhasin, S.; Mukhopadhyay, D. A framework to counter statistical ineffective fault analysis of block ciphers using domain transformation and error correction. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 1905–1919. [CrossRef]
27. Shannon, C.E. Communication theory of secrecy systems. *Bell Syst. Tech. J.* **1949**, *28*, 656–715. [CrossRef]
28. Imdad, M.; Ramli, S.N.; Mahdin, H. An enhanced key schedule algorithm of PRESENT-128 block cipher for random and non-random secret keys. *Symmetry* **2022**, *14*, 604. [CrossRef]
29. Abikoye, O.C.; Haruna, A.D.; Abubakar, A.; Akande, N.O.; Asani, E.O. Modified advanced encryption standard algorithm for information security. *Symmetry* **2019**, *11*, 1484. [CrossRef]
30. Zakaria, A.A.; Azni, A.H.; Ridzuan, F.; Zakaria, N.H.; Daud, M. Modifications of key schedule algorithm on RECTANGLE block cipher. In *International Conference on Advances in Cyber Security*; Springer: Singapore, 2020; pp. 194–206.
31. Rukhin, A.; Soto, J.; Nechvatal, J.; Smid, M.; Barker, E.; Leigh, S.; Levenson, M.; Vangel, M.; Banks, D.; Heckert, A.; et al. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications. NIST Special Publication 800-22 Revision 1a. Available online: http://csrc.nist.gov/groups/ST/toolkit/rng/documents/SP800-22rev1a.pdf (accessed on 8 February 2022).
32. Abdullah, N.A.N.; Chew, L.C.N.; Zakaria, A.A.; Seman, K.; Norwawi, N.M. The comparative study of randomness analysis between modified version of LBlock block cipher and its original design. *Int. J. Comput. Inf. Technol.* **2015**, *4*, 867–875.
33. Zakaria, A.A.; Azni, A.H.; Ridzuan, F.; Zakaria, N.H.; Daud, M. Randomness analysis on RECTANGLE block cipher. Proceedings of the 7th International Cryptology and Information Security Conference, Kuala Lumpur, Malaysia, 9–10 June 2020; pp. 133–142.
34. Preishuber, M.; Hutter, T.; Katzenbeisser, S.; Uhl, A. Depreciating motivation and empirical security analysis of chaos-based image and video encryption. *IEEE Trans. Inf. Forensics Secur.* **2018**, *13*, 2137–2150. [CrossRef]
35. Zhu, B.; Dong, X.; Yu, H. MILP-based differential attack on round-reduced GIFT. In *Cryptographers' Track at the RSA Conference*; Springer: Cham, Switzerland, 2019; pp. 372–390.
36. Zhou, C.; Zhang, W.; Ding, T.; Xiang, Z. Improving the MILP-based security evaluation algorithm against differential/linear cryptanalysis using a divide-and-conquer approach. *IACR Trans. Symmetric Cryptol.* **2019**, *4*, 438–469. [CrossRef]
37. Banik, S.; Pandey, S.K.; Peyrin, T.; Sim, S.M.; Todo, Y.; Sasaki, Y. GIFT: A Small Present. In *International Conference on Cryptographic Hardware and Embedded Systems*; Springer: Cham, Switzerland, 2017; pp. 321–345.
38. Singh, P.; Acharya, B.; Chaurasiya, R.K. A comparative survey on lightweight block ciphers for resource constrained applications. *Int. J. High Perform. Syst. Archit.* **2019**, *8*, 250–270. [CrossRef]