Defence Science Journal, Vol. 75, No. 5, September 2025, pp. 531-538, DOI : 10.14429/dsj.20567

Design of Dynamic F-Function for Lightweight Block Ciphers

Nagaraj Hediyal* and B P Divakar
REVA University, Bengaluru - 560 064, India
**E-mail: nagaraj_hediyal@yahoo.com*

ABSTRACT

Lightweight block ciphers are crucial for securing data in resource-constrained IoT devices and sensors, where low energy consumption, minimal latency, and high efficiency are essential. However, existing ciphers often fail to balance these requirements. This article proposes a dynamic F-function that reduces iterative rounds while ensuring optimal security, flexibility, and scalability. The function integrates a fixed substitution layer with a dynamic permutation layer, employing Hamming weight-based transformations, circular shifts, and XOR operations for enhanced diffusion. It is integrated into SLIM and Lightweight Block Cipher for IoT (LBC-IoT) ciphers and evaluated. Results confirm improved security and efficiency, making it a promising solution for IoT applications.

Keywords: Dynamic; Diffusion; Block cipher; Lightweight; Substitution; Permutation

NOMENCLATURE

L_i	: Left 16-bit half of the state at round i
R_i	: Right 16-bit half of the state at round i
K_i	: 16-bit round key for round i
ρF	: Round function
P	: Permutation in SLIM
P_1, P_2	: Permutations in LBC-IoT
S	: Substitution Layer
$\oplus$	: XOR operation
<<	: Circular left shift
>>	: Circular right shift
$HW(X)$	: Hamming weight of input X
ΔX	: Input difference for differential analysis

1. INTRODUCTION

With Industry 5.0 and the widespread adoption of IoT devices and sensors, ensuring data security in resource-constrained environments is critical. Traditional cryptographic algorithms are impractical due to high computational overhead.

Lightweight block ciphers (LBCs) optimize flexibility and scalability using iterative rounds, user key sizes, and SPN-based F-functions. However, existing LBCs often struggle to balance security, efficiency, and resource constraints.

This work introduces a dynamic F-function leveraging Hamming weight-based shifts and XOR for enhanced diffusion[18]. Integrated into SLIM and LBC-IoT, it improves security and efficiency while reducing iterative rounds.

2. LITERATURE REVIEW

Cryptographic algorithms follow the confusion and diffusion principles introduced by C.E. Shannon in 1949[1]. Lightweight block ciphers typically adopt Feistel, generalized Feistel, or Substitution-Permutation Networks (SPNs)[2-3]. In Feistel structures, the round function involves add-round key, rotation, swapping, substitution, and permutation, iterated to achieve security[2-6].

Lightweight block ciphers use 4×4 non-linear S-boxes for confusion[7-8], while diffusion layers rely on Maximum Distance Separation (MDS) matrices or bit/byte/nibble permutations[9-14]. For example, DES[15] employs bit permutation, AES[16] uses an MDS-based diffusion layer, and Few[17] applies nibble permutation. SPN-based F-functions are widely used in encryption, decryption, and key schedules[18].

2.1 Contributions and Article Organization

The following are the technical and contextual contributions of this research paper:

2.2.1 Novel Dynamic F-Function

Utilizes logical operations, circular shifts, and Hamming weight-based transformations to enhance diffusion while reducing iterative rounds without compromising security.

2.2.2 Advanced Dynamic Diffusion Layer

Incorporates multiplication and division to increase algebraic complexity and strengthen resistance against cryptanalysis.

2.2.3 Improved Flexibility & Security

Achieves high efficiency with fewer rounds while maintaining a fixed user key size, eliminating the need for variable key-length adjustments.

This article is structured as follows: Section 2 presents an overview of SLIM[8] and LBC-IoT[9]. Section 3 details the proposed Dynamic F-function. Section 4 evaluates its impact on security and efficiency. Section 5 concludes the study.

Received : 01 October 2024, Revised : 06 February 2025
Accepted : 30 April 2025, Online published : 01 September 2025

3. OVERVIEW OF MODEL CIPHERS

This section provides a structured overview of the SLIM and LBC-IoT ciphers, which serve as models for integrating the proposed dynamic F function.

Table 1. Specifications

Specifications	SLIM	LBC-IoT
Block size	32-bit	32-bit
Key size	80-bit	80-bit
Rounds	32-bit	32-bit
Branch size	16-bit	16-bit

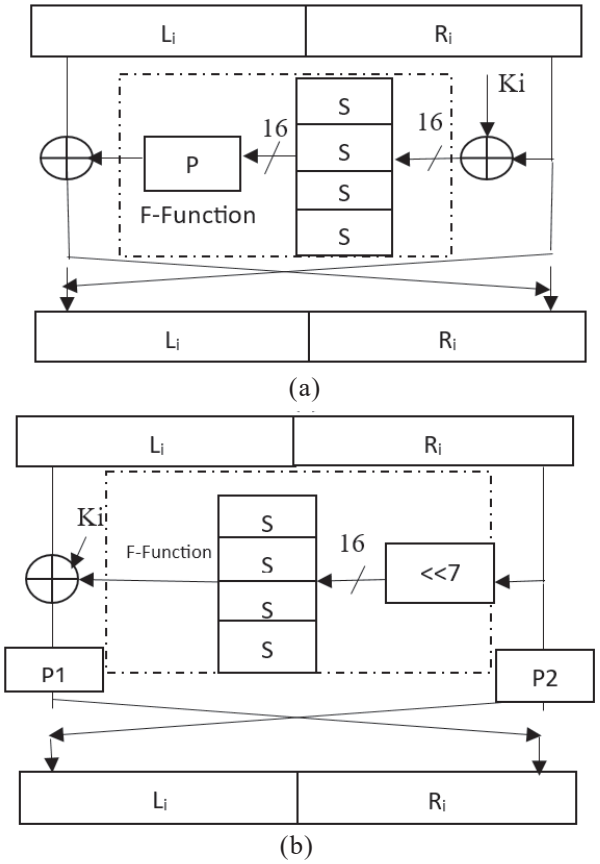

Figure 1. Structure (a) SLIM[8], (b) LBC-IoT[9].

Figure 1 (a) depicts the structure of the SLIM cipher. The following equations represent the data process in the encryption schedule.

$$\begin{cases} L_i = R_{i-1} \\ R_i = L_{i-1} \oplus P(S(K_i \oplus R_{i-1})) \end{cases} \quad (1)$$

The round function follows the structure:

- Li and Ri are inputs (16-bits each)to the function.
- Ri is XORed with Ki (16-bit).
- The function substitutes values using S-boxes.
- The permutation layer shuffles the bits for better diffusion and XORed with Li. The results are swapped.

Figure 1 (b) depicts the structure of the LBC-IoT cipher. The following equations represent the data process in the encryption schedule.

$$\begin{cases} L_i = P_2(R_{i-1}) \\ R_i = P_1[L_{i-1} \oplus K_i \oplus S(R_{i-1} \ll 7)] \end{cases} \quad (2)$$

The round function follows the structure:

- Li and Ri are inputs (16-bits each)to the function
- A Circular left shift (<<7) is applied to Ri to enhance diffusion
- The result of circularly shifted Ri is substitute values using S-boxes
- The substituted values are XORed with Li (16-bit) and Ki (16-bit), followed by permutation using P1
- Simultaneously, Ri undergoes permutation using P2
- Finally, the resultant branches undergo swapping.

Table 2. Trails of SLIM and LBC-IoT

SLIM				
Linear trails				
Round	**Block1**	**Block 2**	**Active S-box**	**Remark**
1	0x0000	0x8000	1	
2	0x8000	0x0420	2	4
3	0x0420	**0x6000**	1	
Differential trails				
Round	**Block1**	**Block 2**	**Active S-box**	**Remark**
1	0x0000	0x8000	1	5
2	0x8000	0x8001	2	
3	0x8001	0x1003	2	
LBC-IoT				
Linear trails				
Round	**Block1**	**Block 2**	**Active S-box**	**Remark**
1	0x0004	0x2000	1	
2	0x0400	0x0044	1	4
3	0x4010	**0x4028**	2	
Differential trails				
Round	**Block1**	**Block 2**	**Active S-box**	**Remark**
1	0x0004	0x0100	1	
2	0x1000	0x0440	1	4
3	0x4020	0x02A0	2	

4. DYNAMIC F-FUNCTION

In this section, the authors discuss preliminaries, structure, flow chart, and the operation of the dynamic diffusion or permutation layer.

4.1 Preliminaries

This section explains the concepts of hamming weight, circular left shift, circular right shift, and non-linearity.

Definition 1. For any $x \in F_2^n$, Hamming weight HW(x) is the number of 1s in x.

For example, let *x*=0001001001101111, then hw(x) is 8.

Note: Denote by F_2 the finite field of two elements {0, 1} and by F_2^n the n-dimensional vector space over F_2.

Definition 2. A left circular shift permutes an n-entry tuple such that,

$$\sigma i \equiv i+1 \bmod n \qquad for\ all\ i = 1.... n \tag{3}$$

Let x = 0001001001101111, then, the circular left shift by 1 yield,

x<<1 = 0010010011011110.

Definition 3. A right circular shift permutes an n-entry tuple such that,

$$\sigma i = i-1 \bmod n \qquad for\ all\ i = 1....n \tag{4}$$

Let x = 0001001001101111, then, the circular right shift by 1 yield,x>>1 = 1000100100110111.

Definition 4. (Active S-box) An S-box is active if its input and output masks for that characteristic are non-zero.

Definition 5. (A weight of a state) Let $x = [x_1, x_2, ...x_n \in F_2^n$. Then, $n_\alpha = x / \alpha$, where n_α is called a set of 'α' values. The number of 'α' sets, that have at least one non-zero bit is the weight of a state and is denoted by [$w_\alpha(x)$].

Definition 6. (Branch Number) Let $x = x_1, x_2, ...x_n \in F_2^n$. Let $f: \{0,1\}^{16} \rightarrow \{0,1\}^{16}$ be a function to which x is a 16-bit input whose branch number is given by

$$\beta(f) = \min_{x=0, x\in 0,1^{16}} (wt(x) + wt(f(x))) \tag{5}$$

Definition 7. (Differential branch number) Let ΔX and DDL(ΔX) (Dynamic Diffusion Layer) be the input and output differences. Let β_d be the differential branch number and is given by

$$\beta(DDL) = \min_{\Delta x \neq 0} (wt(\Delta x) + wt(DDL(\Delta x))) \tag{6}$$

Definition 8. For any $x \in F_2^n$, the upper bound for circular shifts (Left or Right) in a dynamic diffusion layer for an n-bit input word (x) is n-1.

4.2 Dynamic F-function

The top-level structure of the dynamic F-function is in Fig. 2 (a).

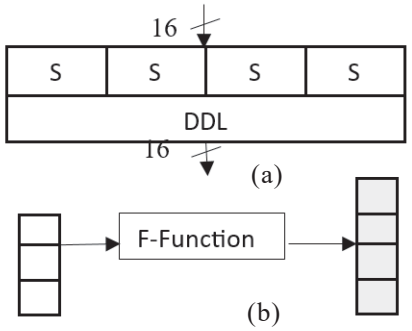

Figure 2. (a) Dynamic F function; and (b) S-box transition.

The structure encompasses four 4-bit substitution boxes arranged in a single row and a permutation layer employs XOR operations with left, right, or mixed circular shifts based on hamming weight, hence the name dynamic diffusion layer.

Let $x = [x_1, x_2, ...x_n] \in F_2^n$, be the input to the dynamic F-function. Where n=16 bits.

Equation 7 represents the iterative transformation applied in the dynamic diffusion layer, where four successive XOR and shift operations modify the input state.

The concatenated X serves as the output of the substitution layer after processing input x. The DDL(X) is defined as follows.

$$DDL(X) = X \oplus X1 \oplus X2 \oplus X3 \oplus X4 \tag{7}$$

Where,

$X1 = X$ CE $HW(X)$,

$X2 = [X1$ CE $HW(X1)$,

$X3 = X2$ CE $HW(X2)]$

$X4 = [X3$ CE $HW(X3)]$.

The ideal branch number for a diffusion layer with four branches could be five (Fig. 2 (b)).

Note: CE indicates the circular right or left shift and exclusive-OR operation. The right or left circular shift depends upon the hamming weight of the input.

The output of the dynamic F-function is as follows.

$$F = DDL(S(x)) \tag{8}$$

Where $S(x)=X$, substitution layer output.

4.3 DDL-Operation

The dynamic diffusion layer accepts a 16-bit input and performs four circular shifts (Left or Right or in combination) and four XOR operations as detailed below:

- Let $X = [x_1 \| x_2 \| x_3 \| x_4]$
- Determine the hamming weight HW(X).
- Check HW(X) is odd or even.
- If HW(X) is odd, perform $X \ll HW(X)$, then $X1 = X \oplus (X \ll HW(X))$. Or, if HW(X) is even, perform $X \gg HW(X) - 1$, then $X1 = X \oplus (X \gg HW(X) - 1)$.
- Repeat steps ii–v four times to complete the DDL transformation.

The dynamic diffusion layer applies four transformations, each using a circular shift and XOR operation. Figure 3 illustrates this iterative process, ensuring full diffusion in four steps.

For brevity, the process is illustrated step by step with the following examples:

i. x=0000 0000 0000 0001
ii. HW(x)= 1.
iii. HW(x) is odd.
iv. x = 0000 0000 0000 0001
⊕ x<< HW(x) = 0000 0000 0000 0010
x = 0000 0000 0000 0011
v. HW(x) = 2.
vi. HW(x) is even.
x = 0000 0000 0000 0011
⊕ x>> HW(x)-1 = 1000 0000 0000 0001
x = 1000 0000 0000 0010

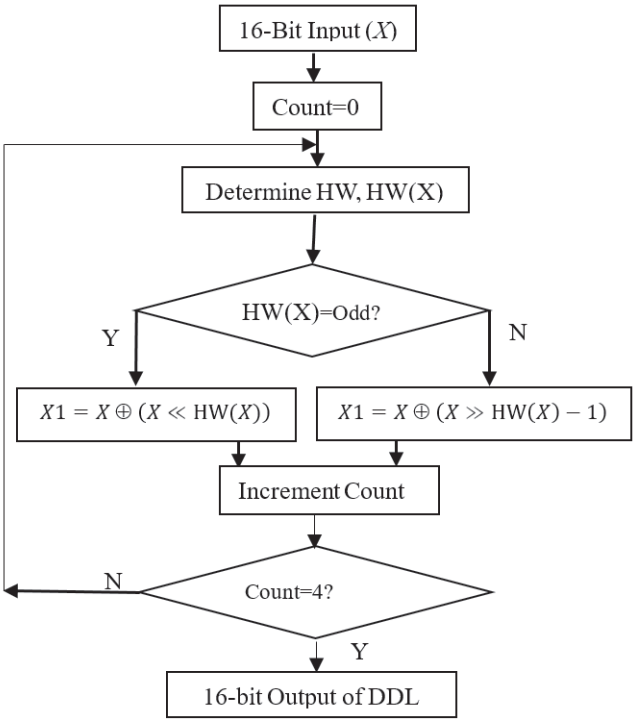

Figure 3. DDL-flow Chart.

vii. HW(x) = 2.
viii. HW(x) is even.
x = 1000 0000 0000 0010
⊕ x>> HW(x)-1 = 0100 0000 0000 0001
x = 1100 0000 0000 0011
ix. HW(x) = 4.
x. HW(x) is even.
x = 1100 0000 0000 0011
⊕ x>> HW(x)-1 = 0111 1000 0000 0000
x = 1011 1000 0000 0011

4.4 Implementation Strategy

Figure 4(a) (SLIM Implementation): SLIM encrypts/ decrypts by processing Li, Ri via substitution, Hamming weight, XOR, and a 2:1 MUX for Ki application.

Figure 4(b) (LBC-IoT Implementation): This figure represents the encryption/decryption process of LBC-IoT, which follows a similar structure but includes two permutations (P1, P2) and circular shift operations for improved diffusion. The design ensures efficient key mixing and security against cryptanalysis. The key schedule implementation remains unchanged[8-9].

5. EVALUATION OF MODIFIED CIPHER MODELS

5.1 Security Analysis

Differential and linear characteristics were identified using the branch-and-bound technique[18,19] and empirically validated. While not all input differences are listed due to space constraints, the selected characteristics indicate strong resistance to differential and linear attacks. Figure 2(b) illustrates the S-box transition path through the Dynamic F-function.

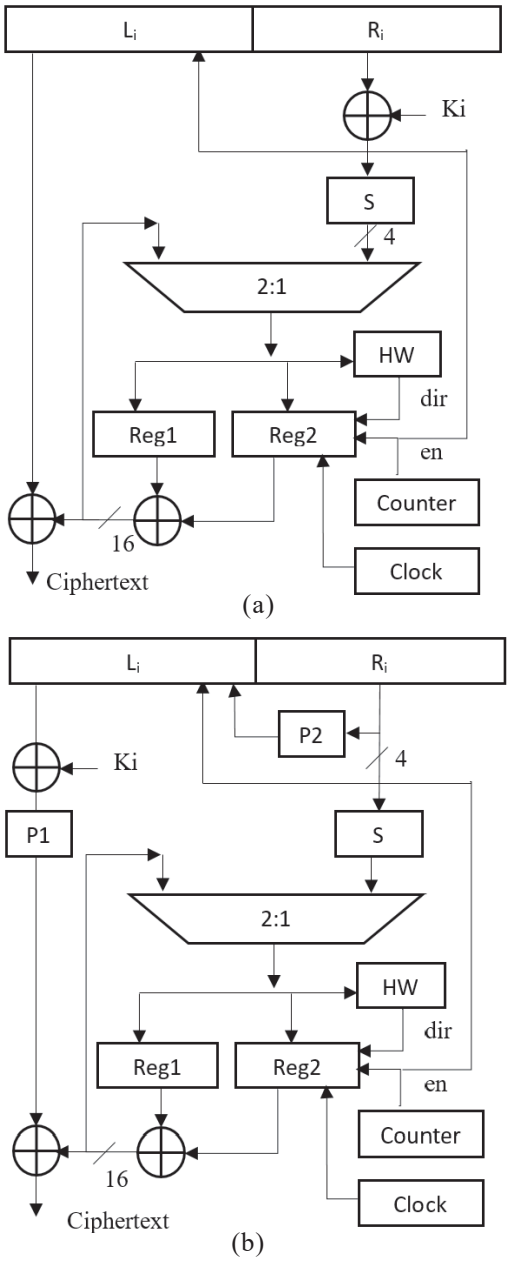

Figure 4. Implementation strategy, (a) SLIM; and (b) LBC-IoT.

5.1.1 Modified SLIM

Theorem 1. Any three-round differential characteristics of modified SLIM cipher exhibit at least seven active S-boxes.

Proof. Active S-boxes per round indicate resistance to differential and linear attacks. Theorem 1 is proved using the following procedure.

Let $\Delta P_i = \Delta L_{i-1} \,||\, \Delta R_{i-1}$ be the i[th] round input difference characteristic with two segments concatenated. The procedure counts active S-boxes by initializing the cipher with various input variants.

Instantiation 1. Let $\Delta P_1 = 0\times0000 \,||\, 0\times8000$ be the input difference to the cipher in the first iteration. Therefore, $\Delta L_{1\text{-}1}$=0×0000 and $\Delta R_{1\text{-}1}$=0×8000. With an active bit in the right half's first nibble, a single active S-box (Δ1=1) emerges in the first iteration. There will be Δ2=2 and Δ3=4 active S-boxes at the end of the 2[nd] and 3[rd] iterations. Thus, modified SLIM has Δ1+Δ2+Δ3=1+2+4=7 active S-boxes over three iterations.

Instantiation 2. Let $\Delta P_1 = 0\times0000 \,||\, 0\times0400$ be the input difference to the cipher in the first iteration. Therefore, ΔL_{1-1}=0×0000 and ΔR_{1-1}=0×0400. With an active bit in the right half's second nibble, a single active S-box (Δ1=1) emerges in the first iteration. There will be Δ2=3 and Δ3=3 active S-boxes at the end of the 2nd and 3rd iterations. Thus, modified SLIM has Δ1+Δ2+Δ3=1+3+3=7 active S-boxes over three iterations.

Instantiation 3. Let $\Delta P_1 = 0\times0000 \,||\, 0\times0020$ be the input difference to the cipher in the first iteration. Therefore, ΔL_{1-1}=0×0000 and ΔR_{1-1}=0×0020. With an active bit in the right half's third nibble, a single active S-box (Δ1=1) emerges in the first iteration. There will be Δ2=2 and Δ3=4 active S-boxes at the end of the 2nd and 3rd iterations. Thus, modified SLIM has Δ1+Δ2+Δ3=1+2+4=7 active S-boxes over three iterations.

Instantiation 4. Let $\Delta P_1 = 0\times0001 \,||\, 0\times0000$ be the input difference to the cipher in the first iteration. Therefore, ΔL_{1-1}=0×0001 and ΔR_{1-1}=0×0000. Since the right half has no active nibble, it results in zero active S-boxes (Δ1=0) at the end of the first iteration. There will be Δ2=3 and Δ3=4 active S-boxes at the end of the 2nd and 3rd iterations. Thus, modified SLIM has Δ1+Δ2+Δ3=0+3+4=7 active S-boxes over three iterations.

Experimental results show that the active S-box count remains **seven (7)** across multiple input difference variants. Thus, any three-round differential characteristic of the modified SLIM cipher has at least **seven active S-boxes**, proving the theorem.

Lemma 1. By Piling-Up-Lemma, the maximum differential probability (MDP) of the 3-round modified cipher SLIM having seven active S-boxes is equal to $P_{3D}=(2^{-2})^7=2^{-14}$.

Lemma 2. The maximum differential probability of the 32-round modified cipher SLIM having 32 *x* 7/3 is $P_D = (2^{-2})32x7/3 \cong 2^{-149.33}$.

Theorem 2. Any three-round linear approximations of modified SLIM cipher exhibit at least seven active S-boxes.

Proof: The active S-box count remains the same since the differential branch number is the same as the linear branch number V. Rijmen, *et al.* (1996)[16].

Hence the Theorem.

Lemma 3. By Piling-Up-Lemma (Matsui), the maximum linear probability (MLP) of the 3-round modified cipher SLIM with seven active S-boxes is $P_{3L}=2^{7-1}\times(2^{-2})^7=2^{-8}$.

Lemma 4. The maxim linear probability of the 32-round modified cipher SLIM having 32 *x* 7/3 active S-boxes is $P_L=2^{11-1}\times(2^{-8})^{11}=2^{-78}$.

The differential trails of modified SLIM are in Table 3 (a).

Theorem 3. The modified SLIM lightweight block cipher resists an impossible differential attack beyond eight iterative rounds.

Proof. The impossible differential characteristic of a six-round modified SLIM with defined states is 0000000α↛00000α00. By exploiting zero-probability differentials, round subkey bits can be recovered. Unlike differential cryptanalysis, this method eliminates incorrect keys from the round-key list, leaving only the correct candidates for the secret key. Since each **S-box is 4-bit**, four subkey bits must be guessed per S-box. A six-round differential characteristic has **14 active S-boxes**, as any three-round characteristic of the modified SLIM cipher contains at least **seven active S-boxes**. The total subkey bits for 14 active S-boxes is **14 × 4 = 56**. If two additional rounds are included at the bottom or on top, the number of subkey bits required to guess increases to 75 (8×7/3×4=75). To attack the cipher of eight rounds, the data complexity required is 2^{75}, which is much more than the available data. Hence, the theorem.

To summarize, the modified SLIM is safe against linear, differential, and impossible differential attacks.

5.1.2 Modified LBC-IoT

Theorem 4. Any three-round differential characteristics of modified LBC-IoT cipher exhibit at least eight active S-boxes

Proof. Let $\Delta P_1 = 0\times0000 \,||\, 0\times8000$ be the input difference to the cipher in the first iteration. Therefore, ΔL_{1-1}=0×0000 and ΔR_{1-1}=0×8000.

To prove Theorem 3, we followed the procedure adopted to prove Theorem 1. The differential trails of modified LBC-IoT are in Table 3 (b).

Lemma 5. By Piling-Up-Lemma (Matsui,1993)[14], the Maximum Differential Probability (MDP) of the 3-round modified cipher LBC-IoT having eight active S-boxes is equal to $P_{3D}=(2^{-2})^8=2^{-16}$.

Lemma 6. The maximum differential probability of the 32-round modified cipher LBC-IoT having 32 *x* 8/3 active S-boxes is $P_D = \left(2^{-2}\right)^{32x8/3} \cong 2^{-170.66}$.

Theorem 5. Any three-round linear approximations of modified LBC-IoT cipher exhibit at least eight active S-boxes.

Proof: The active S-box count remains the same since the differential branch number is the same as the linear branch number V. Rijmen et al. (1996)[18].

Hence the Theorem.

Lemma 7. By Piling-Up-Lemma (Matsui,1993), the Maximum Linear Probability (MLP) of the 3-round modified cipher LBC-IoT with eight active S-boxes is $P_{3L}=2^{8-1}\times(2^{-2})^8=2^{-12}$.

Lemma 8. The maxim linear probability of the 32-round cipher SLIM with eleven active S-boxes is $P_L=2^{11-1}\times(2^{-12})^{11}=2^{-122}$.

Theorem 6. The modified LBC-IoT lightweight block cipher resists an impossible differential attack beyond seven iterative rounds.

Proof. The impossible differential characteristic of a six-round modified LBC-IoT with defined states is 0000000α↛00α00000. By exploiting zero-probability differentials, round subkey bits can be recovered.

Since each **S-box is 4-bit**, four subkey bits must be guessed per S-box. A six-round differential characteristic has **16 active S-boxes**, as any three-round characteristic of the modified LBC-IoT cipher contains at least **seven active S-boxes**. The total subkey bits for 16 active S-boxes is **16×4=64**. If one additional round is included at the bottom or on top, the number of subkey bits required to guess increases to 76 (7×8/3×4=75). To attack the cipher of eight rounds, the data complexity required is 2^{75}, which is much more than the available data. Hence, the theorem.

To summarize, the modified LBC-IoT is safe against linear, differential, and impossible differential attacks.

Table 3. Security evaluation results

Differential trails (SLIM)				
Round	**L_i**	**R_i**	**Active S-box**	**Remark**
0	0x0000	0x8000	-	14
1	0x8000	0xEE00	1	
2	0xEE00	0x22A2	2	
3	0x22A2	0xE428	4	
4	0xE428	0x0989	4	
5	0x0989	0x----	3	
Differential trails (LBC-IoT)				
Round	**L_i**	**R_i**	**Active S-box**	**Remark**
0	0x0000	0x8000	-	8
1	0x0400	0x2304	1	
2	0x4822	0xA177	2	
3	0x7C65	0xA48A	4	

Table 4. Complexity in the AVR ATmega 328P platform

Cipher modified	Metrics	Attribute	Total
SLIM	Time complexity	Round key generation Encryption Decryption	223.43 µs
	Space complexity	Flash memory	1236 Bytes
		SRAM	
		EEPROM	
	Power	Power	22.35 mW
LBC-IoT	Time complexity	Round key generation Encryption Decryption	241.27 µs
	Space complexity	Flash memory	1306 Bytes
		SRAM	
		EEPROM	
	Power	Power	22.83 mW

Table 5. GE estimation details (SLIM & LBC-IoT)

Component	GE count	Remark
32-bit register	32 x 4.25=136	Input data state
Reg 1 (16-bit)	16 x 4.25 = 68	Intermediate state
Reg 2 (16-bit)	16 x 4.25 = 68	Intermediate state
XOR	5 x 7.5 = 37.5	4-bit and 16-bit
MUX	2 x 2.25 = 4.5	Input selection to Reg1 and Reg 2
Substitution	1 x 13 = 13	Input side
Counter and hamming weight	15	To accommodate control signals
Total GE	342	

The branch-and-bound technique identifies high-probability differential characteristics. MILP verification adds insights, but selected trails are experimentally validated as optimal.

4.2 Resource Efficiency

The AVR ATmega328P microcontroller, based on the AVR (Alf and Vegard's RISC) architecture, was used to evaluate the performance of the modified model ciphers.

Table 5 presents the estimated area for the modified encryption/decryption schedules of the SLIM and LBC-IoT ciphers.

4.3 Flexibility and Scalability

The proposed Dynamic F-function ensures flexibility without requiring variable block or user key sizes, as its dynamic diffusion layer maintains security. Additionally, its use of multiplication and division increases complexity.

Table 6 shows that after 12 rounds, security improves significantly, with the Dynamic F-function increasing active S-boxes while maintaining efficiency at a 7.5 % resource cost. The modified SLIM and LBC-IoT ciphers exhibit 75 % and 60 % more active S-boxes, respectively, enhancing resistance to differential and linear attacks while reducing iterative rounds. The added algebraic complexity further strengthens security by making key recovery more difficult.

Performance evaluation on the AVR ATmega328P platform confirms that these security gains require only a 7.5 % increase in resource usage, which remains practical for constrained environments. Unlike PRESENT[4], SIMON[19], and GIFT[20], the Dynamic F-function employs adaptive Hamming weight-based permutation, achieving stronger security with fewer rounds while ensuring IoT efficiency.

5. CONCLUSION

This article introduces a Dynamic F-function to enhance security and efficiency in lightweight block ciphers for resource-constrained IoT systems. It leverages Hamming weight-based transformations to improve diffusion and reduce iterative rounds without sacrificing security strength. Integrated into SLIM and LBC-IoT, the modified F-function increases

Table 6. Comparative analysis of original and modified SLIM and LBC-IoT ciphers

Parameter		Cipher								Rounds
		SLIM				LBC-IoT				
		Old		Modified		Old		Modified		
		MDP	MLP	MDP	MLP	MDP	MLP	MDP	MLP	
Security		2^{-10}	2^{-6}	2^{-14}	2^{-8}	2^{-8}	2^{-5}	2^{-16}	2^{-9}	3
		2^{-20}	2^{-11}	2^{-28}	2^{-15}	2^{-16}	2^{-9}	2^{-32}	2^{-17}	6
		2^{-30}	2^{-16}	2^{-42}	2^{-22}	2^{-24}	2^{-13}	2^{-48}	2^{-25}	9
		2^{-40}	2^{-21}	**2^{-56}**	**2^{-29}**	2^{-32}	2^{-17}	**2^{-64}**	**2^{-33}**	12
		2^{-50}	2^{-26}	2^{-70}	2^{-36}	2^{-40}	2^{-21}	2^{-80}	2^{-41}	15
		2^{-60}	2^{-31}	2^{-84}	2^{-43}	2^{-48}	2^{-25}	2^{-96}	2^{-49}	18
		2^{-70}	2^{-36}	2^{-98}	2^{-50}	2^{-56}	2^{-29}	2^{-112}	2^{-57}	21
		2^{-80}	2^{-41}	2^{-112}	2^{-57}	2^{-64}	2^{-33}	2^{-128}	2^{-65}	24
		2^{-90}	2^{-46}	2^{-126}	2^{-64}	2^{-72}	2^{-37}	2^{-134}	2^{-73}	27
		2^{-100}	2^{-51}	2^{-140}	2^{-71}	2^{-80}	2^{-41}	2^{-160}	2^{-81}	30
		2^{-106}	2^{-54}	2^{-149}	2^{-78}	2^{-85}	2^{-44}	2^{-170}	2^{-86}	32
Resource ASIC 0.13 µm	Key schedule	710.25 GE		710.25 GE		710.25 GE		710.25 GE		No change
	Encryption Schedule	318.5 GE		342		314		342		7.5 % additional area required
Active S-boxes	3 rounds	4		7		5		8		
		75 % increment				60 % increment				

active S-boxes, lowers differential and linear probabilities, and strengthens resistance to cryptanalytic attacks. Performance evaluation confirms these gains with minimal computational overhead, ensuring suitability for real-time applications. The Dynamic F-function strengthens LBCs and informs future designs.

REFERENCES

1. Shannon CE. Communication theory of secrecy systems. The Bell System Technical Journal. 1949 Oct;28(4):656-715.
2. Feistel H. Cryptography and computer privacy. Scientific American. 1973 May 1;228(5):15-23.
3. Feistel H, Notz WA, Smith JL. Some cryptographic techniques for machine-to-machine data communications. Proceedings of the IEEE. 1975 Nov 30;63(11):1545-54.
4. Bogdanov A, Knudsen LR, Leander G, Paar C, Poschmann A, Robshaw MJB, et al. PRESENT: An Ultra-Lightweight Block Cipher. Cryptographic Hardware and Embedded Systems - CHES 2007. 2007;450–66. doi:10.1007/978-3-540-74735-2_31
5. Usman M, Ahmed I, Aslam MI, Khan S, Shah UA. SIT: A lightweight encryption algorithm for secure internet of things. arXiv preprint arXiv:1704.08688. 2017 Apr 27.
6. Barreto PS, Rijmen V. The Khazad legacy-level block cipher. Primitive submitted to NESSIE. 2000 Nov 13;97(106).
7. Bansod G, Patil A, Sutar S, Pisharoty N. ANU: An ultra lightweight cipher design for security in IoT. Security and Communication Networks. 2016 Oct 26;9(18):5238–51. doi: 10.1002/sec.1692.
8. Aboushosha B, Ramadan RA, Dwivedi AD, El-Sayed A, Dessouky MM. SLIM: A Lightweight Block Cipher for Internet of Health Things. IEEE Access. 2020;8:203747–57. doi:10.1109/Access.2020.3036589
9. A. Ramadan R, W. Aboshosha B, Yadav K, M. Alseadoon I, J. Kashout M, Elhoseny M. LBC-IoT: Lightweight block cipher for IoT constraint devices. Computers, Materials & Continua. 2021;67(3):3563–79. doi:10.32604/cmc.2021.015519.
10. Guo Y, Li L, Liu B. Shadow: A lightweight block cipher for IoT nodes. IEEE Internet of Things Journal. 2021 Mar 8;8(16):13014-23.
11. Li L, Liu B, Wang H. QTL: A new ultra-lightweight block cipher. Microprocessors and Microsystems. 2016 Aug;45:45–55. doi: 10.1016/j.micpro.2016.03.011
12. Li L, Liu B, Zhou Y, Zou Y. SFN: A new lightweight block cipher. Microprocessors and Microsystems [Internet]. 2018 Jul 1 [cited 2023 Dec 22];60:138–50. doi: 10.1016/j.micpro.2018.04.009.
13. Kyoji Shibutani, Isobe T, Harunaga Hiwatari, Mitsuda A, Toru Akishita, Shirai T. Piccolo: An Ultra-Lightweight Blockcipher. Lecture Notes in Computer Science. 2011 Jan 1;342–57. doi: 10.1007/978-3-642-23951-9_23.
14. Singh D, Kumar M, Yadav T. RAZOR A lightweight block cipher for security in IoT. Defence Science Journal. 2024 Jan 12;74(01):46–52. doi : 10.14429/dsj.74.18421
15. Biham E, Shamir A. Differential cryptanalysis of DES-like cryptosystems. Journal of Cryptology. 1991 Jan;4(1):3-72.
16. J. Daemen and V.Rijimen. The Design of Rijndael. Springer Verrlog; 2001.
17. Kumar M, Pal SK, Panigrahi A. FeW: A lightweight block cipher. Turkish Journal of Mathematics and Computer Science. 2014;11(2):58-73.
18. M. Matsui. Linear cryptanalysis method for DES cipher.

Advances in Cryptology: Proceedings of EUROCRYPT '93, Springer-Verlag, Berlin, pp. 386-397. 1994.
19. Sun S, Hu L, Wang P, Qiao K, Ma X, Song L. Automatic Security Evaluation and (Related-key) Differential Characteristic Search: Application to SIMON, PRESENT, LBlock, DES(L) and Other Bit-Oriented Block Ciphers. Lecture Notes in Computer Science. 2014 Jan 1;158–78. doi:10.1007/978-3-662-45611-8_9.
20. Banik, S., Bogdanov, A., Isobe, T., Jeong, E., Jeong, K., Lee, S., & Sim, S. M. (2017). GIFT: A Small Present. *Cryptology ePrint Archive*.

CONTRIBUTORS

Mr Nagaraj Hediyal obtained MTech in Computer Applications & Industrial Drives. He is currently a Research Scholar at REVA University, Bengaluru, India. His research interests include: Cryptography and network security, embedded systems, VLSI design, secure communication, advanced power electronics, and renewable energy.
He contributed to this study through conceptualization, design, methodology development, software implementation, visualization, and drafting of the original manuscript.

Dr B.P. Divakar obtained PhD in Power Electronics. He is currently serving as the Director of the Research and Development Cell at REVA University, Bengaluru, India. His research areas include: Power systems, power electronics, renewable energy, and system modelling.
His contributions to this study include investigation, data curation, and reviewing and editing of the manuscript.