KARYA ILHAM PUBLISHING
CONNECTING MINDS, BRIDGING IDEAS, INSPIRING READERS

International Journal of Advanced Research in Computational Thinking and Data Science

Journal homepage:
https://karyailham.com.my/index.php/ctds/index
ISSN: 3030-5225

3D Lightweight Cryptosystem Design for IoT Applications Based on Composite S-Box

Tasnuva Ali[1,3,*], Azni Haslizan Ab Halim[1,2], Nur Hafiza Zakaria[1,2]

[1] Faculty of Science and Technology, Universiti Sains Islam Malaysia, 71800, Bandar Baru, Nilai, Negeri Sembilan, Malaysia
[2] CyberSecurity and Systems (CSS) Research Unit, Faculty of Science and Technology, Universiti Sains Islam Malaysia, 71800, Bandar Baru Nilai, Negeri Sembilan, Malaysia
[3] Daffodil University, Dhaka, Bangladesh

ARTICLE INFO

Article history:
Received 16 June 2024
Received in revised form 26 August 2024
Accepted 4 September 2024
Available online 15 September 2024

Keywords:
Rectangle; IoT; S-Box; 3D; Security

ABSTRACT

The security of the Internet of Things (IoT) depends on strong cryptographic functions that require extensive computation and resources. The security of Internet of Things (IoT) relies on the use of strong cryptographic functions that demand extensive computation and resources. Therefore, the selection of a cryptographic system is influenced by the computational and communicational capabilities of IoT devices, including their energy requirements, memory constraints, and execution time. This paper aims to develop a lightweight and secure encryption algorithm for IoT applications, focusing on advancing cryptanalysis techniques. We propose a new lightweight algorithm named enhanced 3D RECTANGLE, designed to deliver robust security for IoT applications and optimized for cell phones with minimal memory usage, low power consumption, and efficient performance. The RECTANGLE block cipher was chosen for this research due to its high efficiency and speed relative to other lightweight algorithms, despite some security issues. The proposed enhanced 3D RECTANGLE algorithm improves confusion and diffusion properties through a new 3D array block rotation method for 4x4 plaintext, based on a 128-bit key and 16 rounds. Additionally, we designed a 4x4 composite S-Box with a Galois field pipelining structure, offering a more advanced solution compared to the Look-Up Table method. The cryptanalysis, avalanche effect, and bit error rate tests were conducted to verify the security strength of the proposed algorithm. The proposed algorithm was evaluated against two existing algorithms, RECTANGLE and Extended 3D RECTANGLE. The enhanced 3D RECTANGLE algorithm provides better cryptanalysis, demonstrating a 40% avalanche effect and achieving a bit error rate (BER) of 31%, indicating its greater effectiveness compared to the other two lightweight algorithms.

1. Introduction

A broad range of networking and mobile communication systems have become essential in modern society and interact with the world around us. Their ongoing development and integration into various aspects of daily life are expected to change future society [1]. The Internet of Things (IoT)

* *Corresponding author.*
E-mail address: tasnuva@daffodilvarsity.edu.bd

https://doi.org/10.37934/ctds.3.1.4054a

is a broad range of applications enabled by connecting devices such as sensors, actuators, and monitors accessible through the Internet and cell phones. Cryptography provides the foundation for secure communication systems by safeguarding data confidentiality, integrity, authenticity, and non-repudiation [2]. The traditional cryptography solutions focus on providing high levels of security but pay no attention to the constrained device requirements. Moreover, the explosion of Internet of Things (IoT) devices will make cryptography even more crucial. Therefore, robust and efficient cryptography solutions are essential for the upcoming years to safeguard this ever-expanding network.

Lightweight cryptography (LWC) is a research field that focuses on designing schemes for devices with constrained capabilities in power supply, connectivity, hardware, and software. The wide-ranging IoT applications are becoming potential and popular because of their easy data collection process from the real world and their ability to transfer data in different domains. The most common challenge of handling IoT devices is limited resources such as memory, energy, power, and even physical [3]. Therefore, resource-constrained IoT devices are facing high-security issues for transferring data over the network. Moreover, the embedded IoT devices are located in various places, but they are vulnerable to physical security [4]. Despite their limited memory, energy, and power capabilities, it is essential to improve their protection compared to existing designs. The lightweight algorithms are suitable for power-constrained devices that provide definite solutions to exact problems by designers [5]. These algorithms are designed with two primary objectives. The first step is to identify all potential cryptanalysis attacks, which is essential for assessing the security of cryptographic systems. The next phase is to develop a set of specific constraints that are applied on a case-by-case basis [6]. The difficult task is to merge these two phases to get a better performance of any lightweight algorithms. Moreover, many lightweight encryption algorithms use a symmetric key for the encryption process which consists of several encryption rounds [7]. Each encryption round is based on some mathematical functions to create confusion and diffusion. Therefore, the lightweight cipher provides good security like other ciphers for more rounds but eventually increases the consumption of constrained energy which also degrades the performance of the cipher.

Many lightweight block ciphers have been designed over the last two decades to meet the limitations. Many of these designs satisfied one or two targets but failed to give security properly. Moreover, many articles have worked on enhancing the security of proposed algorithms but could not meet all security criteria recommended by NIST. In this paper, different lightweight algorithms are studied thoroughly by addressing the secured transmission of signals in the medium. Among them, SIMECK [8], SPECK [9], GIFT [10], Midori [11], RoadRunneR [12], LRBC [13], SKINNY [14], RECTANGLE [15], BORON [16], Loong [17] are the popular SPN and Feistel networks lightweight algorithms to meet the different performance criteria.

The RECTANGLE algorithm has been chosen due to its better performance in the Figure of Merit (FOM) and security features for this work [18]. Despite its resistance to linear and differential attacks for a minimum of 25 rounds, the algorithm is not secure when the number of rounds is less than 25 [19]. Moreover, the RECTANGLE cipher is vulnerable to different types of attacks, such as statistical saturation attacks, related-key attacks, and slide attacks. These vulnerabilities need to be addressed for future improvements. The effective approach to enhance the security of 2D rectangles is by optimizing the diffusion and confusion steps or round numbers, which can increase the delay time. Therefore, designing an algorithm that minimizes the mentioned attacks is essential while optimizing the confusion and diffusion steps [20]. To improve the confusion property of RECTANGLE, different types of ideal S-Boxes can be designed [21] and the poor key schedule algorithm can be upgraded to improve the diffusion properties [22]. Therefore, an enhanced 3D RECTANGLE cipher should be designed to overcome its limitations for better IoT applications.

In this paper, we have proposed a composite S-Box based on 3D block cipher rotation for the RECTANGLE algorithm. The confusion/substitution step is done by designing a composite S-Box. The diffusion step, equal to shift rows and the mix column is ended by 3D rotation to get the final cipher text. The composite S-Box is designed for 4X4 block size using a Galois field sub-pipeline structure which reduces the number of gates as areas compared to other 8X8 composite S-Boxes [23-25]. Moreover, the shift rows and mix column operation are achieved by 3D rotation which uses 64-bit block size and 128-bit key with 16 iterations. Therefore, the proposed design for only 16 rounds gives improved cryptanalysis, BER, and avalanche effect, compared to other 3D lightweight algorithms.

2. 3D Technique for Cryptosystem

The concept of 3D block cipher involves encrypting data using a three-dimensional approach, which often enhances the complexity and security of traditional block ciphers. The 3D block cipher developed by Nakahara in 2008 implements a block length and key length of 512 bits. These lengths are structured as a 4 x 4 x 4-byte state representation [26]. The cipher applies a 4 × 4 matrix to each column of every vertical slice of the state during encryption. Despite advancements in the security analysis of the block cipher, it requires 22 rounds of iteration in the transformation function, increasing the number of rounds by 57 percent compared to the AES block cipher. However, 3D takes advantage of the three-dimensional states and applies the byte permutation of AES (Shift Rows) in two directions for every two rounds alternately. The National Institute of Standards and Technology (NIST) evaluated block ciphers based on several criteria, including security, efficiency in software and hardware implementations, and adaptability to different applications and requirements. The 3D cipher seems to be secured enough against known block cipher cryptanalysis techniques. The most significant cryptanalytic results for this cipher include a 10-round impossible differential attack [27].

Suri and Deora proposed a 2D rotate block transformation as an alternative to the four-step AES block cipher. It involves five transformations where the entire block is rotated in a clockwise direction by 0, 90, 180, or 270 degrees based on the 2 bits of the key value. The algorithm proposed adding some new steps to enhance prediction accuracy in a single step for brute-force attacks [28]. However, in practice, this approach leads to an increase in energy consumption and overall cost rather than achieving the intended improvement in accuracy. Moreover, Suri and Deora presented an innovative method for encrypting data using 3D array rotations. This technique employs a three-dimensional 3 × 3 × 3 matrix to securely store the initial plaintext and has good randomness properties. However, the algorithm has a significant weakness in its diffusion properties, as the middle matrix remains unchanged during each round of iteration. The encryption method also involves the execution of a significant number of encryption rounds, comprising 64 iterations, and utilizes a large key size that is eight times multiplied by its round number. Furthermore, the testing only covered four out of the 15 statistical tests from the NIST test suite, which raises concerns about the algorithm's overall standard and reliability [29]. However, the 3D structure may increase the block cipher algorithm's strength, but it may reduce its efficiency due to the increased number of block sizes, key sizes, and encryption rounds.

In subsequent work, immune-inspired approaches were used to construct a 3D-AES that passed the statistical tests [30]. This immune-inspired method increased the AES block size from 256 to 512 bits. Later on, a non-alternative 3D structure with 512-bit blocks was proposed by Wang that secured against linear and differential cryptanalysis [31]. Recently, Mushtaq proposed a key schedule algorithm using 3D hybrid cubes, which do not correlate between input and output data [32].

Due to their complexity, the 3D ciphers are more resistant to various attacks such as side-channel and algebraic attacks. Also, modern hardware with parallel processing capabilities can make

encryption and decryption processes more efficient by using the 3D structure. Therefore, 3D ciphers provide better security, flexibility, and performance than traditional 2D ciphers due to their advanced structure. All previous articles have suggested different 3D cipher design algorithm techniques, which aim to enhance security with large key sizes. However, they face a significant challenge when deployed on low-memory IoT devices.

Therefore, the main focus of the paper is to develop a lightweight 3D cryptosystem to ensure the security of transmitted data in an efficient IoT system. Moreover, the advanced cryptosystem's security will be compared to existing algorithms to demonstrate better performance against attacks with fewer rounds, making it applicable to future IoT applications.

3. 3D Cryptosystem Design

Previous work on 3D block cipher designs has demonstrated better randomness properties but requires 64 rounds and a large key size [33]. Consequently, the focus of this research is to achieve better confusion and diffusion properties with fewer rounds. The proposed 3D bit rotation method uses an 8-bit subkey in each round. Previous techniques for 3D rotation required a key size of at least 512 bits, which was achieved through different methods such as slice rotation [20], specific axis rotation [21], or lateral shift rotation [22]. The large block and key size are not appropriate for designing IoT devices. Hence, we propose an extended 3D RECTANGLE algorithm design that incorporates axis or shift rotations of the rows with a 128-bit key and a 64-bit block size.

3.1 Enhanced 3D Lightweight Algorithm Design

In the proposed lightweight algorithm design, the 64-bit plaintext is initially converted using a composite S-Box and then stored in a 4x4x4 array organized along the X/Y/Z axes. The 128-bit key is split into 16 groups to generate 16 subkeys according to a key schedule algorithm. Each round applies an 8-bit subkey to perform diffusion through an enhanced 3D rotation that involves rotating the plates and shifting operations to enhance the cipher's security. After completing 16 rounds with 16 different subkeys, the final 3D ciphertext is generated. Figure 1 shows the extended 3D RECTANGLE Lightweight Algorithm Design.

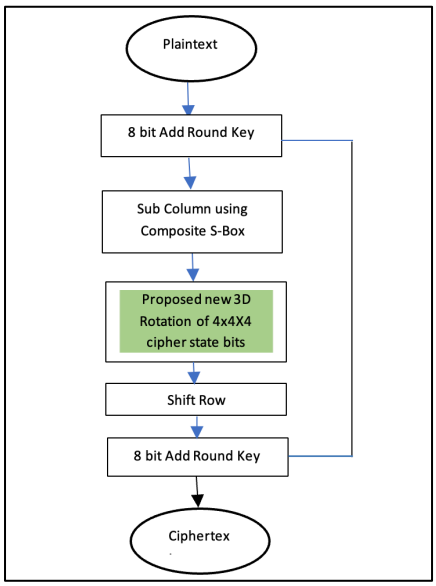

Fig. 1. Extended 3D RECTANGLE lightweight algorithm design

3.1.1 Composite S-Box design

The S-Box based on the finite fields {GF (2^4)} and GF {($(2^2)^2$)} uses a combinational and sub-pipelining structure that enhances resistance to linear and differential cryptanalysis while also achieving a reduction in the number of gates compared to traditional Look-Up Table (LUT) S-Boxes [25]. The lightweight ciphers from previous research [14-16] employ LUT-based S-Boxes, which have certain security vulnerabilities in hardware implementations. The S-Box design begins with finding the irreducible polynomials for the 4 × 4 matrix, and x^4+x+1 is selected for further S-Box calculation. We then perform an isomorphic mapping (δ) from {GF (2^4)} to GF {($(2^2)^2$)}, which involves constructing the conversion matrix. The next step is to calculate the multiplicative inverse (MI) matrix using the conversion matrix values. The inverse is then processed through the Inverse Isomorphic Mapping (δ^{-1}) to get {GF (2^4)}. Finally, the Affine Transformation (AT) is performed to complete the S-Box design for use in the diffusion process.

3.1.1.1 Composite Field Conversion Matrix (CM)

The minimal polynomial calculation for 4X4 Matrix is:

$$m_\alpha(x) = (x+\alpha)\,(x+\alpha^{2m}) \tag{1}$$

In the context of 4 × 4 matrix pipelining, where n = 2 and m = 2, the minimal polynomial simplifies to:

$$m_\alpha(x) = (x+\alpha)\,(x+\alpha^4) = x^2 + x\alpha + x\alpha^4 + \alpha^5 \tag{2}$$

We consider the irreducible polynomial $P(x) = x^4+x+1$, which leads to the relation

$$x^4 = x+1 \tag{3}$$

Using this relation, we can derive that:

$$\alpha^4 = \alpha+1 \tag{4}$$

3.1.1.2 Isomorphic function (δ)

The conversion matrix equation from {GF (2^4)} to {GF ($(2^2)^2$)} can be expressed as:

$$A= \overline{a_{00}} + (\overline{a_{10}} + \overline{a_{11}})\alpha + (\overline{a_{01}} + \overline{a_{10}} + \overline{a_{11}})\,\alpha^2 + \overline{a_{11}}\,\alpha^3 \tag{5}$$

Finally, the elements of the CM are:

$$a_0 = \overline{a_{00}} \tag{6}$$

$$a_1 = (\overline{a_{10}} + \overline{a_{11}}) \tag{7}$$

$$a_2 = \overline{a_{01}} + \overline{a_{10}} + \overline{a_{11}} \tag{8}$$

$$a_3 = \overline{a_{11}} \tag{9}$$

Here, {GF (2^4)} to GF {((2^2))^2} CM is

$$\delta \times a = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (10)$$

where "a" is the 4-bit input message.

3.1.1.3 Multiplicative Inverse (MI) calculation

The Multiplicative Inverse (MI) can be determined after performing the isomorphic function calculation. The steps to find the Multiplicative Inverse (MI) are as follows (see Figure 2):

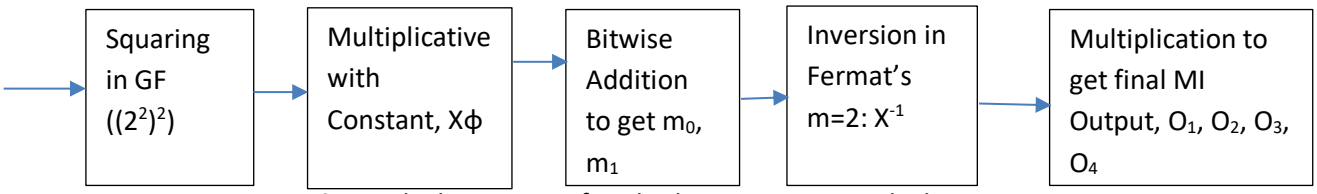

Fig. 2. Block Diagram of Multiplicative Inverse Calculation

The MI module is designed using dsch2 software to get the final O_1, O_2, O_3, O_4 output. All the add operations are considered as XOR operations. To design the Multiplicative inverse module, we need 18 XOR gates, 12 AND gates are required for 4X4 square matrix. Figure 3 shows the multiplicative Inverse Calculation Using DSCH2 Software.

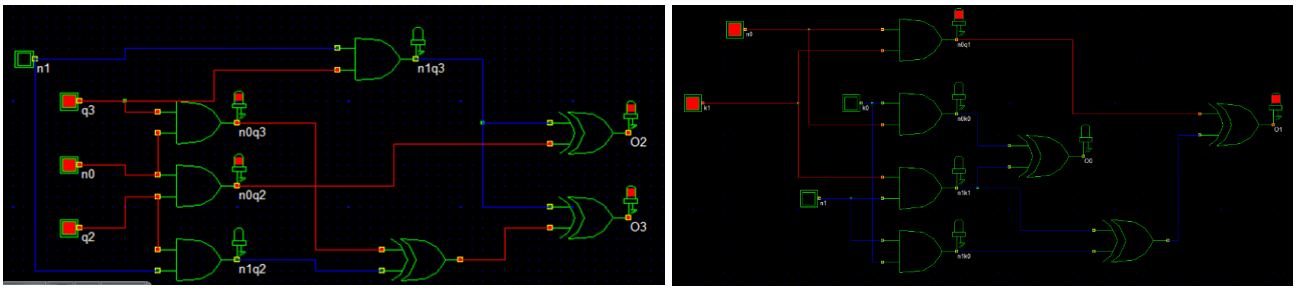

Fig. 3. Multiplicative Inverse Calculation Using DSCH2 Software

3.1.1.4 Inverse isomorphic (δ^{-1})

The inverse isomorphic (δ^{-1}) function converts GF {((22))2} to {GF (24)} using the following equation

$$A = \overline{a_{00}} + (\overline{a_{01}} + \overline{a_{10}})\alpha + (\overline{a_{01}} + \overline{a_{11}}) \alpha^2 + \overline{a_{11}} \alpha^3 \quad (11)$$

Therefore, the equation of δ^{-1} is:

$$a_0 = \overline{a_{00}} \quad (12)$$

$$a_1 = (\overline{a_{01}} + \overline{a_{10}}) \quad (13)$$

$a_2 = \overline{a_{01}} + \overline{a_{11}}$ (14)

$a_3 = \overline{a_{11}}$ (15)

Here, GF $\{((2^2))^2\}$ to GF (2^4) CM is

$$\delta^{-1} \text{ x a} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (16)$$

Therefore, the isomorphic inverse function output is applied to the Affine Transformation (AT) block to get the final Composite S-Box output.

3.1.1.5 Affine Transformation (AT)

The equation of Affine Transformation and Inverse Affine Transformation is:

AT = Ax + C (17)

AT-1 = A-1x + C (18)

A and A-1 are 4x4 Affine Matrixes, x is 4-bit inputs which come from Inverse Isomorphic Function (δ-1), '+' is XOR operation and C is the Affine Constant. Therefore,

$$\text{Affine Transformation} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{x} \begin{bmatrix} a3 \\ a2 \\ a1 \\ a0 \end{bmatrix} + \begin{bmatrix} c3 \\ c2 \\ c1 \\ c0 \end{bmatrix} \quad (19)$$

$$\text{Inverse Affine Transformation} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \text{x} \begin{bmatrix} a3 \\ a2 \\ a1 \\ a0 \end{bmatrix} + \begin{bmatrix} \overline{c3} \\ \overline{c2} \\ \overline{c1} \\ \overline{c0} \end{bmatrix} \quad (20)$$

Hence, a_0, a_1, a_2, a_3 are δ^{-1} outputs for Affine Transformation and δ for Inverse affine Transformation, c_0, c_1, c_2, c_3 are Affine Constants which can be 16 possible combinations. In this paper, we take 0110 constant to calculate further equations. We need 16 XOR gates and 16 AND gates for the Affine Transformation module. Therefore, the final output for the Composite S-Box is depicted in Table 1

Table 1
Composite S-Box Value

Inputs	0_h	1_h	2_h	3_h	4_h	5_h	6_h	7_h	8_h	9_h	A_h	B_h	C_h	D_h	E_h	F_h
Outputs	6_h	E_h	F_h	1_h	D_h	B_h	8_h	0_h	9_h	2_h	5_h	C_h	3_h	4_h	A_h	7_h

3.1.2 Axis wise plate arrangement

The proposed 3D RECTANGLE is a minimal block size-based cipher that is suitable for implementing IoT devices. The plaintext of the block cipher algorithm is based on a cube. A cube is

an array of plaintext that is designed for three independent vectors across three different axes X/Y/Z in a 4X4X4 array (see Figure 4). Hence, each plaintext value is changed using a composite S-Box table and then arrange to axis wise plate such as n-bit length (X-axis), n bits length (Y-Axis), and n bits length (Z -axis) where the block length is divisible by n^2. Each axis is divided into 4 plates. The 64 bits data block can be distributed like the figure:

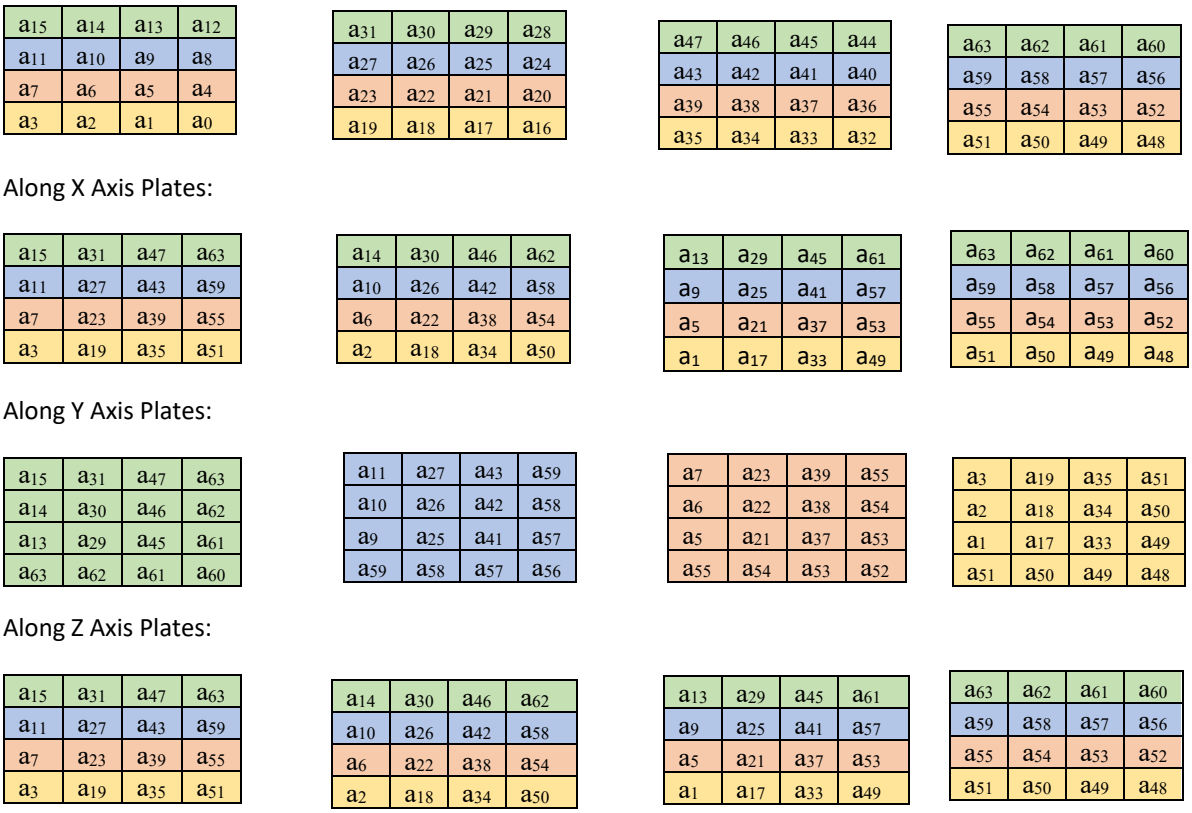

Fig. 4. Bit arrangement across X/Y/Z axis

3.1.3 Key schedule algorithm

The 3D Rectangle is designed for a 128-bit key. The 128-bit key is divided into 16 sub-keys accordingly. Each round key is 8 bits in length which is responsible for operating the rotation and diffusion to get the cipher text. The process starts by dividing the 128-bit key into 16 individual group keys, which are structured as $[W_0] = [w_7 \,||\, \ldots\ldots\ldots\ldots\ldots \,||\, w_1 \,||\, w_0]$, $[W_1] = [w_{15} \,||\, \ldots \ldots\ldots|| \, w_9 \,||\, w_8]$, and continues up to $[W_{15}] = [w_{127} \,||\, \ldots \ldots\ldots\ldots\ldots|| \, w_{121} \,||\, w_{120}]$. For each round, a 1-bit Circular Left Shift is applied to the round key before performing the substitution step using the S-Box, which generates a new key as $(K_{3,j} \,||\, K_{2,j} \,||\, K_{1,j} \,||\, K_{0,j}) = S(K_{3,j} \,||\, K_{2,j} \,||\, K_{1,j} \,||\, K_{0,j})$.

The resulting round key is then XOR-ed with the previous round key to produce the key for the next round, following the pattern $[W_1]' = [W_1] \oplus [W_0]$, and so on up to $[W_{15}]' = [W_{15}] \oplus [W_{14}]$. This process is iteratively repeated to derive a total of K_{16} round keys.

3.1.4 Key bits distribution

The diffusion of the text can be done using rotation 90, 180, or 270 degrees rotation or shifting of elements of the above plates or original plates specified by the 8-bit key to get the cipher text. Each round key is 8 bits long where 2 bits define the axis, 2 bits define the plate number, 2 bits express

types of rotation of the plate, and the last 2 bits perform shift row operation. Table 2 illustrates the key 8 bits Arrangement.

Table 2
Key 8 bits Arrangement

Shift Row (2 bits)		Rotation Degree (2 bits)		Plate Number selection (2 bits)		Axis (2 Bits)	
S_7	S_6	R_5	R_4	P_3	P_2	A_1	A_0

3.1.4 Rotation policy depending on key

Every plate of the cipher state consists of 16 bits. Hence, the four plates of X, Y, and Z axis are distributed as X (x_3, x_2, x_1, x_0), Y (y_3, y_2, y_1, y_0) and Z (z_3, z_2, z_1, z_0). Each axis consisting of four plates, is also required to accommodate all 64 bits of the RECTANGLE block size. Diffusion of the text can be done using clockwise rotations or circular left shift rotations of the particular plate in a specific axis using an 8-bit key arrangement. The leftmost 2 bits (A_1, A_0) are used for axis identification (X/Y/Z). The axis bits 00, 01, and 10 determine the X, Y, and Z axis respectively. Nevertheless, the axis bit 11 determines the X to Z shift followed by rotation bits. The 2 bits (P_3, P_2) are used for 4 plate numbers of the selected axis which need to rotate. The rotation bits 00, 01, and 10 determine 90^0, 180^0, and 270^0 rotation to the selected plate followed by the axis and plate number bits. In the case of rotation bits and axis bits, both are 11, then each row will apply a 3-bit Circular Left Shift rotation of the round cipher text. Therefore, the last two MSB bits are applied for shift rows operation where 00, 01, and 10 bits allow the selected plate for 1-bit CLS, 2-bit CLS, and 3-bit CLS rotation. In the case of shift rows bit 11, then row 0, row 1, row 2, and row 3 of the round cipher get 0-bit CLS, 1-bit CLS, 2-bit CLS, and 3-bit CLS respectively. Table 3 shows the rotation policy.

Table 3
Rotation policy

Shift Rows (2 bits)			Rotation (2 bits)			Plate Number (2 bits)			Axis (2 bits)		
0	0	1-bit CLS	0	0	90° clockwise	0	0	Plate 1	0	0	X-axis
0	1	2-bit CLS	0	1	180° clockwise	0	1	Plate 2	0	1	Y-axis
1	0	3-bit CLS	1	0	270° clockwise	1	0	Plate 3	1	0	Z-Axis
1	1	All rows CLS were row 0: no shift, row 1: 1-bit shift, row 2: 2 bit shift and row 3: 3 bit shift.	1	1	1 bit circular shift rotation determined by Axis bits (X/ Y/Z axis plates)	1	1	Plate 4	1	1	X to Z shift

3.1.5 Rotation operation

The 128-bit key of RECTANGLE is divided into 16 subkeys where 16 rounds are used to get the final cipher text. The subkeys are generated using a key schedule algorithm. After 16 rounds of operation, we will get the final cipher text of the enhanced 3D RECTANGLE algorithm.

3.1.6 Test vectors

The test vectors of the proposed algorithm are presented in Table 4 where plaintext, key, and ciphertext are given. All the input-output data of the proposed algorithm are given in the form of hexadecimals in Table 4.

Table 4
Test vectors of the proposed algorithm

Plaintext	Key	Ciphertext
0000000000000000	1283791ABCBA3456789ABC2EDFC45678	46026FFE64FD6205
1111111111111111	1283791ABCBA3456789ABC2EDFC45678	D7F2FFFEF7FF6235
FFFFFFFFFFFFFFFF	2136ABCDE321ACBC7893412ABCD90FBC	2EF5FF62B6EEBF75
321ABCDEFAAB3435	00000000000000000000000000000000	1EEDC34A755C1D93
2134567ABCDEFA23	11111111111111111111111111111111	6F84B1155ADB74E9
2134567ABCDEFA23	FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	FE1DB805C34A75F1
321ABCDEFAC56789	2134587659ABCDEFACD234567890A123	80FFBB207620AD87

4. Results

4.1 Cryptanalysis

Cryptanalysis is the process of finding algorithm weaknesses to extract plaintext from ciphertext without using the key or algorithm. There are several methods for performing cryptanalysis, depending on the cryptanalysis access to the plaintext or ciphertext. The two most important attacks against block ciphers are linear cryptanalysis and differential cryptanalysis.

4.1.1 Linear Cryptanalysis

Linear Cryptanalysis is a form of known plaintext attack where the cryptanalyst has access to a series of plaintexts and their corresponding ciphertexts. The advantage of this attack is to identify linear approximations involving plaintext bits, ciphertext bits, and subkey bits that occur with a high or low probability. The process of linear cryptanalysis includes two calculations: identifying all possible values for the Linear Approximation Table (LAT) and using these LAT values in pilling up the Limma equation to determine linear relationships. Table 5 shows the LAT Calculation.

Table 5
LAT Calculation

		LAT OUTPUT															
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
INPUT	**0**	+8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	0	0	0	0	-4	0	+4	-2	-2	-2	-2	-2	+2	-2	+2
	2	0	+4	-2	+2	0	0	-2	-2	0	0	-2	-2	0	-4	-2	+2
	3	0	0	+2	+2	0	0	+2	+2	+2	-2	-4	0	+2	-2	+4	0
	4	0	0	0	-4	-2	+2	+2	+2	0	0	0	-4	+2	-2	-2	-2
	5	0	+4	0	0	-2	+2	+2	+2	+2	-2	+2	+2	-4	0	0	0
	6	0	0	-2	+2	+2	+2	+4	0	0	+4	-2	**-2**	-2	+2	0	0
	7	0	0	+2	-2	+2	-2	0	0	-2	+2	0	0	-4	-4	+2	-2
	8	0	+2	0	-2	0	-2	+4	-2	-2	0	+2	0	+2	0	+2	+4
	9	0	+2	0	-2	-4	-2	0	-2	0	+2	-4	+2	0	+2	0	-2
	A	0	-2	-2	0	-4	+2	-2	0	-2	0	0	-2	-2	0	+4	+2
	B	0	+2	+2	0	0	-2	-2	0	+4	+2	+2	-4	0	+2	+2	0
	C	0	-2	-4	+2	-2	-4	+2	0	+2	0	+2	0	0	-2	0	-2
	D	0	+2	-4	-2	+2	0	-2	+4	0	+2	0	+2	+2	0	+2	0
	E	0	-2	+2	0	-2	0	0	+2	+2	+4	0	+2	0	-2	-2	+4
	F	0	-2	-2	-4	+2	0	0	-2	+4	-2	-2	0	-2	0	0	+2

After calculating the Linear Approximation Table (LAT) values, the next step is to compute the probability bias using LAT values and a count number for 16-bit combinations. In this case, a count of 10,000 is used for this calculation. The target bias for effective resistance against linear attacks is 1/32, or roughly 0.03125. The proposed design achieves a bias near this value, which ensures more reliable and secure data encryption.

4.1.2 Differential Cryptanalysis

One of the most impactful techniques in block cipher cryptanalysis is differential cryptanalysis, developed by Biham and Shamir [34]. This method, also known as a chosen plaintext attack, allows the cryptanalyst to choose random plaintexts for encryption and analyze the resulting ciphertexts to identify high differential probabilities. Differential cryptanalysis is a vital security evaluation technique that relies on the calculation of a distribution table (see Table 6).

Table 6
Differential Distribution Table

	β															
α	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	2	2	2	4	2	0	2	0	0	2
2	0	0	0	2	0	2	0	0	4	0	2	0	0	2	2	2
3	0	2	0	2	2	2	0	4	0	0	0	0	2	0	0	2
4	0	2	0	0	0	2	2	2	0	0	4	2	0	0	2	0
5	0	2	2	2	0	0	2	0	2	0	0	0	4	0	2	0
6	0	0	2	2	0	2	2	0	0	2	0	**2**	0	0	0	4
7	0	2	0	0	2	0	4	0	0	2	0	0	0	2	2	2
8	0	0	2	0	0	0	0	2	0	0	0	2	2	2	4	2
9	0	0	0	2	4	0	0	2	2	2	0	2	0	0	2	0
A	0	0	2	4	2	0	2	2	0	0	2	0	0	2	0	0
B	0	2	0	2	0	0	0	0	0	2	2	4	2	2	0	0
C	0	0	0	0	2	4	2	0	2	0	0	2	2	2	0	0
D	0	0	4	0	2	2	0	0	0	2	2	0	2	0	2	0
E	0	2	2	0	0	2	0	2	2	2	0	0	0	4	0	0
F	0	2	2	0	2	0	0	0	2	2	2	2	0	0	0	2

The probability of maximal differential characteristics is 2^{-2}. Therefore, the cipher differential characteristics with r= 32 is 2^{-62} which is a satisfactory level of differential cryptanalysis resistance. Hence, the proposed enhanced 3D RECTANGLE provides a significant margin of linear and differential cryptanalysis to pass the linear and differential test.

4.2 BER Analysis of The Proposed Algorithm

This statistical test measures the relationship between the plaintext and the ciphertext. The measurement will be based on the variations in the ciphertext resulting from a particular alteration in the plaintext. Bit error denotes the average number of output bits that change when a single plaintext bit is altered across varying plaintext sizes. A good result of the bit error rate should be close to 50 percent or 0.5 modification of the plaintext bits [13]. This test has been developed using Python programming, and the comparison of bit error rate results has been taken for a random key and five

random plaintext samples for the proposed enhanced 3D RECTANGLE and other testing algorithms. Table 7 shows the comparison of Key Sensitivity Test Results (All Key bits F).

Table 7
Comparison of Key Sensitivity Test Results (All Key bits F)

Plaintext	Key	Similar Bits	Proposed Average Different Bit	Proposed BER
All zero (Hex)	Random 1	668	22.25	0.34765625
All one	Random 2	701	20.1875	0.3154296875
All F	Random 3	731	18.3125	0.2861328125
Random 1	Random 4	538	30.375	0.474609375
Random 2	Random 5	539	30.3125	0.4736328125

4.3 Avalanche Effect

The avalanche effect test is one of the most important parameters to evaluate the confusion property of an algorithm. The avalanche effect deals with the dependency of each output bit on the input bits. The avalanche effect, E can be expressed as:

$$E = \frac{1}{s}\sum_{i=1}^{s}\left|C_i - P_i\right| \quad (21)$$

The key sensitivity test will be performed to examine the avalanche effect of the proposed enhanced 3D RECTANGLE algorithm. The 3D RECTANGLE algorithm was employed to convert randomly generated 64-bit plaintexts into ciphertext using 128-bit keys.

4.3.1 Key Sensitivity Test

In a cryptosystem, a small modification of the secret key results in a different ciphertext. In the key sensitivity test, the key is slightly modified by changing a number or character for every digit position from the first-place digit to the last place. The difference between the two corresponding ciphertexts as the original ciphertext and the changing ciphertext with a one-digit key difference is calculated. The result of the test is tabulated in Table 8. The expected result for a good block cipher should be within the range of 0.5 or 50% modification of the cipher text bits using the BER formula [35]. The same plaintext has been tested using a 128-bit plaintext/ciphertext correlation (PCC) key which consists of 128 sequences to make a comparison among the proposed enhanced 3D RECTANGLE algorithm, extended RECTANGLE algorithm, and RECTANGLE for key sensitivity test. Table 9 shows the comparison of Key Sensitivity Test Results (All Key Bits Zero)

Table 8
Comparison of Key Sensitivity Test Results (All Key bits F)

Plaintext	Key	Similar Bits	Proposed Average Different Bit	Simple Rectangle Average Similar bit	BER (Simple Rectangle)
All zero (Hex)		5038	24.64	29.7656	0.472900390625
All one	All F (128	4734	27.015625	32.6953125	0.5186767578125
All F	Flips)	4892	25.78125	29.8671875	0.4744873046875
Random 1		5072	24.375	34.625	0.548828125
Random 2		4934	25.4453125	34.828125	0.552001953125

Table 9
Comparison of Key Sensitivity Test Results (All Key Bits Zero)

Plaintext	Key	Similar Bits	Proposed Average Different Bit	Proposed BER
Random 1		4560	28.375	0.4476744186046512
Random 2	All 0 bits	4472	29.0625	0.4583333333333333
Random 3	(128 Flips)	4468	29.09375	0.4588178294573643
Random 4		4672	27.5	0.43410852713178294
Random 5		4396	29.65625	0.4675387596899225

The result shows that the proposed enhanced 3D RECTANGLE average BER has obtained approximately 0.31. Similar observations considering one fixed key while keeping a fixed plaintext were tested against extended 3D RECTANGLE and RECTANGLE algorithms depicted in Table 10. The BER of the extended RECTANGLE and RECTANGLE is approximately 0.49 and 0.495 respectively.

Table 10
Comparison of Key Sensitivity Test Results

Algorithm Name	Average Bit Error Rate	Reference
Enhanced 3D RECTANGLE	31 %	Proposed
Extended 3D RECTANGLE	50 %	[4]
TEA	49.12 %	[33]
SIMECK	53 %	[33]
QLT	52.56 %	[33]
PRINT	49.08 %	[33]
PRINCE	58.18 %	[33]
LRBC	58 %	[33]
LED	52.83 %	[33]

4. Conclusions

This proposed work aims to design a secure cryptosystem that improves the RECTANGLE algorithm through the use of a composite S-Box and 3D shift rotation techniques. The composite field S-Box employs a pipelined design to reduce both gate count and area, while still providing robust security. The 3D shift rotation further strengthens the security of the proposed algorithm. Moreover, the algorithm achieves a Bit Error Rate (BER) improvement of 31% and demonstrates effectiveness against both linear and differential cryptanalysis. Additionally, the avalanche effect analysis shows that the enhanced 3D RECTANGLE algorithm shows better results than existing algorithms.

References

[1] Kobo, Hlabishi I., Adnan M. Abu-Mahfouz, and Gerhard P. Hancke. "A survey on software-defined wireless sensor networks: Challenges and design requirements." *IEEE access* 5 (2017): 1872-1899. https://doi.org/10.1109/ACCESS.2017.2666200

[2] Christidis, Konstantinos, and Michael Devetsikiotis. "Blockchains and smart contracts for the internet of things." *IEEE access* 4 (2016): 2292-2303. https://doi.org/10.1109/ACCESS.2016.2566339

[3] Thakor, Vishal A., Mohammad Abdur Razzaque, and Muhammad RA Khandaker. "Lightweight cryptography algorithms for resource-constrained IoT devices: A review, comparison and research opportunities." *IEEE Access* 9 (2021): 28177-28193. https://doi.org/10.1109/ACCESS.2021.3052867

[4] Christidis, Konstantinos, and Michael Devetsikiotis. "Blockchains and smart contracts for the internet of things." *IEEE access* 4 (2016): 2292-2303. https://doi.org/10.1109/ACCESS.2016.2566339

[5] Marjani, Mohsen, Fariza Nasaruddin, Abdullah Gani, Ahmad Karim, Ibrahim Abaker Targio Hashem, Aisha Siddiqa, and Ibrar Yaqoob. "Big IoT data analytics: architecture, opportunities, and open research challenges." *ieee access* 5 (2017): 5247-5261. https://doi.org/10.1109/ACCESS.2017.2689040

[6] Zakaria, Abdul Alif, A. H. Azni, Farida Ridzuan, Nur Hafiza Zakaria, and Maslina Daud. "Extended RECTANGLE algorithm using 3D bit rotation to propose a new lightweight block cipher for IoT." *IEEE Access* 8 (2020): 198646-198658. https://doi.org/10.1109/ACCESS.2020.3035375

[7] Usman, Muhammad, Irfan Ahmed, M. Imran Aslam, Shujaat Khan, and Usman Ali Shah. "SIT: a lightweight encryption algorithm for secure internet of things." *arXiv preprint arXiv:1704.08688* (2017). https://doi.org/10.14569/IJACSA.2017.080151

[8] Li, Hang, Jiongjiong Ren, and Shaozhen Chen. "Improved integral attack on reduced-round simeck." *IEEE Access* 7 (2019): 118806-118814. https://doi.org/10.1109/ACCESS.2019.2936834

[9] Ren, Jiongjiong, and Shaozhen Chen. "Cryptanalysis of reduced-round speck." *IEEE Access* 7 (2019): 63045-63056. https://doi.org/10.1109/ACCESS.2019.2917015

[10] Dalmasso, Loic, Florent Bruguier, Pascal Benoit, and Lionel Torres. "Evaluation of SPN-based lightweight crypto-ciphers." *IEEE Access* 7 (2019): 10559-10567. https://doi.org/10.1109/ACCESS.2018.2889790

[11] Zhao, Hongluan, Guoyong Han, Letian Wang, and Wen Wang. "MILP-based differential cryptanalysis on round-reduced Midori64." *IEEE Access* 8 (2020): 95888-95896. https://doi.org/10.1109/ACCESS.2020.2995795

[12] Liu, Juhua, Guoqiang Bai, and Xingjun Wu. "Efficient hardware implementation of roadrunner for lightweight application." In *2016 IEEE Trustcom/BigDataSE/ISPA*, pp. 224-227. IEEE, 2016. https://doi.org/10.1109/TrustCom.2016.0067

[13] Biswas, Arpita, Abhishek Majumdar, S. Nath, A. Dutta, and Krishna Lal Baishnab. "LRBC: a lightweight block cipher design for resource constrained IoT devices." *Journal of Ambient Intelligence and Humanized Computing* (2023): 1-15. https://doi.org/10.1109/AsiaJCIS.2018.00020

[14] Ge, Jing, Yifan Xu, Ruiqian Liu, Enze Si, Ning Shang, and An Wang. "Power attack and protected implementation on lightweight block cipher SKINNY." In *2018 13th Asia Joint Conference on Information Security (AsiaJCIS)*, pp. 69-74. IEEE, 2018.

[15] Philip, Merly Annie, V. Vaithiyanathan, and Kurunandan Jain. "Implementation analysis of rectangle cipher and its variant." In *2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, pp. 474-479. IEEE, 2018. https://doi.org/10.1109/RTEICT42901.2018.9012154

[16] Sutar, Swapnil A. "Differential power attack analysis of ultra-lightweight block cipher BORON." In *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, pp. 365-370. IEEE, 2018. https://doi.org/10.1109/ICECA.2018.8474902

[17] Liu, Bo-Tao, Lang Li, Rui-Xue Wu, Ming-Ming Xie, and Qiu Ping Li. "Loong: A family of involutional lightweight block cipher based on SPN structure." *IEEE Access* 7 (2019): 136023-136035. https://doi.org/10.1109/ACCESS.2019.2940330

[18] Ali, Tasnuva, A. H. Azni, and Nur Hafiza Zakaria. "Issues in Lightweight Encryption Algorithm For mHealth."

[19] Zhang, Wentao, Zhenzhen Bao, Dongdai Lin, Vincent Rijmen, Bohan Yang, and Ingrid Verbauwhede. "RECTANGLE: a bit-slice lightweight block cipher suitable for multiple platforms." *Cryptology ePrint Archive* (2014). https://doi.org/10.1007/s11432-015-5459-7

[20] Wang, Jingjing, Chunxiao Jiang, Tony QS Quek, Xinbing Wang, and Yong Ren. "The value strength aided information diffusion in socially-aware mobile networks." *IEEE Access* 4 (2016): 3907-3919. https://doi.org/10.1109/ACCESS.2016.2600526

[21] Aghaie, Anita, Mehran Mozaffari Kermani, and Reza Azarderakhsh. "Fault diagnosis schemes for secure lightweight cryptographic block cipher RECTANGLE benchmarked on FPGA." In *2016 IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, pp. 768-771. IEEE, 2016. https://doi.org/10.1109/ICECS.2016.7841315

[22] Zhang, Wentao, Zhenzhen Bao, Vincent Rijmen, and Meicheng Liu. "A New Classification of 4-bit Optimal S-boxes and Its Application to PRESENT, RECTANGLE and SPONGENT." In *Fast Software Encryption: 22nd International Workshop, FSE 2015, Istanbul, Turkey, March 8-11, 2015, Revised Selected Papers 22*, pp. 494-515. Springer Berlin Heidelberg, 2015. https://doi.org/10.1007/978-3-662-48116-5_24

[23] Prathiba, A., and VS Kanchana Bhaaskaran. "Lightweight S-box architecture for secure internet of things." *Information* 9, no. 1 (2018): 13. https://doi.org/10.3390/info9010013

[24] El-Sheikh, Hanem M., Omayma A. El-Mohsen, Senior Talaat Elgarf, and Abdelhalim Zekry. "A new approach for designing key-dependent S-box defined over GF (24) in AES." *International Journal of Computer Theory and Engineering* 4, no. 2 (2012): 158. https://doi.org/10.7763/IJCTE.2012.V4.442

[25] Wong, Ming Ming, ML Dennis Wong, Asoke K. Nandi, and Ismat Hijazin. "Construction of optimum composite field architecture for compact high-throughput aes s-boxes." *IEEE transactions on very large scale integration (VLSI) systems* 20, no. 6 (2011): 1151-1155. https://doi.org/10.1109/TVLSI.2011.2141693

[26] Nakahara Jr, Jorge. "3D: A three-dimensional block cipher." In *International Conference on Cryptology and Network Security*, pp. 252-267. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. https://doi.org/10.1007/978-3-540-89641-8_18

[27] NakaharaJr, Jorge. "New impossible differential and known-key distinguishers for the 3D cipher." In *Information Security Practice and Experience: 7th International Conference, ISPEC 2011, Guangzhou, China, May 30–June 1, 2011. Proceedings 7*, pp. 208-221. Springer Berlin Heidelberg, 2011. https://doi.org/10.1007/978-3-642-21031-0_16

[28] Suri, Pushpa R., and Sukhvinder Singh Deora. "Design of a modified Rijndael algorithm using 2D Rotations." *IJCSNS Int. J. Comp. Sci. Netw. Secur* 11, no. 9 (2011): 141-145.

[29] Suri, Pushpa R., and Sukhvinder Singh Deora. "3D Array Block Rotation Cipher: An Improvement using shift." *Global Journal of Computer Science and Technology* 11, no. 19 (2011).

[30] Ariffin, Suriyani, Ramlan Mahmod, Azmi Jaafar, and Muhammad Rezal Kamel Ariffin. "Immune systems approaches for cryptographic algorithm." In *2011 Sixth International Conference on Bio-Inspired Computing: Theories and Applications*, pp. 231-235. IEEE, 2011. https://doi.org/10.1109/BIC-TA.2011.33

[31] Wang, Qian, and Chenhui Jin. "A non-alternate 3D structure and its practical security evaluation against differential and linear cryptanalysis." *Science China. Information Sciences* 61, no. 5 (2018): 058102. https://doi.org/10.1007/s11432-017-9181-4

[32] Mushtaq, Muhammad Faheem, Sapiee Jamel, Siti Radhiah B. Megat, Urooj Akram, and Mustafa Mat Deris. "Key schedule algorithm using 3-dimensional hybrid cubes for block cipher." *International Journal of Advanced Computer Science and Applications* 10, no. 8 (2019). https://doi.org/10.14569/IJACSA.2019.0100857

[33] Rukhin, Andrew, Juan Soto, James Nechvatal, Miles Smid, Elaine Barker, Stefan Leigh, Mark Levenson et al. *A statistical test suite for random and pseudorandom number generators for cryptographic applications*. Vol. 22. Gaithersburg, MD, USA: US Department of Commerce, Technology Administration, National Institute of Standards and Technology, 2001. https://doi.org/10.6028/NIST.SP.800-22

[34] Biham, Eli, and Adi Shamir. "Differential cryptanalysis of DES-like cryptosystems." *Journal of CRYPTOLOGY* 4 (1991): 3-72. https://doi.org/10.1007/BF00630563

[35] Wong, Ming Ming, ML Dennis Wong, Asoke K. Nandi, and Ismat Hijazin. "Construction of optimum composite field architecture for compact high-throughput aes s-boxes." *IEEE transactions on very large scale integration (VLSI) systems* 20, no. 6 (2011): 1151-1155. https://doi.org/10.1109/TVLSI.2011.2141693