International Journal of Computer Networks and Applications (IJCNA)
DOI: 10.22247/ijcna/2025/15 Volume 12, Issue 2, March – April (2025)

RESEARCH ARTICLE

NDN: An Ultra-Lightweight Block Cipher to Secure IoT Nodes

Nagaraj Hediyal
School of Electrical and Electronics, REVA University, Bengaluru, Karnataka, India.
✉ nagaraj_hediyal@yahoo.com

Divakar B.P
Research and Development Cell, REVA University, Bengaluru, Karnataka, India.
divakar@reva.edu.in

Received: 12 January 2025 / Revised: 06 March 2025 / Accepted: 22 March 2025 / Published: 30 April 2025

Abstract – The rapid growth of Internet of Things (IoT) technologies in critical infrastructures, including smart grids, healthcare, and intelligent traffic management systems, has significantly enhanced modern living. However, securing resource-constrained IoT nodes presents substantial challenges. This article introduces Neural-Network Driven (NDN), an ultra-lightweight block cipher designed for IoT nodes, employing a novel combination of a 4×4 substitution layer, a primitive polynomial-based bit transformation, an inversion function for enhanced complexity, and a neural network-inspired permutation using a 16-point radix-4 discrete Fourier transform (DFT). NDN supports 64-bit data blocks with 80-bit and 128-bit keys, achieving scalability and adaptability across diverse IoT applications. Comprehensive security analysis demonstrates its robustness against differential, linear, algebraic, related-key, and impossible differential cryptanalysis. Performance evaluation across ASIC and AVR RISC platforms validates NDN's efficiency in real-world IoT environments. This study provides a significant step toward secure, scalable, and adaptable cryptographic solutions for future IoT infrastructures.

Index Terms – Lightweight, Block, Cipher, Energy, Complexity, Artificial, Decipher, Feistel.

1. INTRODUCTION

The rapid expansion of the Internet of Things (IoT) across critical domains, such as smart grids, medical IoT, and industrial automation, has necessitated the development of robust, lightweight cryptographic solutions. IoT devices often operate in resource-constrained environments, requiring ciphers that balance security, scalability, adaptability, and energy efficiency. Over the past two decades, extensive research has been on lightweight block ciphers (LBCs); however, many evaluation methodologies remain inadequate, relying predominantly on conventional metrics such as Gate Equivalents (GE), latency, and memory usage. While these metrics are essential, they fail to account for real-world requirements, such as infrastructure adaptability, operational scalability, and security under dynamic conditions [1, 2]. As a result, many existing comparisons offer a superficial analysis of cryptographic performance, limiting true innovation in the field [3].

Since the round key schedule employs an artificial neural network-based function, specifically a 16-point Radix-4 Discrete Fourier Transform (DFFT), to enhance permutation, the abbreviation Neural-Network Driven (NDN) is incorporated in the title to emphasize this aspect. This novel approach strengthens security by introducing dynamic key transformations, increasing resistance to differential and algebraic attacks. NDN is included in the title to highlight the significance of the key schedule, which is as crucial as the encryption process in ensuring balanced cryptographic strength and efficiency. Due to its technical complexity, this description is omitted from the title but is here to highlight its cryptographic significance. This manuscript presents NDN as a state-of-the-art ultra-lightweight block cipher designed to address the unique security challenges of IoT networks. Unlike conventional LBCs that employ rigid encryption architectures, NDN leverages programmable bit transformation within its round function and a lightweight key scheduling mechanism to enhance security while maintaining computational efficiency. The cipher's design ensures adaptability across various IoT environments, making it suitable for deployment in smart grids, healthcare systems, and industrial IoT applications. Furthermore, this research introduces a context-aware evaluation framework, complementing traditional performance indicators with real-world applicability metrics to facilitate a scientific, application-driven assessment of cryptographic solutions [4, 5].

1.1. Statement of Problem

Existing lightweight block ciphers (LBCs) often lack adaptability and scalability in real-world IoT environments. Most designs rely on fixed transformation patterns, making them vulnerable to targeted attacks and limiting their efficiency in dynamic applications. Additionally, traditional evaluation metrics fail to accurately assess the applicability of these ciphers in practical IoT scenarios, resulting in suboptimal cryptographic solutions. This research presents NDN, an adaptable, scalable, and secure lightweight block cipher designed to meet the unique constraints of IoT devices while ensuring robust security by addressing the above limitations/challenges.

1.2. Objectives of Research

The objectives of this research are as follows:

- Design an ultra-lightweight block cipher (NDN) that integrates programmable bit transformation and a dynamic key scheduling mechanism.
- Evaluate its (NDN) security aspects against differential, linear, algebraic, related-key, and impossible differential cryptanalysis.
- Benchmark it's (NDN) performance against existing lightweight block ciphers using traditional (GE, latency) and context-aware metrics.
- Validate its (NDN) applicability through real-world implementation on ASIC and AVR RISC platforms.

1.3. Contributions

The technical and contextual contributions of this article are as follows.

1.3.1. Dynamic Round Function with High Diffusion and Security

- Contribution:

NDN features a dynamic round function using key-driven bit permutation based on a degree-4 primitive polynomial and a diffusion function with a branch number of 5, ensuring rapid propagation of differences across rounds. This design enhances resistance to differential and linear cryptanalysis by introducing high variability.

- Significance:

Unlike lightweight ciphers with fixed rotations, the dynamic nature of NDN ensures superior cryptographic strength, making it highly suitable for high-security IoT applications like industrial IoT and smart grids.

1.3.2. Dynamic Key Schedule with Adaptive Control

- Contribution:

NDN key schedule incorporates dynamic bit transformations, rotation, segmentation, complement operations, and bit-level permutations, ensuring high randomness in round keys and robust resistance to cryptanalytic attacks.

- Significance:

Unlike traditional fixed-key schedules, this adaptive key schedule disrupts predictability, enhancing security against key schedule-based attacks across diverse IoT domains.

1.3.3. Comprehensive Evaluation with New Metrics

- Contribution:

NDN performance assessment employs context-aware, composite, security-centric, and structural metrics, extending beyond traditional measures like Gate Equivalents (GE), latency, and memory size. These new metrics assess scalability, flexibility, and adaptability for real-world IoT applications.

- Significance:

This holistic evaluation ensures that NDN meets practical IoT demands, emphasizing meaningful innovations over superficial comparisons and setting a higher standard for cryptographic research.

1.3.4. Scalability and Flexibility for Heterogeneous IoT Domains

- Contribution:

NDN supports variable round configurations without increasing memory or key size, enabling scalability across diverse IoT environments, from low-power medical wearables to high-speed industrial controllers.

- Significance:

This flexibility addresses the critical gap in existing ciphers by offering customizable security levels, ensuring efficient operation across heterogeneous IoT applications.

1.3.5. Practical Applicability and Real-World Testing

- Contribution:

The real-world applicability of NDN is validated through testing under realistic IoT operational parameters, demonstrating high energy efficiency, low latency, and optimal hardware area usage.

- Significance:

By incorporating real-world testing and practical metrics, this work establishes NDN as a well-balanced solution for securing resource-constrained IoT systems, setting a benchmark for future cryptographic designs.

1.4. Article Outline

The details of the rest of the sections in this article are as follows. Section 2 presents the related work. Section 3 describes the design aspects of NDN, followed by a security analysis in Section 4. Section 5 describes the performance analysis. Section 6 presents the research summary. Section 7 provides the conclusion with future scope.

2. RELATED WORK

The increasing proliferation of IoT devices across critical domains such as smart grids, medical IoT, and industrial systems has created an urgent need for robust, lightweight cryptographic solutions. These devices are highly resource-constrained, requiring ciphers that offer security besides scalability, adaptability, and energy efficiency. Despite significant research in lightweight block ciphers (LBCs) over the past two decades, the current evaluation landscape remains inadequate, relying heavily on traditional metrics like Gate Equivalents (GE) [6]-[8], latency, and memory usage [9]-[16]. These metrics fail to capture critical real-world aspects such as context-specific performance, infrastructure adaptability, and scalability under varying security requirements. Consequently, superficial comparisons often mislead the research community, hindering the development of innovative cryptographic solutions.

This manuscript presents the NDN lightweight block cipher that addresses these challenges. NDN introduces programmable bit transformation in its round function and a lightweight key scheduling mechanism, enabling dynamic adaptability and improved scalability. Unlike existing ciphers that follow rigid designs, NDN offers flexibility and security tailored to the diverse and dynamic needs of IoT applications. The proposed evaluation framework incorporates traditional and context-aware metrics, ensuring a comprehensive NDN performance analysis. This article demonstrates the NDN's suitability for real-world IoT domains through case studies and comparative analysis, fostering a shift toward more scientific, application-driven cryptographic evaluations.

2.1. Overview of Existing Ciphers

Lightweight block ciphers (LBCs) have been the topic of substantial research [17]-[25] over the past two decades, spurred by the increased demand for secure and efficient cryptographic solutions in limited contexts such as Internet of Things devices. Numerous LBCs have addressed specific application needs, focusing primarily on minimizing hardware complexity and ensuring reasonable performance in resource-limited environments. However, despite the proliferation of LBCs, a critical analysis reveals that many existing designs lack true innovation, relying instead on minor tweaks to well-established structures.

Table 1 Comparison of Lightweight Block Ciphers Implementations

Sl #	Block cipher	Key size (bits)	Block size (bits)	Rounds	Structural Aspects	Ref
1	HIGHT, Hong et al., 2006	128	64	32	GFN, Initial Transformation, Round Function, Final Transformation	[5]
2	PRESENT Bogdanov et al. 2007	80	64	25	SPN, Add Round Key, Substitution, Permutation, Rotation, Counter XOR	[1]
3	Piccolo Shibutani et al. 2011	80	64	30	FSPN, Whitening Key, XOR, F-function, Permutation, Constant Values	[3]
4	LBlock Wu et al. 2011	80	64	32	GFN, XOR, Substitution, Permutation, Rotation, Counter XOR	[53]
5	Klein Gong et al. 2011	64/80/96	64	12/16/20	SPN, Shift, Feistel, Nibble Substitution, Nibble Rotation, Nibble Mix	[54]
6	ANU Gaurav et al. 2016	128/80	64	25	FSPN, Shift, Substitution, Permutation, Rotation, Counter XOR	[32]
7	SIT Muhammad et al. 2017	64	64	5	FSPN, Add Round Key, F-function, F-function Matrix	[52]
8	SFN Li et al. 2018	96	64	20	FSPN, Add Round Key, Substitution, Permutation, XOR	[13]
9	FEW ManojKumar et al. 2019	80/128	64	32	FSPN, Add Round Key, Substitution, Permutation, Rotation, Counter XOR	[9]
10	Shadow Ying et al. 2020	128	64	32	ARX, AND, XOR, Add Round Key, Add Constant	[14]
11	SLIM Bassam et al. 2020	80	32	32	SPN, Substitution, Permutation, Add Round Key, Shift, XOR	[10]

12	LBC-IoT Rabie et al. 2021	80	32	32	SPN, Substitution, Permutation, Add Round Key, Shift, XOR	[11]
13	RBFK Rana et al 2023	64/128	64	5	Custom, Add Round Key, G-function, Swap, Add Constant	[51]
14	SLA Nahla et al 2023	80/128	64	16	SPN, Add Round Key, Substitution, Permutation, Rotation, Counter XOR	[31]
15	Razor Dheeraj et al. 2024	128	64	32	SPN, Add Round Key, Substitution, Diffusion, Rotation, Counter XOR	[50]

Table 1 provides a comparative overview of 15 prominent LBCs proposed between 2006 and 2024. These ciphers have been widely recognized in the literature and evaluated based on traditional metrics such as Gate Equivalents (GE), latency, and memory usage. However, as detailed below, they share common design flaws that limit their applicability in dynamic, real-world IoT scenarios.

2.2. Critical Analysis of Existing Ciphers

This section describes the critical review of existing lightweight block ciphers with publications from 2006 to 2024.

2.2.1. Lack of Dynamic Features

The most significant limitation of existing ciphers is their lack of dynamic adaptability in design. Nearly all ciphers, listed in Table 1, rely on static round functions, fixed key schedules, and rigid encryption structures, restricting their flexibility and resilience against evolving security threats [26] – [40]. Cryptanalysis [41]– [49] examined the impact of fixed round functions and static key schedules, highlighting their vulnerability to various attacks. While slight variations in round functions or key schedules exist, they do not introduce genuine adaptability or programmable behavior.

For Example:

- Fixed Round Functions: Ciphers such as [50] – [54] (RAZOR, PRESENT, HIGHT, and others) rely on static round functions with fixed bit shifts and substitutions. There is no provision for dynamic round adjustment based on input characteristics or security requirements.
- Static Key Schedules: Key schedules in most ciphers involve simple operations like fixed constant addition, round counters, or whitening keys. These approaches lack innovation and fail to enhance the unpredictability or flexibility of key generation.

2.2.2. Superficial Tweaks to Established Designs

Many ciphers listed are tweaked versions of earlier designs like PRESENT [1] and HIGHT [5]. For example:

- Piccolo (2011) [3] and LBlock (2011) [53] closely resemble the structure of PRESENT [1], with minor modifications to round functions and key scheduling.
- Although they are new ideas, SLIM (2020) [10] and LBC-IoT (2021) [11] are based on previous SPN-based systems and do not significantly improve scalability or adaptability.

Thus, the trend of superficial tweaks restricts genuine innovation, as these ciphers are optimized for specific metrics like GE or latency while overlooking broader security and operational requirements.

2.2.3. Irrelevance of Traditional Metrics

The reliance on traditional metrics such as GE and latency for cipher evaluation has led to unscientific and unfair comparisons. While these metrics provide insights into hardware efficiency, they fail to capture critical aspects like scalability, adaptability, and resilience to cryptanalysis. Consequently, many ciphers that excel in GE or latency are incorrectly deemed superior despite their inherent design limitations.

For example, due to the low GE count, the cipher SIT [52] was declared superior. However, when evaluated using the proposed context-aware metrics, it becomes evident that SIT [52] lacks scalability, adaptability, and dynamic functionality, making it unsuitable for real-world IoT applications.

2.3. Discussion

The related work on existing lightweight block ciphers suggests a clear pattern of reliance on static designs and superficial tweaks. Traditional evaluation metrics fail to capture the true efficacy of these ciphers, leading to unscientific comparisons and limiting innovation in the field. The NDN cipher addresses these gaps and sets a new benchmark for lightweight cryptographic solutions by introducing dynamic adaptability, programmable key schedules, and context-aware design. The subsequent sections will detail how NDN outperforms existing ciphers by adhering to a scientifically rigorous, application-centric evaluation framework.

3. DESIGN ASPECTS OF NDN

The proposed ultra-lightweight block cipher, NDN, features a uniquely structured encryption, decryption, and round-key generation schedule. It utilizes a 64-bit block size and supports 80-bit and 128-bit keys with 12 and 18 iterative rounds, respectively. The notations used in this article are in table 2 as follows.

Table 2 Notations

Symbol	Description
#	Number
⊕	Bitwise XOR
<<	Circular shift left
>>	Circular right shift
ǁ	Concatenation
ρ	Round count
i	Round index
j	Column index
~	Complement

These notations are consistently applied throughout the encryption, decryption, and key generation processes to ensure a compact and uniform representation of operations.

3.1. Specifications

Block size: 64-bit

Key size: 80-bit and 128 bits

Input size: 64-bit (Plaintext)

Output size: 64-bit (Ciphertext)

Iterative Rounds: 12 (NDN-80) and 18 (NDN-128)

Structure: Hybrid Feistel and SPN

Versions: NDN-80 and NDN-128

A 64-bit block size is selected to achieve an optimal balance between security and efficiency, ensuring the cipher remains lightweight while providing adequate strength for IoT applications. The number of rounds—12 for NDN-80 and 18 for NDN-128—is determined through empirical testing and cryptanalysis, optimizing the trade-off between performance and security.

The hybrid SPN-Feistel structure combines the diffusion properties of Feistel networks with the confusion capabilities of SPN layers, enhancing resistance to differential and linear cryptanalysis. This dual-version approach provides adaptability for applications with varying security and performance requirements:

- NDN-80 is ideal for lightweight IoT use cases like environmental monitoring and sensor networks.
- NDN-128 is for high-security applications, such as healthcare systems and smart grids.

3.2. Encryption/Decryption Schedule

The encryption/decryption schedule employs substitution, a bit transformation based on a primitive polynomial of degree 4, a complement, and linear diffusion functions.

3.2.1. Preliminaries

Substitution layer: The F-function accepts a 16-bit input, structured as four 4-bit blocks, and applies nibble substitution using Table 3. Table 3 [11] presents the S-box used in this study, which maintains consistency with established cryptographic properties.

Table 3 S-box [11]

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

Here, the S-box is applied four times in parallel as a non-linear operation. The concatenation of four substituted nibbles yields a 16-bit output as in equation (1).

$$S(X) = \{S(X_0) \parallel S(X_1) \parallel S(X_2) \parallel S(X_3)\} \quad (1)$$

For example,

Let X = $\{0000,0011,1010,1110\}$, then

S(X) = $\{S\ (0000), S\ (0011), S\ (1010), S\ (1110)\}$,

S(X) = $\{0000 \parallel 1101 \parallel 0010 \parallel 1011\}$

Each 4-bit nibble of the 16-bit data is substituted independently using the 4x4 S-box. This parallel substitution operation introduces confusion, significantly enhancing the cipher's security against differential cryptanalysis.

Bit transformation layer: A primitive polynomial of degree 4, described by equation (2), is the basis for generating bit transformations.

$$x^4 \oplus x^3 \oplus 1 \quad (2)$$

There are 2^4 options to choose from for bit transformation. Tables 4, 5, and 6 are the three selected options. Tables 4 and 5 are employed in the F function of the encryption/decryption schedule, whereas Table 6 is in round key generation.

Table 4 Bit transformation (BT1)

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

Table 5 Bit transformation (BT2)

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

Table 6 Bit transformation (BT3)

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

Equation (3) defines the bit transformation function.

$$BT(X) = \begin{cases} BT1[[S(X)] \; if \; K_{i-j(0)} = 0 \\ BT2[S(X)] \; if \;\; K_{i-j(0)} = 1 \end{cases} \qquad (3)$$

Where: the bit transformations BT1 and BT2 correspond to Table 4 and Table 5. Letters i and j represent the round and the sub-key number (j=1 or 2, since there are two keys per round); the (0) subscript indicates the 0th bit of the respective subkey. For example, $K_{1-1(0)}$ represents the 0th bit of the first subkey of round one.

For example:

Let $\mathrm{X} = \{x_{15} \dots \dots . x_0\} = \{0000\ 1101\ 0010\ 1011\}$,

then using Table 4 and Table 5

$\mathrm{BT1(X)} = \{x_{15} \dots \dots . x_0\} = \{1100\ 0011\ 1001\ 0001\}$

$\mathrm{BT2(X)} = \{x_{15} \dots \dots . x_0\} = \{1011\ 0000\ 1110\ 0100\}$

The bit transformation layer utilizes a degree-4 polynomial, $x^4 \oplus x^3 \oplus 1$, which enhances complexity and strengthens resistance against differential attacks. Tables 4, 5, and 6 are selected from 24 different bit transformation functions, enhancing non-linearity and increasing the avalanche effect.

Diffusion layer: Equation (4) describes the diffusion function with a branch size of 5 (number).

$$\mathrm{f(X)} = \mathrm{X} \oplus (\mathrm{X} \ll 1) \oplus (\mathrm{X} \ll 5) \oplus (\mathrm{Y} \ll 9) \oplus (\mathrm{X} \ll 12) \qquad (4)$$

For example:

let $\mathrm{X} = \{1000\ 0110\ 0001\ 0000\}$, then

$\mathrm{f(X)} = \{0110\ 0001\ 0100\ 1100\}$

The diffusion function ensures that small changes in the input (e.g., flipping a single bit) result in significant changes in the output. XOR'ing the input with shifted versions of itself ensures that the cipher exhibits high avalanche behavior and is essential for resisting differential and linear cryptanalysis.

Complement Function: Equation (5) represents the complement function.

$$C(X) = \sim(X) \qquad (5)$$

For example,

Let $X = 0110\ 0001\ 0100\ 1100$, then

$C(X) = \sim X = 1001\ 1110\ 1011\ 0011$

An additional non-linearity is added to the data by a complement function. The complement function is part of branches 2 and 3 of the encryption/decryption schedule.

F-Function: The F function is the core part of the cipher/decipher schedule in the proposed work. The F function operates on a 16-bit data segment and produces an equivalent segment as output. The structure of the F function is in Figure 1.

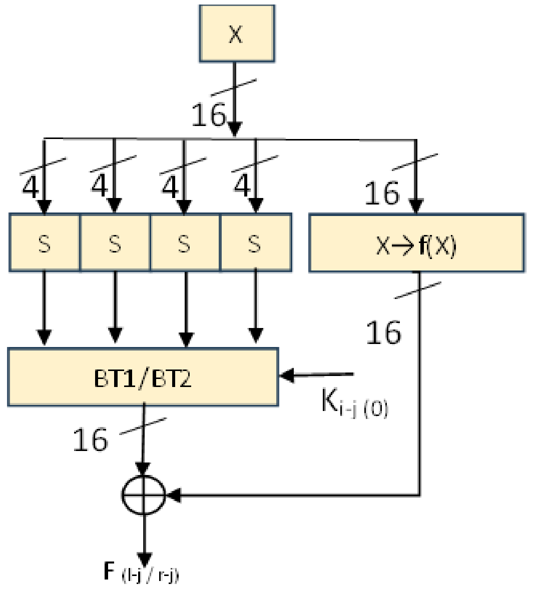


Figure 1 F Function

Equation (6) describes the F function.

$$F_{(l,i/r,i)} = f(X) \oplus \begin{cases} BT1[[S(X)] \; if \; K_{i-j(0)} = 0 \\ BT2[S(X)] \; if \;\; K_{i-j(0)} = 1 \end{cases} \qquad (6)$$

Here, *l* and *r* indicate the left and right-side F functions respectively. The variable *i* represents the round number, while *j* (where *j* = 1 or 2) denotes the sub-key number. The subscript (0) indicates the 0th bit of the respective sub-key. $S(X)$ is the substitution of the input X using the S-box.

BT1 and BT2 are the bit transformation results from Tables 4 and 5, respectively. The value of the 0th bit from the round key K_{i-j} (0) determines whether Table 4 or Table 5 permutes the substituted data *S(X)*. The $f(X)$ is the diffusion layer, The f(X) is the diffusion layer, where circular shifts and XOR operations dictate the mixing of data bits.

3.2.2. Encryption/Decryption Schedule

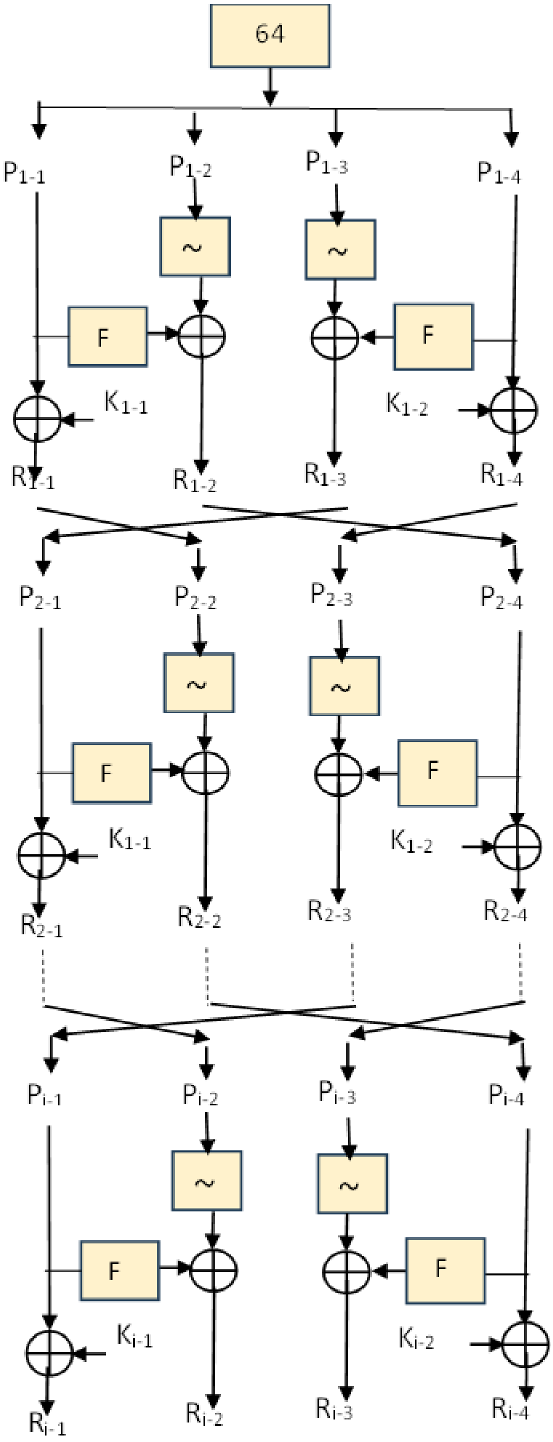


Figure 2 Encryption/Decryption Schedule

The proposed encryption process transforms a 64-bit plaintext ‘P’ into a 64-bit ciphertext ‘C’ over ‘ρ’ iterative rounds. Each round employs two 16-bit round keys derived from the user-supplied key (80-bit or 128-bit) and processes the data using the proposed F-function.

Input: A 64-bit plaintext and the user-supplied key (80-bit or 128-bit).

Output: A 64-bit ciphertext after ρ rounds of encryption.

Round Keys: Each round uses two 16-bit round keys derived from the user key.

Execution Flow: Each round applies the F-functions to left and right segments, XORs with keys, and swaps positions to ensure diffusion.

Equation (7) describes the encryption process.

$$Enc_i: \begin{Bmatrix} \{0,1\}^{64} \times \{\{0,1\}^{16}\}^2 \to \{0,1\}^{64} \\ \left(P_{64}, K_{i-1(16)}, K_{i-2(16)} \to C_{64}\right) \end{Bmatrix} \quad (7)$$

Where: P_{64} is the plain text, $K_{i-1(16)}$ and $K_{i-2(16)}$ are the two 16-bit round keys for the ith round.

C_{64} is the ciphertext.

```
Input: P (64-bit Plaintext), K (user key 80-bit or 128-bit), ρ (number of rounds).
Output: C (64-bit Ciphertext).
Begin:
1. Divide P into four 16-bit segments: $P_{i-1}$, $P_{i-2}$, $P_{i-3}$, and Pi-4
2. For i = 1 to ρ do:
a. Compute:
    $R_{i-1} = P_{i-1} \oplus K_{i-1}$
    $R_{i-2} = F_{l-i}(P_{i-1}) \oplus \sim(P_{i-2})$
    $R_{i-3} = F_{r-i}(P_{i-4}) \oplus \sim(P_{i-3})$
    $R_{i-4} = P_{i-4} \oplus K_{i-2}$
b. If i < ρ (intermediate rounds):
    Swap the segments
    $P_{n-1} = R_{i-3}$
    $P_{n-2} = R_{i-1}$
    $P_{n-3} = R_{i-4}$
    $P_{n-4} = R_{i-2}$
c. If i = = ρ (final round):
    Concatenate segments:
    $C = R_{i-1} \parallel R_{i-2} \parallel R_{i-3} \parallel R_{i-4}$
```

End:

Algorithm 1 Encryption Process

3.2.3. Encryption Execution Process

Initialization: The plaintext P is divided into four segments of 16 bits each and dented by P_{i-1}, P_{i-2}, P_{i-3}, and $P_{i-4.}$

3.2.3.1. Round Operations:

1. Each round uses two 16-bit round keys $K_{i-1(16)}, K_{i-2(16)}$ derived from user keys 80-bit or 128-bit.
2. The left and right F-functions operate on P_{i-1} and Pi-4, respectively, incorporating substitution, bit transformation, and diffusion.
3. The outputs of the F-functions are XORed with the complemented values of P_{i-2} and P_{i-3}.
4. The diffusion across rounds is ensured by swapping segments.

Final Round: After ρ rounds, the concatenated segments produce the final ciphertext C. [Refer to Algorithm 1 above]

Decryption Process: The decryption process is the reverse of the encryption process, with round keys applied in reverse order. The intermediate steps are identical, ensuring simplicity and efficiency. The last round output is un-swapped.

3.3. Round Key Generation Schedule

The round key generation schedule transforms the user-supplied key into round keys with enhanced non-linearity and diffusion. The 80-bit and 128-bit schedules leverage a combination of bit transformation (Table 5), complement, and permutation (16-point radix-4 DFFT). While the steps are similar, the number of iterations and final round keys differ, ensuring scalability for varying security requirements.

Unlike traditional key schedules, which often rely solely on permutations or transformations, the proposed schedule integrates a unique combination of primitive polynomial-based bit transformations, complement operations, and neural network-inspired DFFT permutations. It ensures high non-linearity and diffusion, making the generated round keys resilient to advanced cryptanalytic techniques.

Input: 80-bit or 128-bit user-supplied key.

Output: A sequence of round keys (24 round keys /80-bit user key and 40 round keys /128-bit user key).

Critical Components:

- Bit Transformation: The bit transformation leverages Table 5, derived from the primitive polynomial $x^4 \oplus x^3 \oplus 1$. This operation introduces non-linearity by mapping input bits $k_0 \ldots k_{15}$ to their transformed values. For example, if the input is {0001, 0110, 1000, 1011}, the corresponding transformed output is {1101, 0010, 1001, 0110} (see Table 5 for mapping).
- Complement Function: Adds further non-linearity to selected nibbles.
- Permutation: After the initial transformations and complement steps, the key bits undergo permutation using a neural network-inspired 16-point radix-4 DFFT. This permutation is described by (Equation 8), and the mapping: If $X = \{x_{15}, \ldots \ldots \ldots \ldots \ldots., x_0,\}$, then

$$\mathrm{P(X)} = \{x_{15}, x_{11}, x_7, x_3, x_{14}, x_{10}, x_6, x_2, x_{13}, x_9, x_5, x_1, x_{12}, x_8, x_4, x_0,\} \tag{8}$$

For example, let X = {0000 0000 0000 0100}, then

$P\,(X) = \{0000\ 0001\ 0000\ 0000\}$

The mathematical complexity of solving a single 16-point Radix 4 DFFT involves 2 x 4 x 3 complex multiplications and 2 x 4 x 8 additions to solve. Tables 7 and 8 are prepared based on equation (8).

3.3.1. 80-Bit Key Schedule

User-Supplied Key:

The 80-bit user-supplied key UK_{80} is represented as:

$UK_{80} = k_{79} \parallel k_{78} \parallel k_{77} \ldots \ldots . \parallel k_0$

Segmentation:

The key is divided into five groups of 16 bits each:

$(k_{79} \ldots k_{64})$, $(k_{63} \ldots k_{48})$, $(k_{47} \ldots k_{32})$, $(k_{31} \ldots k_{16})$, and $(k_{15} \ldots k_0)$.

Bit Transformation:

Apply the bit transformation using Table 5 to the first 16 bits $(k_{15} \ldots k_0)$, as defined by $x^4 \oplus x^3 \oplus 1$.

Complement Function: Apply the complement function to specific nibbles shown below.

$$\begin{cases} Nibble(k_{20}, .., k_{23}) = \sim Nibble(k_{20}, .., k_{23}) \\ Nibble(k_{36}, .., k_{39}) = \sim Nibble(k_{36}, .., k_{39}) \\ Nibble(k_{52}, .., k_{55}) = \sim Nibble(k_{52}, .., k_{55}) \\ Nibble(k_{68}, .., k_{71}) = \sim Nibble(k_{68}, .., k_{71}) \end{cases}$$

$$\begin{cases} Nibble(k_{28}, .., k_{31}) = \sim Nibble(k_{28}, .., k_{31}) \\ Nibble(k_{44}, .., k_{47}) = \sim Nibble(k_{44}, .., k_{47}) \\ Nibble(k_{60}, .., k_{63}) = \sim Nibble(k_{60}, .., k_{63}) \\ Nibble(k_{76}, .., k_{79}) = \sim Nibble(k_{76}, .., k_{79}) \end{cases}$$

Permutation:

The rearrangement of key bits follows the values specified in Table 7, enhancing diffusion after applying segmentation, bit transformation, and complement functions.

Table 7 Rearrangement of 80-Bit Key Bits

k_i	k_p	k_i	k_p	k_i	k_p	k_i	k_p	k_i	k_p
0	64	16	48	32	32	48	16	64	0
1	68	17	52	33	36	49	20	65	4
2	72	18	56	34	40	50	24	66	8
3	76	19	60	35	44	51	28	67	12
4	65	20	49	36	33	52	17	68	1
5	69	21	53	37	37	53	21	69	5
6	73	22	57	38	41	54	25	70	9
7	77	23	61	39	45	55	29	71	13
8	66	24	50	40	34	56	18	72	2
9	70	25	54	41	38	57	22	73	6
10	74	26	58	42	42	58	26	74	10
11	78	27	62	43	46	59	30	75	14
12	67	28	51	44	35	60	19	76	3
13	71	29	55	45	39	61	23	77	7
14	75	30	59	46	43	62	27	78	11
15	79	31	63	47	47	63	31	79	15

Round keys:

At the end of the process, generate four 16-bit round keys by concatenating segments:

$k_{i-1} = (k_{79} \ldots k_{64}) = (k_{15} \ldots k_0), i = 1$,

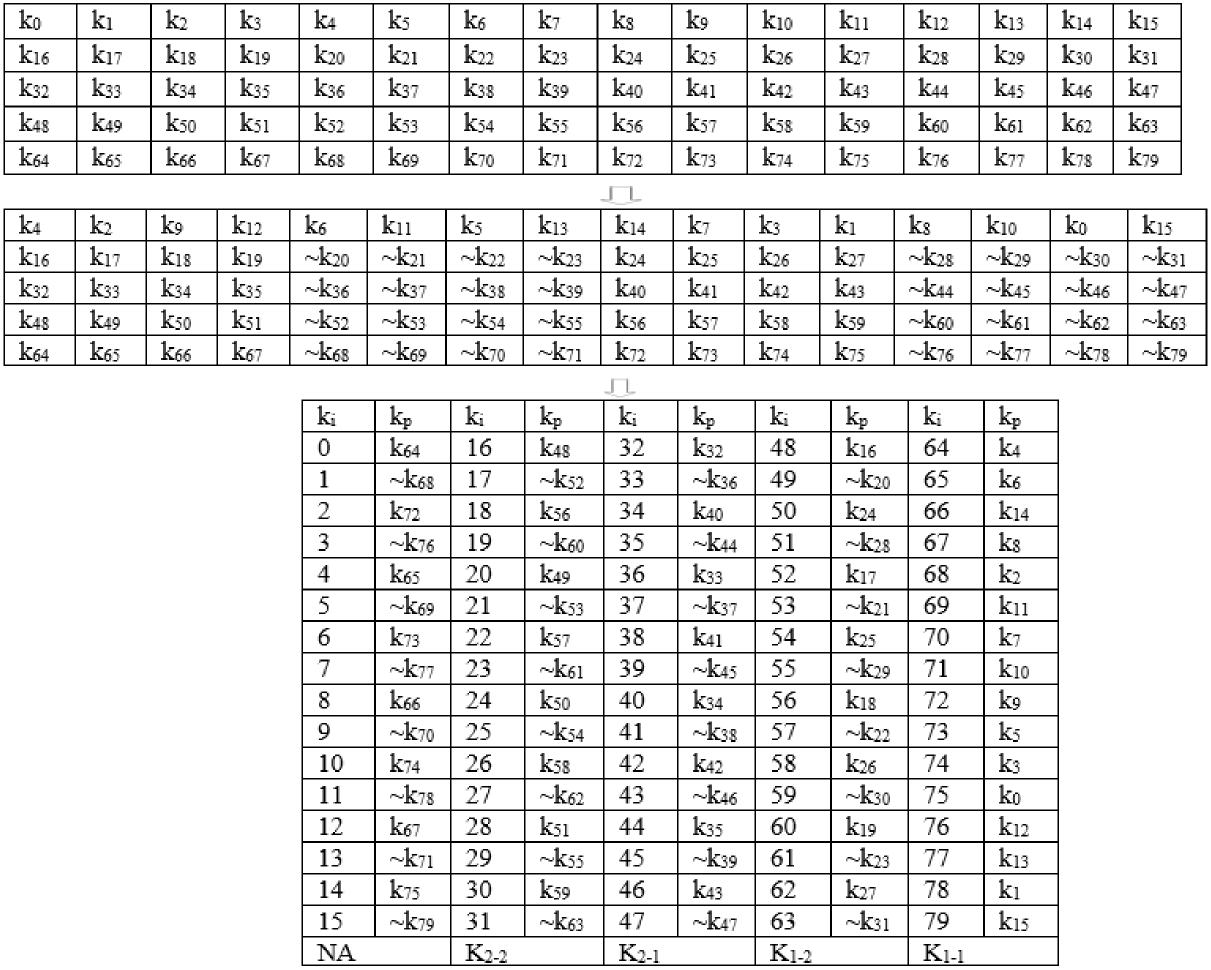
$k_{i-2} = (k_{63} \ldots k_{48}) = (k_{31} \ldots k_{16}), i = 1$,

$k_{i-1} = (k_{47} \ldots k_{32}) = (k_{47} \ldots k_{32}), i = 2$, and

$k_{i-2} = (k_{31} \ldots k_{16}) = (k_{63} \ldots k_{48}), i = 2$.

Iteration:

Rotate the key $UK_{80} \gg 9$ and repeat for six iterations to produce 24 round keys. Figure 3 depicts the round key generation using an 80-bit user key.



k_0	k_1	k_2	k_3	k_4	k_5	k_6	k_7	k_8	k_9	k_{10}	k_{11}	k_{12}	k_{13}	k_{14}	k_{15}
k_{16}	k_{17}	k_{18}	k_{19}	k_{20}	k_{21}	k_{22}	k_{23}	k_{24}	k_{25}	k_{26}	k_{27}	k_{28}	k_{29}	k_{30}	k_{31}
k_{32}	k_{33}	k_{34}	k_{35}	k_{36}	k_{37}	k_{38}	k_{39}	k_{40}	k_{41}	k_{42}	k_{43}	k_{44}	k_{45}	k_{46}	k_{47}
k_{48}	k_{49}	k_{50}	k_{51}	k_{52}	k_{53}	k_{54}	k_{55}	k_{56}	k_{57}	k_{58}	k_{59}	k_{60}	k_{61}	k_{62}	k_{63}
k_{64}	k_{65}	k_{66}	k_{67}	k_{68}	k_{69}	k_{70}	k_{71}	k_{72}	k_{73}	k_{74}	k_{75}	k_{76}	k_{77}	k_{78}	k_{79}

k_4	k_2	k_9	k_{12}	k_6	k_{11}	k_5	k_{13}	k_{14}	k_7	k_3	k_1	k_8	k_{10}	k_0	k_{15}
k_{16}	k_{17}	k_{18}	k_{19}	$\sim k_{20}$	$\sim k_{21}$	$\sim k_{22}$	$\sim k_{23}$	k_{24}	k_{25}	k_{26}	k_{27}	$\sim k_{28}$	$\sim k_{29}$	$\sim k_{30}$	$\sim k_{31}$
k_{32}	k_{33}	k_{34}	k_{35}	$\sim k_{36}$	$\sim k_{37}$	$\sim k_{38}$	$\sim k_{39}$	k_{40}	k_{41}	k_{42}	k_{43}	$\sim k_{44}$	$\sim k_{45}$	$\sim k_{46}$	$\sim k_{47}$
k_{48}	k_{49}	k_{50}	k_{51}	$\sim k_{52}$	$\sim k_{53}$	$\sim k_{54}$	$\sim k_{55}$	k_{56}	k_{57}	k_{58}	k_{59}	$\sim k_{60}$	$\sim k_{61}$	$\sim k_{62}$	$\sim k_{63}$
k_{64}	k_{65}	k_{66}	k_{67}	$\sim k_{68}$	$\sim k_{69}$	$\sim k_{70}$	$\sim k_{71}$	k_{72}	k_{73}	k_{74}	k_{75}	$\sim k_{76}$	$\sim k_{77}$	$\sim k_{78}$	$\sim k_{79}$

k_i	k_p	k_i	k_p	k_i	k_p	k_i	k_p	k_i	k_p
0	k_{64}	16	k_{48}	32	k_{32}	48	k_{16}	64	k_4
1	$\sim k_{68}$	17	$\sim k_{52}$	33	$\sim k_{36}$	49	$\sim k_{20}$	65	k_6
2	k_{72}	18	k_{56}	34	k_{40}	50	k_{24}	66	k_{14}
3	$\sim k_{76}$	19	$\sim k_{60}$	35	$\sim k_{44}$	51	$\sim k_{28}$	67	k_8
4	k_{65}	20	k_{49}	36	k_{33}	52	k_{17}	68	k_2
5	$\sim k_{69}$	21	$\sim k_{53}$	37	$\sim k_{37}$	53	$\sim k_{21}$	69	k_{11}
6	k_{73}	22	k_{57}	38	k_{41}	54	k_{25}	70	k_7
7	$\sim k_{77}$	23	$\sim k_{61}$	39	$\sim k_{45}$	55	$\sim k_{29}$	71	k_{10}
8	k_{66}	24	k_{50}	40	k_{34}	56	k_{18}	72	k_9
9	$\sim k_{70}$	25	$\sim k_{54}$	41	$\sim k_{38}$	57	$\sim k_{22}$	73	k_5
10	k_{74}	26	k_{58}	42	k_{42}	58	k_{26}	74	k_3
11	$\sim k_{78}$	27	$\sim k_{62}$	43	$\sim k_{46}$	59	$\sim k_{30}$	75	k_0
12	k_{67}	28	k_{51}	44	k_{35}	60	k_{19}	76	k_{12}
13	$\sim k_{71}$	29	$\sim k_{55}$	45	$\sim k_{39}$	61	$\sim k_{23}$	77	k_{13}
14	k_{75}	30	k_{59}	46	k_{43}	62	k_{27}	78	k_1
15	$\sim k_{79}$	31	$\sim k_{63}$	47	$\sim k_{47}$	63	$\sim k_{31}$	79	k_{15}
NA		K_{2-2}		K_{2-1}		K_{1-2}		K_{1-1}	

Figure 3 Illustrates the Complete Round Key Generation Process for an 80-Bit User-Supplied Key

3.3.2. 128-Bit Key Schedule

User-Supplied Key:

The 128-bit user-supplied key UK_{128} is represented as:

$UK_{128} = k_{127} \parallel k_{78} \parallel k_{77} \dots \dots . \parallel k_0$

Segmentation:

The key is divided into eight groups of 16 bits each:

$(k_{127} \dots k_{112})$, $(k_{111} \dots k_{96})$, $(k_{95} \dots k_{80})$, $(k_{79} \dots k_{64})$, $(k_{63} \dots k_{48})$, $(k_{47} \dots k_{32})$, $(k_{31} \dots k_{16})$, and $(k_{15} \dots k_0)$.

Bit Transformation:

Apply the bit transformation using Table 5 to the first 16 bits $(k_{15} \dots k_0)$, as defined by $x^4 \oplus x^3 \oplus 1$.

Complement Function:

Apply the complement function to specific nibbles shown below.

$$\begin{cases} Nibble(k_{20},..,k_{23}) = \sim Nibble(k_{20},..,k_{23}) \\ Nibble(k_{36},..,k_{39}) = \sim Nibble(k_{36},..,k_{39}) \\ Nibble(k_{52},..,k_{55}) = \sim Nibble(k_{52},..,k_{55}) \\ Nibble(k_{68},..,k_{71}) = \sim Nibble(k_{68},..,k_{71}) \end{cases}$$

$$\begin{cases} Nibble(k_{28},..,k_{31}) = \sim Nibble(k_{28},..,k_{31}) \\ Nibble(k_{44},..,k_{47}) = \sim Nibble(k_{44},..,k_{47}) \\ Nibble(k_{60},..,k_{63}) = \sim Nibble(k_{60},..,k_{63}) \\ Nibble(k_{76},..,k_{79}) = \sim Nibble(k_{76},..,k_{79}) \end{cases}$$

Permutation:

Rearrange the key bits based on Table 8, enhancing diffusion after applying segmentation, bit transformation, and complement functions.

Table 8 Rearrangement of 128-Bit Key Bits (Table 8 is an Original Contribution Developed in this Work.)

k_i	k_p	k_i	k_p	k_i	k_p	k_i	k_p	k_i	k_p	k_i	k_p	k_i	k_p	k_i	k_p
0	112	16	96	32	80	48	67	64	48	80	32	96	16	112	0
1	116	17	100	33	84	49	68	65	52	81	36	97	20	113	4
2	120	18	104	34	88	50	72	66	56	82	40	98	24	114	8
3	124	19	108	35	92	51	76	67	60	83	44	99	28	115	12
4	103	20	97	36	81	52	65	68	49	84	33	100	17	116	1
5	117	21	101	37	85	53	69	69	53	85	37	101	21	117	5
6	121	22	105	38	89	54	73	70	57	86	41	102	25	118	9
7	125	23	109	39	93	55	77	71	61	87	45	103	29	119	13
8	114	24	98	40	82	56	66	72	50	88	34	104	18	120	2
9	118	25	102	41	86	57	70	73	54	89	38	105	22	121	6
10	122	26	106	42	90	58	79	74	58	90	42	106	26	122	10
11	126	27	110	43	94	59	78	75	62	91	46	107	30	123	14
12	115	28	99	44	83	60	64	76	51	92	35	108	19	124	3
13	119	29	103	45	87	61	71	77	55	93	39	109	23	125	7
14	123	30	107	46	91	62	75	78	59	94	43	110	27	126	11
15	127	31	111	47	95	63	79	79	63	95	47	111	31	127	15

Round keys:

At the end of the process, generate eight 16-bit round keys by concatenating segments:

$k_{i-1} = (k_{127} \dots k_{112}) = (k_{15} \dots k_0)$i=1,
$k_{i-2} = (k_{111} \dots k_{96}) = (k_{31} \dots k_{16})$,i=1
$k_{i-1} = (k_{95} \dots k_{80}) = (k_{47} \dots k_{32})$,i=2,
$k_{i-2} = (k_{79} \dots k_{64}) = (k_{63} \dots k_{48}), i = 2$

$k_{i-1} = (k_{63} \dots k_{48}) = (k_{79} \dots k_{64})$i=3,
$k_{i-2} = (k_{47} \dots k_{32}) = (k_{95} \dots k_{80})$,i=3
$k_{i-1} = (k_{31} \dots k_{16}) = (k_{111} \dots k_{96})$i=4, and
$k_{i-2} = (k_{15} \dots k_{0}) = (k_{127} \dots k_{112}), i = 4$

Iteration:

Rotate the key $UK_{128} \gg 15$ and repeat for five iterations to produce 40 (5 x 8 = 40) round keys. The proposed round key generation schedule ensures high non-linearity and diffusion through a unique combination of bit transformations, complement functions, and neural network-inspired 16-point radix-4 DFFT permutations. By leveraging a primitive polynomial for bit transformations and incorporating complement operations, the design introduces robust resistance to differential and linear cryptanalysis. The neural network-based permutation further enhances diffusion, ensuring that even minor changes in the user-supplied key propagate widely across the round keys. The scalable approach supports 80-bit and 128-bit keys, addressing diverse security and performance requirements. These innovations provide a strong cryptographic foundation, making the NDN cipher suitable for securing resource-constrained IoT devices.

3.4. Design Implications of NDN

The design of the NDN cipher incorporates several innovative elements that enhance its cryptographic strength and practical applicability in diverse IoT environments. This section highlights key design implications of NDN to distinguish it from conventional lightweight block ciphers.

3.4.1. Dynamic Round Function

The encryption/decryption schedule of NDN incorporates a dynamic round function where bit transformations are selected based on key bits. This dynamic variability offers a significant advantage over traditional designs, where static round functions are employed. The key-driven selection process introduces unpredictability, preventing attackers from exploiting fixed patterns during cryptanalysis. Additionally, a primitive polynomial of degree 4 and a diffusion function with a branch number of 5 ensures a strong avalanche effect, where a minor change in the input results in widespread changes in the output. By integrating dynamic elements in the round function, NDN achieves enhanced security, making it resilient to attacks such as differential cryptanalysis, where attackers attempt to exploit predictable data patterns.

3.4.2. Advanced Round Key Generation Schedule

The NDN cipher employs a novel round key generation schedule that significantly improves key randomness and non-linearity. Unlike traditional key schedules that rely on fixed permutations and simple transformations, NDN integrates:

- Primitive polynomial-based bit transformations: The bit transformations derived from a polynomial $x^4 \oplus x^3 \oplus 1$ ensure high non-linearity, making it difficult for attackers to predict key patterns.
- Complement functions: A complement function is introduced at selective positions to enhance the randomness of key bits and diffusion of key material across rounds.
- Neural network-inspired DFFT permutations: The 16-point radix-4 DFFT permutation step ensures that minor changes in the user key propagate widely across the round keys. It results in a high diffusion, providing robust resistance against linear and differential cryptanalysis. This advanced approach ensures the round keys maintain randomness at a high degree across all iterations, making the cipher more secure than designs relying solely on conventional key scheduling techniques.

3.4.3. Efficiency and Applicability

The lightweight nature of NDN's design, combined with dual key-size support (80-bit and 128-bit), ensures its applicability across a wide range of IoT applications. Key design considerations include:

- NDN-80: Optimized for low-power IoT devices, such as environmental sensors and medical wearables, where energy efficiency and low hardware overhead are critical.
- NDN-128: Designed for high-security applications, such as smart grids and connected electric vehicles, where enhanced cryptographic strength is required.

The dual-version approach provides flexibility, enabling developers to choose the appropriate version for specific IoT deployment accordingly.

Conclusion:

The proposed NDN cipher represents a significant advancement in the design of lightweight cryptographic solutions for IoT applications. By integrating dynamic round functions, advanced round key generation schedules, and lightweight cryptographic operations, NDN addresses key challenges in IoT security, including adaptability, scalability, and resistance to cryptanalytic attacks. The dual-version design of NDN (NDN-80 and NDN-128) offers flexibility for various IoT domains, from low-power applications to high-security infrastructures. These design choices make NDN a robust, adaptable, and efficient cryptographic solution suitable for securing next-generation IoT devices and systems.

3.5. Implementation Strategies

This section describes the implementation aspects of the proposed NDN as hardware resource requirements of the proposed lightweight block cipher are analyzed to ensure optimal integration with various hardware platforms, including FPGA, ASIC, CPLD, and PLD technologies. This analysis is essential for securing resource-constrained environments such as IoT devices/systems by balancing cryptographic robustness and resource efficiency.

The NDN cipher employs a two-branch round function structure, depicted in Figure 4, where each round executes this structure twice to achieve high non-linearity and diffusion with minimal resource consumption.

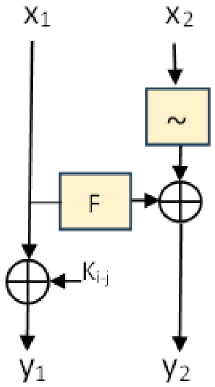


Figure 4 Two Branch Structure

The following equation describes the two-branch structure of NDN.

$$\begin{cases} y_1 = x_1 \oplus k_{i-j} \\ y_2 = F(x_1) \oplus \sim(x_2) \end{cases}$$

Where:

- i denotes the round number,
- j represents the key number (1 or 2),
- $F(x_1)$ is the round function applied on x_1, executed as per Equation (6), and
- $\sim(x_2)$ denotes the complement of x_2.

This design ensures efficient hardware utilization while maintaining high security through robust cryptographic primitives, such as dynamic bit transformations and a complement function. The dual-execution round function minimizes hardware complexity by reusing components, reducing the overall gate count while delivering high performance.

3.5.1. ASIC Implementation (0.13μm)

This subsection evaluates the hardware resource requirements of the NDN lightweight block cipher using 0.13μm ASIC technology. The analysis focuses on determining the gate equivalents (GEs) for fundamental components, ensuring a low-overhead design suitable for integration into resource-constrained IoT devices.

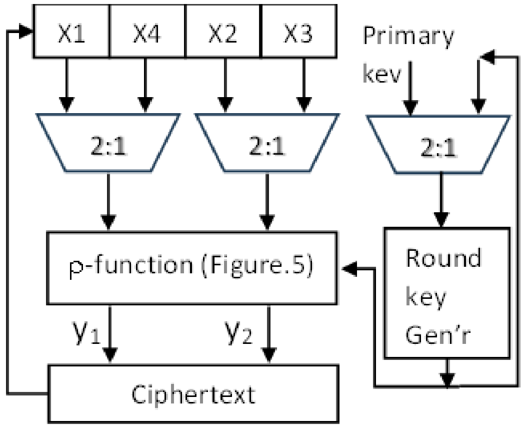


Figure 5 ASIC Internal Implementation

The core logic elements and their respective GE costs are as follows:

- D flip-flop: 4.25 GEs
- XOR/XNOR gate: 2 GEs
- 2:1 multiplexer (MUX): 2.25 GEs
- NOT gate: 0.75 GEs

For the NDN cipher, the 64-bit data block requires 272 GEs for data state storage, calculated as 64×4.25=272. The key state storage for the 80-bit and 128-bit versions requires 340 GEs and 544 GEs, respectively. The round function, designed to minimize hardware complexity, primarily uses XOR, NOT, multiplexers, and S-boxes. Figure 5 illustrates the internal implementation strategy of the NDN cipher using the two-branch structure in ASIC technology. This approach ensures an optimized hardware footprint by leveraging minimal gate equivalents for essential operations. Table 9 summarizes the area requirements for 80-bit and 128-bit key versions of NDN.

Table 9 Area Requirements for Hardware Implementation

Modules	Details	GE	TGE
Data register	64-bit data state and 32-bit output data state	408	552
Multiplexer	2:1 x 64	144	
Round function	S-box	52	152
	2:1 x 16	36	
	XOR	32	
	Diffusion	32	
80-bit key	80-bit state	340	364

schedule	NOT	24	
128-bit key schedule	128-bit state	544	568
	NOT	24	
80-bit NDN			1068
128-bit NDN			1272
GE→Gate Equivalent TGE→Total Gate Equivalent			

Encryption/Decryption Cycle Count:

A 2 × 16-bit data chunk is processed every round by the encryption and decryption procedures, which employ the two-branch structure. Each round of the cipher requires two clock cycles. The 12-round (NDN-80) requires 24 clock cycles, and the 18-round (NDN-128) requires 36 cycles. The cipher's efficient cycle count ensures fast throughput and low latency, making it suitable for real-time Internet of Things applications.

3.5.2. AVR RISC μC Implementation

The ATmega 328 microcontroller, based on the AVR RISC architecture, was utilized to evaluate the software performance of the proposed NDN cipher. The evaluation ensures that the cipher meets the resource constraints typical of IoT devices, particularly execution time and memory usage.

The total execution time is 99.12 µs, which includes key generation, encryption, and decryption. Execution time for encryption and decryption requires 40 µs each, and the remaining time for key generation. Table 10 provides a detailed breakdown of the time and space complexity metrics.

Table 10 Resource Requirements with AVR RISC μC

Metrics	Attribute	Value
Time complexity	Key generation	19.12 µs
	Encryption	40 µs
	Decryption	40 µs
Space complexity	Flash memory	2076kB
	SRAM	45B
	EEPROM	18B

This analysis underscores the NDN cipher’s efficiency in terms of time and memory. The compact design ensures minimal memory usage across Flash, SRAM, and EEPROM, making it ideal for resource-constrained IoT devices. Additionally, the fast execution time of 40 µs for encryption and decryption supports real-time operations, while the low overhead for key generation enhances the overall responsiveness of cryptographic processes.

4. SECURITY ANALYSIS

The security analysis of a cryptographic cipher critically evaluates its strength against various attacks, ensuring reliability for sensitive applications. This section examines the proposed NDN cipher's resistance to key cryptanalytic techniques, including differential and linear cryptanalysis. The analysis demonstrates the cipher's ability to maintain confidentiality and integrity, even under severe attack scenarios, through theoretical definitions, mathematical proofs, and empirical validation. The findings establish NDN as a robust solution for securing resource-constrained IoT devices in real-world applications.

4.1. Differential Cryptanalysis

Differential cryptanalysis is a crucial attack model that evaluates a cipher's resilience to differences in plaintext-ciphertext pairs. The unique design with dynamic bit transformations and a robust diffusion layer NDN demonstrates strong resistance to differential cryptanalysis. Differential cryptanalysis signifies the recovery of the secret key by analyzing the differences in plaintext and ciphertext pairs. The indicator of the cipher's resilience to differential attack is the total number of active S-boxes during the cipher's iterative rounds [39]-[40].

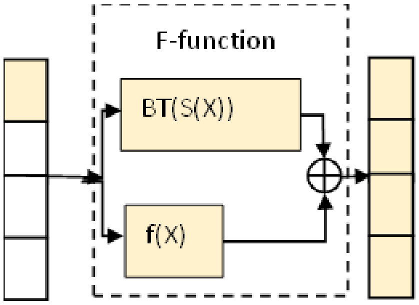


Figure 6 S-Box Transition

Definition 1. Differential Probability of an S-box (DPS) [10]

For an S-box S_i: $F_2^m \rightarrow F_2^{m'}$, where i = 1, 2, …, n, the differential probability represents the likelihood of a specific input difference Δx producing a given output difference Δy. This probability is computed by:

$$DP^{s_i}(\Delta x \rightarrow \Delta y) = \frac{\neq \{x \in F_2^m | S_i(x) \oplus S_i(x \oplus \triangle x) = \triangle y\}}{2^m}$$

The Δx and Δy represent the input and output differences, respectively.

Theorem 1: Upper Bound on Differential Probability (DP)

For any given S-box S_i, the DP satisfies the condition:

$DP^{s_i}(\Delta x \rightarrow \Delta y) \leq 1$

Additionally, for specific cases:

Additionally, for specific differential characteristics:

If $\Delta x = 0$ and $\Delta y = 0$, then

$DP^{s_i}(\Delta x \rightarrow \Delta y) = 1$

If $\Delta x \neq 0$, then

$DP^{s_i}(\Delta x \rightarrow \Delta y) = 0$

Theorem 2. Maximum Differential Probability (MDP) [10]

The MDP of an S-box S_i is defined by:

$$MDP^{Si} = \underset{\Delta x, \Delta y}{max} DP^{Si}(\Delta x \rightarrow \Delta y)$$

This value measures the S-box's resistance to differential attacks.

Definition 2: Active S-boxes

An S-box is active if at least one of its bit’s changes during the encryption. The number of active S-boxes reflects the cipher's resistance to differential attacks.

Theorem 3. (Minimum Active S-boxes) The dynamic bit transformation and diffusion functions ensure that any three consecutive rounds of the NDN cipher activate at least 15 S-boxes. This result is derived by applying the branch and bound technique considering differential propagation over multiple rounds. Figure 6 shows the differential branch number in F-function as 5.

Proof. The proof of this theorem is provided by analyzing the differential patterns (with the propagation of bits) over three iterative rounds with multiple input test vector instantiations. The proposed cipher has a 64-bit block and is segmented into four subblocks of 16-bits each. The encryption/decryption operation is on 16-bit subblocks with equal-sized subkey bits.

Let $\Delta \mathrm{P_i} = \Delta \mathrm{P_{i-1}} \parallel \Delta \mathrm{P_{i-2}} \parallel \Delta \mathrm{P_{i-3}} \parallel \Delta \mathrm{P_{i-4}}$ be the i^th round input difference characteristic with concatenated four subblocks.

1st input test vector instantiation.

Let $\Delta \mathrm{P_1} = [0x1000 \parallel 0x0000 \parallel 0x0000 \parallel 0x0000]$, such that $\Delta \mathrm{P_{i-1}} = 0x1000$, $\Delta \mathrm{P_{i-2}} = 0x0000$, $\Delta \mathrm{P_{i-3}} = 0x0000$, $\Delta \mathrm{P_{i-4}} = 0x0000$.

Each cipher round contains two F functions (refer to Figure 2). The input to the first F function on the left side (i.e., 1st branch) is $\Delta \mathrm{P_{i-1}} = 0x1000$, which contains one active S-box. In contrast, the input to the second F function on the right side (i.e., 4th branch) is $\Delta \mathrm{P_{i-4}} = 0x0000$, which contains zero active S-boxes. Thus, in the first round, there is one active F function with a single active S-box, meaning ΔP1=1+0. At the end of the second round, ΔP2=4+4, with each F function contributing four active S-boxes. Similarly, at the end of the third round, ΔP3=4+2, with the left-side and right-side F functions contributing 4 and 2 active S-boxes, respectively. Therefore, the minimum number of active S-boxes over three iterative rounds is: $\Delta \mathrm{P_1} + \Delta \mathrm{P_2} + \Delta \mathrm{P_3} = 1 + 8 + 6 = 15$.

2nd input test vector instantiation.

Let $\Delta \mathrm{P_1} = [0x1001 \parallel 0x0000 \parallel 0x0000 \parallel 0x0000]$, such that $\Delta \mathrm{P_{i-1}} = 0x1001$, $\Delta \mathrm{P_{i-2}} = 0x0000$, $\Delta \mathrm{P_{i-3}} = 0x0000$, $\Delta \mathrm{P_{i-4}} = 0x0000$.

The input to the first F function on the left side, (i.e., 1st branch) is $\Delta \mathrm{P_{i-1}} = 0x1001$, which contains two active S-boxes. On the other hand, the input to the second F function on the right side (i.e., 4th branch) is $\Delta \mathrm{P_{i-4}} = 0x0000$, which contains zero active S-boxes. Thus, in the first round, there is one active F function with two active S-boxes, resulting in $\Delta \mathrm{P_1} = 2 + 0$. At the end of the second round, $\Delta \mathrm{P_2} = 4 + 4$ with each F function contributing four active S-boxes. Similarly, at the end of the third round $\Delta \mathrm{P_3} = 3 + 2$ with left-side and right-side F functions contributing 3 and 2 active S-boxes. Therefore, the minimum number of active S-boxes over three iterative rounds is:$\Delta \mathrm{P_1} + \Delta \mathrm{P_2} + \Delta \mathrm{P_3} = 2 + 8 + 5 = 15$.

3rd input test vector instantiation.

Let $\Delta \mathrm{P_1} = [0x0001 \parallel 0x0000 \parallel 0x0000 \parallel 0x0001]$, such that $\Delta \mathrm{P_{i-1}} = 0x0001$, $\Delta \mathrm{P_{i-2}} = 0x0000$, $\Delta \mathrm{P_{i-3}} = 0x0000$, $\Delta \mathrm{P_{i-4}} = 0x0001$.

The input to the first F function on the left side (i.e., 1st branch) is $\Delta \mathrm{P_{i-1}} = 0x0001$, which contains one active S-box. Similarly, the input to the second F function on the right side (i.e., 4th branch) is $\Delta \mathrm{P_{i-4}} = 0x0001$, which contains one active S-box. Thus, in the first round, both active F functions contribute one active S-box each, resulting in $\Delta \mathrm{P_1} = 1 + 1$. At the end of the second round, $\Delta \mathrm{P_2} = 4 + 4$ with each F function contributing four active S-boxes. Similarly, at the end of the third round, $\Delta \mathrm{P_3} = 2 + 3$, with left-side and right-side F functions contributing 2 and 3 active S-boxes. Therefore, the minimum number of active S-boxes over three iterative rounds is: $\Delta \mathrm{P_1} + \Delta \mathrm{P_2} + \Delta \mathrm{P_3} = 2 + 8 + 5 = 15$.

4th input test vector instantiation.

Let $\Delta \mathrm{P_1} = [0x0000 \parallel 0x0001 \parallel 0x0000 \parallel 0x0000]$, such that $\Delta \mathrm{P_{i-1}} = 0x0000$, $\Delta \mathrm{P_{i-2}} = 0x0001$, $\Delta \mathrm{P_{i-3}} = 0x0000$, $\Delta \mathrm{P_{i-4}} = 0x0000$.

The input to the first F function on the left side (i.e., 1st branch) is $\Delta \mathrm{P_{i-1}} = 0x0000$, which contains zero active S-boxes. On the other hand, the input to the second F function on the right side (i.e. 4th branch) is $\Delta \mathrm{P_{i-4}} = 0x0000$, which contains zero active S-boxes. Thus, in the first round, there are no active F functions, contributing zero active S-boxes each, resulting in $\Delta \mathrm{P_1} = 0 + 0$. At the end of the second round, $\Delta \mathrm{P}_2 = 4 + 4$ with each F function contributing four active S-boxes. Similarly, at the end of the third round, $\Delta \mathrm{P}_3 = 4 + 3$ with left-side and right-side F functions contributing four and three active S-boxes. Therefore, the minimum number of active S-boxes over three iterative rounds is:

$$\Delta \mathrm{P}_1 + \Delta \mathrm{P}_2 + \Delta \mathrm{P}_3 = 0 + 8 + 7 = 15.$$

Based on test conditions and results, if ΔP1 contains more active nibbles, the minimum number of active S-boxes after three rounds remains at least 15, as observed in experiments. Consequently, any three-round differential characteristic of NDN will have at least 15 active S-boxes under all possible input conditions.

Lemma 1. MDP for 3 rounds

The three-round NDN cipher exhibits a maximum differential probability (MDP) of:

$$MDP^3 = (2^{-2})^{15} = 2^{-30}$$

This shows resistance to differential attacks due to the large number of active S-boxes per round.

Lemma 2. MDP for 12 Rounds

For a 12-round cipher, the maximum differential probability is:

$$MDP^{12} = (2^{-2})^{(5\times12)} = 2^{-120}$$

This demonstrates that NDN has an extremely low probability of successful differential attacks across multiple rounds.

4.2. Linear Cryptanalysis

Linear cryptanalysis, which seeks to exploit linear approximations between input and output, is similarly mitigated by NDN’s design. The non-linear S-box and high branch number in the diffusion layer contribute to its robustness. It is introduced by Matsui (1993).

Definition 2. Linear Probability of S-box (LPS)

The linear probability for a given S-box is given by:

$$LP^{Si}(\Gamma y \to \Gamma x) = \left(\frac{\neq \{x \in F_2^m | x \cdot \Gamma x = S_i(x) \cdot \Gamma y\}}{2^{m-1}} - 1\right)^2$$

Where $x \cdot \Gamma x$ represents the parity of the bitwise product of x and Γx.

Theorem 4. Maximum Linear Probability (MLP)

The maximum linear probability of an S-box $\mathrm{S_i}$ is defined as:

$$MLP^{Si} = \underset{\Gamma x, \Gamma y}{max} LP^{Si}(\Gamma y \to \Gamma x)$$

This quantifies the likelihood of a successful linear approximation being valid across multiple rounds.

Theorem 5. Linear Probability Bound

For any S-box $\mathrm{S_i}$ the differential probability satisfies:

$$LP^{Si}(\Gamma y \to \Gamma x) \leq\leq 1$$

Similar to differential probability, the bound ensures that the probability of any linear approximation holds across rounds.

Definition 3. Linear approximation refers to the method of approximating the relationship between plaintext, ciphertext, and subkey bits by constructing a linear expression with high probability.

Theorem 6. (Minimum Active S-boxes) A minimum of 15 S-boxes will be active in any three consecutive rounds for the NDN cipher, similar to the behavior in differential cryptanalysis. This result follows from the fact that differential and linear branch numbers are equivalent.

Proof: Since differential and linear cryptanalysis rely on similar principles, the count of active S-boxes in each case remains consistent. The theorem follows from the properties of the S-boxes and the cipher's round structure.

Lemma 3. MLP for 3 rounds

For a 3-round NDN cipher, the maximum linear probability (MLP) is:

$$MLP^3 = (2^{-2})^{15} = 2^{-30}$$

This low probability suggests that the cipher is resistant to linear cryptanalysis as well.

Lemma 4. MLP for 12 rounds

The Maximum Linear Probability (MLP) for a 12-round NDN cipher is calculated as 2^{-120}, equivalent to the result obtained for differential cryptanalysis. This low probability indicates a negligible chance of constructing successful linear approximations across multiple rounds, ensuring robust resistance.

For a 12-round NDN cipher, the maximum linear probability is:

$$MLP^{12} = (2^{-2})^{(5\times12)} = 2^{-120}$$

This demonstrates robust resistance to linear attacks.

The NDN cipher shows significant resilience to both differential and linear cryptanalysis. The combination of active S-boxes, bit transformations, complement functions, and diffusion layers provides strong resistance to attacks. The low MDP and MLP values, especially for 12 rounds, ensure that the cipher is secure under extensive cryptanalysis.

4.3. Algebraic cryptanalysis

An algebraic attack seeks to exploit the algebraic structure of a cipher to derive secret keys by solving a system of multivariate equations. Courtois and Pieprzyk [47] introduced this method, highlighting its potential against ciphers with weak nonlinear components or insufficient diffusion.

Theorem 7. The proposed NDN lightweight block cipher resists algebraic attack.

Proof. An algebraic attack seeks to exploit the algebraic structure of a cipher to derive secret keys by solving a system of multivariate equations. Courtois and Pieprzyk [47] introduced this method, highlighting its potential against ciphers with weak nonlinear components or insufficient diffusion.

The primary nonlinear component of the NDN cipher is the 4×4 S-box, which contributes significant complexity to the cipher's algebraic representation. Each S-box is expressed as a system of multivariate quadratic equations.

Representation of a Single S-box:

- The 21 quadratic equations involving eight (8) variables describe the 4 x 4 S-box used in NDN cipher: 4 input and 4 output variables.

Total Equations and Variables in NDN:

- For an NDN cipher with an 80-bit key and 12 rounds:

Number of S-boxes per round: $2 \times 4 = 8$

Total S-boxes: $12 \times 8 = 96$

Total equations: $96 \times 21 = 1152$

Total variables: $96 \times 8 = 876$

- For an NDN cipher with a 128-bit key and 18 rounds:

Number of S-boxes per round: $2 \times 4 = 8$

Total S-boxes: $18 \times 8 = 144$

Total equations: $144 \times 21 = 3024$

Total variables: $144 \times 8 = 1152$

Recent advances in algebraic cryptanalysis, as explored in [47], emphasize the need for high S-box non-linearity and robust diffusion layers. NDN addresses these requirements through its unique combination of 4×4 S-boxes and SPN structure, outperforming comparable lightweight ciphers in algebraic resistance.

Attempts to represent the 12-round NDN cipher in Algebraic Normal Form (ANF) using SAGE resulted in a computationally infeasible system of 1,152 equations with 768 variables, further demonstrating resistance.

The algebraic degree of the NDN cipher grows significantly due to the iterative application of 4×4 S-boxes and the diffusion layers. After 12 rounds, the degree reaches approximately 2^{10}, rendering higher-order algebraic attacks impractical.

The dynamic bit transformations in the NDN key schedule further complicate the algebraic representation of the cipher. These transformations ensure that each round key introduces new non-linearities, increasing the difficulty of solving the equations for key recovery.

The high number of equations and variables makes solving the system computationally infeasible. Furthermore, transforming the entire cipher to its Algebraic Normal Form (ANF) significantly increases the time and effort required due to its high non-linearity and diffusion properties.

Conclusion: The complexity of the NDN cipher's algebraic structure, driven by its robust S-box supported by complement function and multi-round design, ensures resistance to algebraic attacks.

4.4. Related Key Cryptanalysis

A related-key attack exploits patterns in round keys to recover the master key. NDN’s dynamic key schedule prevents such attacks by ensuring high non-linearity and diffusion in round keys. Bhim et al. [43] demonstrated such attacks using combined boomerang and rectangle techniques.

Theorem 8. The proposed NDN lightweight block cipher resists related key attack.

Proof. The number of active S-boxes in related-key contexts demonstrates how resistant the NDN cipher is to related-key attacks.

Minimum Active S-boxes in Three Rounds:

As shown in Theorem 3, any three consecutive rounds in the NDN cipher have at least 15 active S-boxes. Each active S-box contributes a differential and linear probability of 2^{-2}, ensuring strong diffusion and confusion properties.

Active S-boxes for 80-bit Key (12 Rounds):

For the NDN cipher with an 80-bit key and 12 rounds:

Total active S-boxes = $\frac{12 \times 15}{3} = 60$

$MDP^{12} = MLP^{12} = (2^{-2})^{(60)} = 2^{-120}$

Active S-boxes for 128-bit Key (18 Rounds):

For the NDN cipher with a 128-bit key and 18 rounds:

Total active S-boxes = $\frac{18 \times 15}{3} = 90$

$MDP^{18} = MLP^{18} = (2^{-2})^{(90)} = 2^{-180}$

The NDN cipher's key schedule introduces additional complexity in related key settings. Key bits are subjected to non-linear transformations and permutations, ensuring high diffusion and non-linearity in derived round keys. It prevents predictable relationships between keys, further strengthening resistance to related-key attacks.

4.5. Impossible Differential Cryptanalysis

Impossible Differential Cryptanalysis (IDC) is a cryptanalytic technique that focuses on ciphers by utilizing impossible differential transitions—pairs of input and output differentials that cannot occur for any key value due to the cipher's inherent properties. Attackers employ these pairs to eliminate invalid keys during the key-recovery phase.

Significance:

- Powerful Cryptanalytic Tool: IDC is particularly effective against ciphers with structural weaknesses or poorly designed S-box layers, where such impossible transitions are not inherently blocked.
- Differential Trail Identification: It constructs impossible trails over multiple rounds to exploit inconsistencies in the cipher's differential behaviour.
- Focus on S-box Layer: The S-box is a critical component in determining the presence of such impossible pairs, as it governs the local differential properties of the cipher.

IDC and NDN Cipher: Redundancy of IDC Analysis

Zero Entries in the DDT:

- The Difference Distribution Table (DDT) for the NDN cipher inherently lists all possible and impossible differential transitions at the S-box level.
- Zero entries in the DDT represent input-output differential pairs that are impossible. The DDT encodes these pairs by default, eliminating the need for manual or automated construction of impossible trails.

SP-Network Characteristics:

- The Substitution-Permutation Network (SPN) structure of NDN, combined with its robust design choices, inherently prevents impossible differential trails from propagating across rounds.
- The 4×4 S-box effectively blocks invalid transitions at the local level, while the diffusion properties, such as branch numbers and transformations, enhance this effect globally across the cipher rounds.

Focus on Practical Security:

- Since IDC relies on detecting impossible differential transitions, and the NDN cipher inherently eliminates these through its S-box design and DDT properties, any effort spent on IDC analysis for such a design is redundant. Thus, NDN is resistant to Impossible Differential Cryptanalysis (IDC).

The NDN lightweight block cipher has been rigorously analyzed against various cryptanalytic techniques, including differential, linear, algebraic, related-key, and impossible differential attacks. The results confirm that the cipher strongly resists all known cryptanalytic methods. Its robust design, featuring well-structured S-boxes, high diffusion properties, and a non-linear key schedule, ensures the NDN cipher is highly secure. Therefore, the NDN cipher can be considered resilient to known attacks and provides a solid foundation for securing resource-constrained environments such as IoT devices.

4.6. Core Innovations Driving Unmatched Security Excellence

The proposed NDN cipher demonstrates superior security characteristics through several innovative design elements, which enhance its resilience collectively against known cryptanalytic techniques. The key highlights of the security analysis are as follows:

4.6.1. Primitive Polynomial-Based Bit Transformation

The primitive polynomial $x^4 \oplus x^3 \oplus 1$ enables 16 distinct bit transformation options. Carefully selected transformations enhance non-linearity and randomness in the encryption process. This mechanism strengthens the cipher’s resistance to differential and linear cryptanalysis by amplifying the avalanche effect and increasing algebraic complexity.

4.6.2. Diffusion Function with Branch Number 5

The diffusion function, defined with a branch number of 5, ensures significant propagation of bit changes across rounds. This design guarantees that even a single-bit change in the plaintext propagates widely, making it difficult for attackers to trace input-output correlations. The branch number of 5 ensures optimal diffusion, contributing to the high number of active S-boxes per round, thereby enhancing the cipher’s defense against differential and linear attacks.

4.6.3. Complement Function for Additional Non-Linearity

The complement function, integrated into the encryption and key generation schedules, introduces further non-linearity by flipping specific bits during processing. This operation complicates differential and algebraic trails, reducing the probability of successful cryptanalytic attacks further.

4.6.4. 16-Point Radix-4 DFFT Permutation for Enhanced Key Diffusion

The round key generation schedule employs a 16-point radix-4 Discrete Fourier Transform (DFFT)-based permutation inspired by neural network principles. This permutation maximizes the diffusion of key bits across rounds, ensuring that even minor changes in the user key lead to vastly different round keys. The complexity of solving such a permutation in algebraic terms further strengthens the cipher's resistance to algebraic and related-key attacks.

4.6.5. High Number of Active S-Boxes in Multiple Rounds

The analysis establishes that any three consecutive rounds of the NDN cipher activate a minimum of 15 S-boxes. This ensures a maximum differential probability (MDP) and maximum linear probability (MLP) of 2^{-30} for three rounds and 2^{-120} for 12 rounds, providing robust defense against differential and linear cryptanalysis.

4.6.6. Resistance to Impossible Differential and Related-Key Attacks

The inherent design properties of the S-box and diffusion layers block impossible differential transitions, rendering Impossible Differential Cryptanalysis (IDC) ineffective. In addition, the dynamic bit transformations and DFFT-based key permutations ensure that related-key attacks are highly impractical, as round keys exhibit high diffusion and non-linearity across rounds.

5. PERFORMANCE EVALUATION ANALYSIS

Performance evaluation is critical in assessing the practicality of lightweight block ciphers for real-world IoT applications. With diverse operating environments and constraints of IoT systems—from low-power sensors to high-security industrial controllers—this evaluation must go beyond traditional metrics. In this chapter, the NDN cipher is rigorously benchmarked against state-of-the-art lightweight ciphers using a comprehensive portfolio of metrics covering hardware and software implementations, cryptographic robustness, and extended performance criteria.

The evaluation encompasses three key dimensions:

- Hardware Implementation Comparison: ASIC technology is employed to assess the hardware efficiency of the NDN and compare it with that of existing leading ciphers. This section highlights resource consumption, such as gate equivalents, showcasing the cipher suitability for direct integration into IoT devices.
- Software Implementation Comparison: The performance of cipher on an AVR RISC-based microcontroller is analyzed, emphasizing memory utilization, time complexity, and adaptability for resource-constrained environments. Comparative analysis with prominent ciphers further validates NDN cipher edge in real-world scenarios.
- Cryptographic Robustness and Extended Metrics: The cipher's suitability is assessed using traditional metrics such as resistance to differential, linear, and algebraic attacks. Additionally, extended metrics evaluate its scalability, memory efficiency, and efficient round key generation. These assessments highlight NDN's adaptability across various IoT domains, including smart grids, autonomous systems, and healthcare.

This comprehensive evaluation highlights NDN's balanced performance in resource efficiency, speed, and security, reinforcing its suitability for securing next-generation IoT infrastructures. This chapter highlights how NDN outperforms existing solutions, setting a new standard in lightweight cryptographic design through detailed comparisons and analyses.

5.1. Performance in Hardware

This comparison highlights the hardware resource efficiency of the NDN cipher, showcasing its gate-equivalent requirements relative to other well-known lightweight ciphers implemented using ASIC technologies. Table 11 compares the hardware performance of the NDN cipher with existing lightweight block ciphers.

Table 11 Comparison of Hardware implementation

Algorithm	Block Size	Key Size	GE	Technology µm
Present [1]	64	80	1570	0.18
Piccolo [3]	64	80	1136	0.13
Piccolo [3]	64	128	1196	0.13
SLIM [10]	32	80	1028.25	0.13
LBC-IoT [11]	32	80	1028.25	0.13
RAZOR [50]	64	128	1260	0.13
SIT [52]	64	64	1050	0.13

RECTANGLE [55]	64	80/128	1250	0.13
RBFK [51]	64	64/128	1200	0.13
NDN Proposed	64	80	1068	0.13
NDN Proposed	64	128	1272	0.13

Analysis:

Table 11 implies that NDN requires slightly higher GEs than ciphers like SLIM (32-bit cipher) and SIT (64-bit cipher), but it offers significantly enhanced security through its dynamic key schedule, diffusion layer with a branch number of 5, and primitive polynomial-based bit transformations. In addition, NDN provides better scalability, supporting 80-bit and 128-bit key sizes with minimal resource overhead.

5.2. Performance in Software

The software implementation comparison emphasizes the NDN cipher excellence in memory footprint and processing time on AVR RISC architecture-based microcontrollers, making it suitable for IoT devices with constrained resources. Table 12 highlights the software performance metrics.

Table 12 Comparison of Software Implementation

Algorithm	Block size	Key size	Memory (KB)	Time (µs)
Present [1]	64	80	3048	847.33
Piccolo [3]	64	80	2016	673.25
Piccolo [3]	64	128	2234	728.96
SLIM [10]	32	80	2045	89.75
LBC-IoT [11]	32	80	2045	90.12
RAZOR [50]	64	128	2136	110.34
SIT [52]	64	64	2080	102.45
RECTANGLE [55]	64	80/128	2180	120.67
RBFK [51]	64	64/128	2139	99.12
NDN Proposed	64	80	2139	110.23
NDN Proposed	64	128	Memory (KB)	Time (µs)

Analysis:

The enhanced table demonstrates NDN's competitive performance in software implementation on AVR RISC architecture-based microcontrollers. While RECTANGLE and SIT offer slightly lower memory usage, NDN outperforms most ciphers in execution time, especially in encryption and decryption operations, making it highly suitable for real-time applications in resource-constrained environments. Additionally, the dual-key support of NDN ensures adaptability across diverse IoT infrastructures.

5.3. Performance Against Cryptographic Attacks

Evaluating a cipher's resilience to cryptographic attacks is crucial for ensuring its robustness in real-world applications. This section presents a detailed comparison (Table 13) of the NDN cipher's resistance to various attacks, such as differential, linear, algebraic, related-key, and impossible differential cryptanalysis, benchmarking its security strength against established lightweight ciphers.

Analysis:

Table 13 highlights the cryptographic strength of the NDN cipher compared to existing lightweight block ciphers across various attack vectors:

Differential and Linear Cryptanalysis:

NDN-80 and NDN-128 demonstrate exceptionally low differential probability (MDP) and linear probability (MLP), of 2^{-120} and 2^{-180}, respectively. These values indicate a high resistance to differential and linear attacks, outperforming ciphers like PRESENT, SLIM, and RECTANGLE.

- PRESENT achieves comparable differential resistance with MDP = 2^{-120}, but lacks linear probability evaluation, making it less comprehensive in security analysis.

- SLIM and LBC-IoT show moderate resistance with MDP = 2^{-90}, reflecting reasonable cryptographic strength but inferior to NDN.
- RAZOR stands out with an impressive MDP and MLP of 2^{-192}, indicating very high resistance, rivaling NDN in cryptographic robustness.

Table 13 Performance Against Cryptographic Attacks

Cipher	Differential Probability (MDP)	Linear Probability (MLP)	Related-Key Resistance	Algebraic Complexity	IDC Resistance
NDN-80	2^{-120}	2^{-120}	High	High	Resistant
NDN-128	2^{-180}	2^{-180}	Very High	Very High	Resistant
PRESENT [1]	2^{-120}	-	Moderate	Moderate	Moderate
RECTANGLE [55]	-	-	Low	Moderate	Weak
SLIM [10]	2^{-90}	2^{-92}	Moderate	Moderate	Resistant
LBC-IoT [11]	2^{-90}	2^{-90}	Moderate	Moderate	Resistant
RBFK [51]	2^{-12}	-	Low	Low	Weak
SLA [31]	2^{-80}	-	Moderate	High	Resistant
RAZOR [50]	2^{-192}	2^{-192}	Very High	Very High	Strong
FEW [9]	2^{-100}	2^{-101}	Moderate	High	Moderate

Related-Key Attack Resistance:

NDN ciphers exhibit high resilience to related-key attacks due to the dynamic key schedule involving bit transformations and permutation-based diffusion.

- RAZOR also shows strong resistance, attributed to its complex key schedule.
- SLIM and LBC-IoT provide moderate resistance, while RECTANGLE and RBFK demonstrate lower resistance, making them less suitable for high-security environments.

Algebraic Complexity:

NDN’s algebraic complexity is high for 80-bit and 128-bit keys due to the iterative use of non-linear S-boxes, bit transformations, and diffusion layers.

- RAZOR matches NDN in terms of very high algebraic complexity.
- PRESENT and RECTANGLE offer moderate resistance, making them vulnerable to advanced algebraic attacks.

Impossible Differential Cryptanalysis (IDC):

NDN resists impossible differential cryptanalysis (IDC) due to its carefully designed S-boxes and SPN structure.

- SLIM, LBC-IoT, and SLA exhibit resistance to IDC, whereas RECTANGLE and RBFK are susceptible to this attack.

Insights:

- NDN’s superiority: The NDN cipher (80-bit and 128-bit versions) stands out as a robust solution, excelling in differential, linear, algebraic, and related-key attack resistance.
- RAZOR’s competitive edge: RAZOR competes closely with NDN, particularly in algebraic complexity and resistance to advanced attacks.
- Weaknesses in older ciphers: RECTANGLE and RBFK lag in cryptographic strength, highlighting the need for innovative designs like NDN to address modern IoT security challenges.

5.4. Broader Performance Metrics

Table 14 provides a comparative overview of key performance metrics like scalability, memory efficiency, and adaptability. NDN cipher flexible architecture supports consistent performance across different key sizes, making it highly scalable. Its compact design ensures low memory consumption and is suitable for resource-constrained IoT environments. In addition, the fast and diffusive round key generation mechanism enhances its adaptability across various domains, such as smart grids, healthcare, and automotive systems. Compared to well-established ciphers like PRESENT, RECTANGLE, and AES, NDN offers a balanced trade-off between performance and resource efficiency, making it a versatile choice for diverse IoT infrastructures.

Table 14 Broader Performance Metrics Comparison

Cipher	Block Size	Key Size	Rounds	Scalability	Memory Efficiency	Round Key Generation Efficiency	Adaptability
NDN-80	64	80	12	High (supports key scaling)	Excellent (low memory use)	Fast & Diffusive	IoT, Smart Grids, Vehicles
NDN-128	64	128	18	High (supports key scaling)	Excellent (low memory use)	Fast & Diffusive	IoT, Smart Grids, Vehicles
PRESENT	64	80	31	Moderate	Good	Moderate	IoT only
RECTANGLE	64	80	25	Moderate	Moderate	Moderate	IoT only
SLIM	32	80	32	Moderate	Good	Moderate	IoT only
LBC-IoT	32	80	32	Moderate	Good	Moderate	IoT only
RBFK	64	80/128	5	Low	Poor	Moderate	IoT only
SLA	64	80/128	16	Moderate	Good	Moderate	IoT, Industrial IoT
RAZOR	64	128	32	High	Moderate	High Cost	IoT, Industrial IoT
FEW	64	80/128	32	Moderate	Good	Moderate	IoT only

Analysis: The analysis of the Table 14 is as follows.

Scalability:

NDN demonstrates high scalability by supporting 80-bit and 128-bit key sizes. This flexibility enables it to accommodate a wide range of security needs in IoT applications, spanning low-power sensors to high-security smart grids.

- RAZOR also shows high scalability due to its multi-key size support.
- PRESENT, RECTANGLE, SLIM, and other ciphers show moderate scalability due to their fixed key schedules or limited adaptability to varying security needs.
- Despite supporting multiple key sizes, AES incurs a high computational cost, making it less scalable for lightweight IoT environments.

Memory Efficiency:

NDN achieves excellent memory efficiency, with minimal SRAM and flash memory usage, making it ideal for constrained-resource devices.

- PRESENT, SLIM, and LBC-IoT exhibit good memory efficiency, but they lack the low-latency performance and scalability of NDN.
- AES requires significant memory resources, reducing its suitability for lightweight IoT applications

Round Key Generation Efficiency:

NDN’s fast and diffusive round key generation is a standout feature, enabled by its permutation-based key schedule and bit transformations.

- RAZOR offers high resistance to attacks but incurs a high cost in round key generation due to its complex key schedule.
- PRESENT, RECTANGLE, and similar ciphers show moderate efficiency in key generation, balancing performance and security.

Adaptability Across Infrastructures:

NDN’s versatility is evident in its applicability to IoT, smart grids, and vehicles, ensuring broad deployment potential.

- SLA and RAZOR also demonstrate adaptability, extending to industrial IoT applications.
- Ciphers like PRESENT, RECTANGLE and SLIM are primarily suited for generic IoT-only environments due to their limited flexibility and scalability.

5.5. Discussions and Insights

Table 14 provides a holistic comparison of the NDN cipher with established lightweight block ciphers. The discussion below highlights key aspects of the evaluation:

Scalability:

The NDN cipher outperforms most existing lightweight ciphers in scalability due to its dual key size support (80-bit and 128-bit) and adaptable round function. This flexibility ensures that NDN can be effectively deployed across IoT devices with varying security and performance requirements. Unlike PRESENT and RECTANGLE, which have fixed key sizes and limited scalability, NDN's key schedule allows seamless scaling while maintaining efficiency. While AES offers flexibility with different key sizes (128, 192, and 256 bits), its computational cost makes it unsuitable for resource-constrained environments.

Memory Efficiency:

NDN demonstrates excellent memory efficiency, requiring minimal SRAM, flash memory, and EEPROM resources. This advantage is critical for IoT devices that typically have limited memory capacity. Comparatively, AES and RAZOR demand higher memory usage due to their complex key schedules and block sizes. PRESENT and RECTANGLE offer reasonable memory usage but lack the high efficiency shown by NDN, making the latter more suitable for ultra-constrained devices like sensors and wearable technologies.

Round Key Generation Efficiency:

The NDN cipher's round key generation process leverages a fast and diffusive mechanism driven by a unique combination of bit transformations, complement functions, and neural network-inspired permutations. It results in efficient round key generation without sacrificing security. Unlike AES, which has a high computational cost for key expansion, and RAZOR, which incurs significant overhead, NDN strikes an optimal balance between speed and security.

Adaptability Across Infrastructures:

One of NDN's standout features is its adaptability across IoT domains, including smart grids, autonomous vehicles, and healthcare systems. This adaptability stems from its scalable design and efficient implementation, making it suitable for environments requiring high throughput and low power consumption.

While PRESENT and RECTANGLE cater primarily to generic IoT applications, they lack the versatility demonstrated by NDN. On the other hand, AES offers broader adaptability but at the cost of high computational resources, limiting its practical use in constrained environments.

Conclusion:

The NDN cipher excels in key performance metrics such as scalability, memory efficiency, and adaptability. Its innovative round key generation mechanism and efficient use of resources make it a superior choice for securing resource-constrained IoT devices. Unlike existing ciphers, which often prioritize specific metrics at the expense of others, NDN achieves a balanced approach, ensuring real-world applicability across diverse infrastructures.

6. SUMMARY OF RESEARCH

The research into lightweight block ciphers has traditionally been driven by metrics such as Gate Equivalents (GE), memory footprint, and processing delay, offering insights into platform-specific implementation aspects. The conventional metrics fail to reflect broader, real-world performance requirements of cryptographic solutions, particularly in heterogeneous IoT environments in a narrow sense. The highly platform-dependent nature of such evaluations has led to irrelevant and inconsistent comparative analyses, limiting scientific rigor and innovation in cipher design.

The term "IoT nodes" is often used too broadly in cryptographic research, neglecting the significant contextual differences across application domains. For example, a wearable health monitor in the medical Internet of Things has quite different limitations than a temperature sensor in an industrial context. Traditional evaluations become superficial and offer no practical insight into real-world deployment if these contextual factors are ignored.

In recognition of these limitations, while we have adhered to conventional comparative practices in this study, we underscore the necessity of context-aware, application-agnostic metrics. These metrics offer a more accurate and scientifically sound basis for evaluating lightweight block ciphers, transcending platform-specific constraints, and providing a clearer picture of real-world utility.

6.1. Scientific Rigor

This research introduces the NDN lightweight block cipher, a cryptographic design that exemplifies scientific rigor through several innovative features tailored for diverse IoT applications. Below are the key scientific advancements:

Artificial Neural Network-Inspired Permutations:

- The NDN cipher employs a permutation mechanism inspired by neural networks in its F-function.
- By leveraging concepts from deep learning, the cipher achieves enhanced non-linearity and diffusion properties, ensuring robust security with minimal computational overhead.

- This novel approach reduces reliance on complex and resource-intensive key scheduling methods, promoting efficiency across IoT platforms.

Key-Bit-Based Permutation in the F-Function:

- A unique dynamic key-bit-based permutation is incorporated in the F-function, enhancing diffusion without complex operations.
- This design improves adaptability, allowing multiple users keys and ensures robustness against cryptanalytic attacks.

Multiple User Key Options:

The NDN cipher supports two key sizes:

- 80-bit key version: Prioritizes energy efficiency for low-power IoT applications.
- 128-bit key variant: Provides enhanced security for critical applications like smart grids and healthcare.

This dual-version approach ensures flexibility, scalability, and future-proofing, making the cipher adaptable to evolving security needs.

Simplicity and Scalability:

- The NDN cipher achieves simplicity by avoiding resource-heavy operations.
- Its reliance on lightweight primitives such as XOR, bit shifts, and modular additions ensures efficient performance in both hardware and software implementations.
- The cipher's scalable design enables deployment in diverse environments, ranging from battery-powered medical devices to industrial IoT systems requiring real-time responsiveness.

Application-Agnostic Design:

- Unlike existing ciphers designed for specific scenarios, the NDN cipher adopts an application-agnostic approach, ensuring broad usability across different IoT domains.
- This generalizability makes it a suitable cryptographic solution across IoT devices, from wearable sensors to autonomous vehicles.

The NDN cipher achieves superior performance through innovative security mechanisms and computational efficiency. Unlike lightweight ciphers that rely on static transformation patterns, NDN incorporates dynamic adaptability, significantly enhancing resilience against differential, algebraic, and related-key attacks. Performance evaluations on ASIC and AVR RISC platforms confirm that NDN achieves a better trade-off between security, efficiency, and adaptability when compared to conventional lightweight block ciphers (LBCs). It ensures robust protection for resource-constrained IoT environments, positioning NDN as a promising solution for next-generation cryptographic security in IoT applications.

The NDN lightweight block cipher represents a forward-thinking approach to cryptography, addressing the limitations of traditional evaluation methods. This research marks a significant advancement in lightweight cryptography by integrating innovative design features such as neural network-inspired permutations, key-bit-based dynamic diffusion, and application-agnostic scalability. The proposed framework and cipher provide a robust, flexible, and efficient solution tailored for the diverse and resource-constrained landscape of IoT.

7. CONCLUSION AND FUTURE SCOPE

The proposed NDN lightweight block cipher introduces a novel design framework that addresses key limitations in existing cryptographic solutions. NDN ensures scalability, adaptability, and security across diverse IoT domains by integrating neural network-inspired permutations, dynamic key-bit-based transformations, and an application-agnostic design,

The comprehensive performance evaluation demonstrates that NDN optimally balances resource efficiency and cryptographic robustness, outperforming existing ciphers in practical IoT applications. The refined context-aware evaluation metrics validate NDN's real-world applicability, establishing a new benchmark for lightweight block cipher evaluations. It reinforces the impact of the research, as highlighted in the Research Summary, where NDN efficiency and security trade-offs have been rigorously analyzed and appreciated.

This work significantly advances the field of IoT cryptography by providing both a holistic evaluation framework and an adaptable cipher that meets the stringent security, efficiency, and deployment constraints of modern IoT infrastructures—including smart grids, connected vehicles, and healthcare systems. The study bridges theoretical cryptographic research and practical security solutions, fostering the broader adoption of lightweight cryptography in real-world applications.

7.1. Future Scope

Future research on the NDN cipher can explore several key areas for further enhancement:

- Advanced Cryptanalysis: Expanding the cryptanalysis to include side-channel resistance and fault injection attacks can bolster NDN’s robustness.
- Hardware and Software Optimizations: Real-time FPGA testing, low-power ASIC designs, and implementations on

various microcontroller platforms (e.g., ARM, RISC-V) can enhance NDN's efficiency for diverse IoT environments.

- Protocol and System Integration: Integrating NDN into IoT protocols like MQTT or CoAP can ensure secure end-to-end communication across IoT networks.
- Real-World Deployment and Metrics Refinement: Large-scale pilot deployments in smart grids and autonomous vehicles, combined with refined metrics such as real-time energy usage and throughput, can validate NDN's practical applicability.

REFERENCES

[1] Bogdanov, A., Knudsen, L. R., Leander, G., Paar, C., Poschmann, A., Robshaw, M. J. B., Seurin, Y., & Vikkelsoe, C. (2007). "PRESENT: An Ultra-Lightweight Block Cipher." Lecture Notes in Computer Science, 4727, Cryptographic Hardware and Embedded Systems - CHES 2007, 450–466. https://doi.org/10.1007/978-3-540-74735-2_31.

[2] Knudsen, L., & Robshaw, M. J. B. (2011). Block Cipher Companion. Springer. ISBN 978-3-642-17341-7.

[3] Shibutani, K., Isobe, T., Hiwatari, H., Mitsuda, A., Akishita, T., & Shirai, T. (2011). "Piccolo: An Ultra-Lightweight Block Cipher." Cryptographic Hardware and Embedded Systems – CHES 2011, Lecture Notes in Computer Science, 6917, 342–357. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-23951-9_23.

[4] Lim, C. H., & Korkishko, T. (2005). "mCRYPTON - a lightweight block cipher for security of low-cost RFID tags and sensors." WISA'05, Lecture Notes in Computer Science, 3786, 243–258, Springer-Verlag.

[5] Hong, D., Sung, J., Hong, S., Lim, J., Lee, S., Koo, B., Lee, C., Chang, D., Lee, J., Jeong, K., Kim, H., Kim, J., & Chee, S. (2006). "HIGHT: A new block cipher suitable for low-resource devices." CHES'06, Lecture Notes in Computer Science, 4249, 46–59, Springer-Verlag.

[6] Beaulieu, R., Shors, D., Smith, J., Clark, S. T., Weeks, B., & Wingers, L. (2013). "The SIMON and SPECK families of lightweight block ciphers." Cryptology ePrint Archive, Report 2013/404.

[7] Suzaki, Tomoyasu, et al. "Twine: A lightweight, versatile block cipher." ECRYPT workshop on lightweight cryptography. Vol. 2011. Springer Berlin, Heidelberg, 2011.

[8] Canni`ere, C. D., Dunkelman, O., & Knezevic, M. (2009). "KATAN and KTANTAN – a family of small and efficient hardware-oriented block ciphers." CHES, Lecture Notes in Computer Science, 5747, 272–288, Springer.

[9] Kumar, M., Pal, S. K., Yadav, & Panigrahi, A. (2019). "FEW: Lightweight Block Cipher." Turkish Journal of Mathematics and Computer Science, 11(2), 58–73.

[10] Aboshosha, B., Ramadan, R. A., Dwivedi, A. D., El-Sayed, A., & Dessouky, M. M. (2020). "SLIM: A Lightweight Block Cipher for Internet of Health Things." IEEE Access, Special Section on Lightweight Security and Provenance for IoHT. https://doi.org/10.1109/ACCESS.2020.3036589.

[11] Ramadan, R. A., Aboshosha, B. W., Yadav, K., Alseadoon, I. M., Kashout, M. J., & Elhoseny, M. (2021). "LBC-IoT: Lightweight Block Cipher for IoT Constraint Devices." Computers, Materials and Continua, 68(3), 3455–3473. https://doi.org/10.32604/cmc.2021.015519.

[12] Li, L., Zhang, W., Bao, Z., & Lin, D. (2016). "QTL: A new ultra-lightweight block cipher." Microprocessors and Microsystems, 45, 45–55. https://doi.org/10.1016/j.micpro.2016.03.011.

[13] Li, L., Liu, L., Li, S., & He, H. (2018). "SFN: A new lightweight block cipher." Microprocessors and Microsystems, 64, 1–10. https://doi.org/10.1016/j.micpro.2018.04.009.

[14] Guo, Y., Li, L., & Liu, B. (2021). "Shadow: A Lightweight Block Cipher for IoT Nodes." IEEE Internet of Things Journal, 8(16), 13014–13023. https://doi.org/10.1109/JIOT.2021.3064203.

[15] Biham, E., & Shamir, A. (1991). "Differential cryptanalysis of DES-like cryptosystems." Journal of Cryptology, 4(1), 3–72.

[16] Daemen, J., & Rijmen, V. (2001). The Design of Rijndael. Springer.

[17] Heys, H. M. (2017). "A Tutorial on Linear and Differential Cryptanalysis." Crypto 2017.

[18] Matsui, M. (1994). "Linear cryptanalysis method for DES cipher." Advances in Cryptology: EUROCRYPT '93, Springer-Verlag, 386–397.

[19] Nyberg, K. (1994). "Differentially uniform mappings for cryptography." EUROCRYPT 1993, Lecture Notes in Computer Science, 765, Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-48285-7_6.

[20] Kanda, M. (2000). "Practical Security Evaluation against Differential and Linear Cryptanalysis for Feistel Ciphers with SPN Round Function." SAC 2000, Lecture Notes in Computer Science, 2012, Springer-Verlag, 324–338.

[21] Daemen, J., & Rijmen, V. (2001). "The Wide Trail Design Strategy." Cryptography and Coding 2001, Lecture Notes in Computer Science, 2260, 222–238, Springer-Verlag Berlin Heidelberg. https://doi.org/10.1007/3-540-45325-3_20.

[22] Kanda, M. (2001). "Practical Security Evaluation against Differential and Linear Cryptanalysis for Feistel Ciphers with SPN Round Function." SAC 2000, Lecture Notes in Computer Science, 2012, 324-338, Springer-Verlag. https://doi.org/10.1007/3-540-45473-X_24.

[23] Xu, H., Hao, C., Cui, Y., & Qi, W. (2023). "Impossible Differential Cryptanalysis of Lightweight Block Cipher WARP." Research Square. https://doi.org/10.21203/rs.3.rs-3215560/v1.

[24] Yang, Q., Hu, L., Shi, D., Todo, Y., & Sun, S. (2018). "On the Complexity of Impossible Differential Cryptanalysis." Hindawi Security and Communication Networks, Volume 2018, Article ID 7393401, 11 pages. https://doi.org/10.1155/2018/7393401.

[25] Biham, E., Biryukov, A., & Shamir, A. (1999). "Miss in the Middle Attacks on IDEA and Khufu." Fast Software Encryption - FSE'99, Lecture Notes in Computer Science, 1636, 124–138, Springer-Verlag Berlin Heidelberg. https://doi.org/10.1007/3-540-48519-8_9.

[26] Bogdanov, A., & Rijmen, V. (2014). "Linear Hulls with Correlation Zero and Linear Cryptanalysis of Block Ciphers." Designs, Codes and Cryptography, 70, 369–383. https://doi.org/10.1007/s10623-012-9697-z.

[27] Soleimany, H., & Nyberg, K. (2014). "Zero-correlation Linear Cryptanalysis of Reduced-round LBlock." Designs, Codes and Cryptography, 73, 683–698. https://doi.org/10.1007/s10623-014-9976-y.

[28] Yi, W., Chen, S., & Li, Y. (2017). "Zero-correlation Linear Cryptanalysis of SAFER Block Cipher Family Using the Undisturbed Bits." The Computer Journal, 60(4), 613–624. https://doi.org/10.1093/comjnl/bxw086.

[29] Ahmad, A., & Elabdallai, A. M. (1997). "An Efficient Method to Determine Linear Feedback Connections in Shift Registers That Generate Maximal Length Pseudo-Random Up And Down Binary Sequences." Computers & Electrical Engineering, 23(1), 33–39. https://doi.org/10.1016/S0045-7906(96)00009-7.

[30] Rukhin, A., Soto, J., Nechvatal, J., Smid, M., Barker, E., Leigh, S., Levenson, M., Vangel, M., Banks, D., Heckert, A., Dray, J., & Vo, S. (2010). "A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications." NIST Special Publication, April 2010.

[31] Ibrahim, N., & Agbinya, J. (2023). "Design of a Lightweight Cryptographic Scheme for Resource-Constrained Internet of Things Devices." Applied Sciences, 13(7), 4398. https://doi.org/10.3390/app13074398.

[32] Bansod, G., Patil, A., Sutar, S., & Pishoroty, N. (2016). "ANU: An Ultra-lightweight Cipher Design for Security in IoT." Security and

Communication Networks, 9(15), 5238–5251. https://doi.org/10.1002/sec.1692.

[33] Mouha, N., Wang, Q., Gu, D., & Preneel, B. (2011). "Differential and Linear Cryptanalysis Using Mixed-Integer Linear Programming." Information Security and Cryptology, Inscrypt 2011, Lecture Notes in Computer Science, 7537, Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-34704-7_5.

[34] Xiang, Z., Zhang, W., Bao, Z., & Lin, D. (2016). "Applying MILP Method to Searching Integral Distinguishers Based on Division Property for Lightweight Block Ciphers." ASIACRYPT 2016, Lecture Notes in Computer Science, 10031, Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-53887-6_24.

[35] Damaj, I. W., Al-Mubasher, H., & Saadeh, M. (2022). "An Extended Analytical Framework for Heterogeneous Implementations of Light Cryptographic Algorithms." Future Generation Computer Systems. https://doi.org/10.1016/j.future.2022.11.007.

[36] Shirai, T., & Araki, K. (2008). "On Generalized Feistel Structures Using the Diffusion Switching Mechanism." IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, E91-A(8), 2120–2129.

[37] Matsui, M. (1995). "On Correlation between the Order of S-Boxes and the Strength of DES." In De Santis, A. (Ed.), EUROCRYPT 1994, Lecture Notes in Computer Science, 950, 366–375, Springer. https://doi.org/10.1007/3-540-48285-7_6.

[38] Tiwari, V., Jampala, N., Tentu, A. N., & Saxena, A. (2021). "Towards Finding Active Number of S-boxes on Block Ciphers Using Mixed Integer Linear Programming." Informatics, 45(6), 77–87. https://doi.org/10.31449/inf.v45i6.3427.

[39] Knudsen, L. R., & Wagner, D. (2002). "Integral Cryptanalysis." In Daemen, J., & Rijmen, V. (Eds.), Fast Software Encryption - FSE 2002, Lecture Notes in Computer Science, 2365, 112–127, Springer-Verlag. https://doi.org/10.1007/3-540-45661-9_9.

[40] Gilbert, H., & Minier, M. (2000). "A Collision Attack on 7 Rounds of Rijndael." In Proceedings of Third Advanced Encryption Standard Conference, National Institute of Standards and Technology, 230–241.

[41] Biham, E. (1994). "New Types of Cryptanalytic Attacks Using Related Keys." In Helleseth, T. (Ed.), Proceedings of EUROCRYPT '93, Lecture Notes in Computer Science, 765, 398–409, Springer-Verlag. https://doi.org/10.1007/3-540-48285-7_34.

[42] Biryukov, A., & Wagner, D. (2000). "Advanced Slide Attacks." In Preneel, B. (Ed.), Proceedings of EUROCRYPT 2000, Lecture Notes in Computer Science, 1807, 589–606, Springer-Verlag. https://doi.org/10.1007/3-540-45539-6_41.

[43] Daemen, J., & Rijmen, V. (2002). The Design of Rijndael: AES - The Advanced Encryption Standard. Springer-Verlag. ISBN 978-3-662-04722-4.

[44] Dinur, I., & Shamir, A. (2009). "Cube Attacks on Tweakable Black Box Polynomials." In Joux, A. (Ed.), EUROCRYPT 2009, Lecture Notes in Computer Science, 5479, 278–299, Springer. https://doi.org/10.1007/978-3-642-01001-9_16.

[45] Courtois, N. T., & Pieprzyk, J. (2002). "Cryptanalysis of Block Ciphers with Overdefined Systems of Equations." ASIACRYPT 2002, Lecture Notes in Computer Science, Springer, 267–287. https://doi.org/10.1007/3-540-36178-2_17.

[46] Xu, H., Hao, C., Cui, Y., & Qi, W. (2023). "Impossible Differential Cryptanalysis of Lightweight Block Cipher WARP." Research Square. https://doi.org/10.21203/rs.3.rs-3215560/v1.

[47] Biham, E., Biryukov, A., & Shamir, A. (1999). "Miss in the Middle Attacks on IDEA and Khufu." In Knudsen, L. (Ed.), Fast Software Encryption - FSE'99, Lecture Notes in Computer Science, 1636, 124–138, Springer-Verlag. https://doi.org/10.1007/3-540-48519-8_9.

[48] Boura, C., Naya-Plasencia, M., & Suder, V. (2014). "Scrutinizing and Improving Impossible Differential Attacks: Applications to CLEFIA, Camellia, LBlock, and Simon (Full Version)." IACR Cryptology ePrint Archive.

[49] Analog Devices. (2010). "Application Note 3666: Software Optimization of FFTs and IFFTs Using the SC3850 Core." Retrieved from https://www.analog.com/en/technical-articles/fft-optimization.html.

[50] Dheeraj, Kumar, M., & Yadav, T. (2024). "RAZOR: A Lightweight Block Cipher for Security in IoT." Defence Science Journal, 74(1), 46–52. https://doi.org/10.14429/dsj.74.18421.

[51] Rana, S., Mondal, M. R. H., & Kamruzzaman, J. (2023). "RBFK Cipher: A Randomized Butterfly Architecture-Based Lightweight Block Cipher for IoT Devices in the Edge Computing Environment." Journal of Cybersecurity, 6(1). https://doi.org/10.1186/s42400-022-00136-7.

[52] Usman, M., Ahmed, I., Aslam, M. I., Khan, S., & Shah, U. A. (2017). "SIT: A Lightweight Encryption Algorithm for Secure Internet of Things." arXiv Preprint. arXiv:1704.08688.

[53] Wu, W., & Zhang, L. (2011). "LBlock: A Lightweight Block Cipher." In Lopez, J., & Tsudik, G. (Eds.), Applied Cryptography and Network Security - ACNS 2011, Lecture Notes in Computer Science, 6715, Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-21554-4_19.

[54] Gong, Z., Nikova, S., & Law, Y. W. (2012). "KLEIN: A New Family of Lightweight Block Ciphers." In Juels, A., & Paar, C. (Eds.), RFID Security and Privacy - RFIDSec 2011, Lecture Notes in Computer Science, 7055, Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-25286-0_1.

Authors

Mr. Nagaraj Hediyal has an undergraduate degree in Electrical Engineering and a Master’s degree in Computer Applications & Industrial Drives. He has over 30 years of experience in Research and Development in the field of Cryptography, Embedded, VLSI Communication, HVAC controllers, Power Electronics and Renewable Energy. His research areas include cryptography and network security, embedded and VLSI, secured communications, advanced power electronics, renewable energy, etc.

Divakar B. P. received an M.E. degree in power systems from Annamalai University, India in 1991 and a Ph.D. degree in Power Electronics from the Hong Kong Polytechnic University, Hong Kong in 1998. He worked as RA/RF/Staff in the Electrical Engineering department of the Hon Kong Polytechnic University from 1998-2009. He later joined REVA ITM, Bangalore, India as Professor in 2009. His research interests include soft-switching, power factor correction, BMS, HID, LED lighting, and multilevel inverters. He is guiding six Ph.D. scholars and was also the PI of a funded project on BMS. Presently he is working as Director, R&D Cell, REVA University, Bengaluru, India.

How to cite this article:

Nagaraj Hediyal, Divakar B.P, “NDN: An Ultra-Lightweight Block Cipher to Secure IoT Nodes”, International Journal of Computer Networks and Applications (IJCNA), 12(2), PP: 227-251, 2025, DOI: 10.22247/ijcna/2025/15.