



A Novel Block Cipher Construction Based on Latin Square Substitution and Modular Permutation

Robbi Rahim¹, Ronal Watrianthos², Zulidyana Dwi Rusnalasari³

¹STIM Sukma Medan, Medan, Indonesia

²Politeknik Negeri Padang, Padang, Indonesia

³Universitas Negeri Surabaya, Surabaya, Indonesia

*robhirahim@ieee.org

Abstract

Lightweight encryption is necessary to protect data on devices with limited resources, such as Internet of Things (IoT) nodes, radio-frequency identification (RFID) systems, and embedded controllers. PRESENT, PRINCE, and SPECK are examples of lightweight ciphers that work well but use fixed substitution and static diffusion structures that make it difficult for keys to change and make them less resistant to advanced cryptanalysis. This study presents a new lightweight block cipher that combines dynamic substitution based on Latin squares with modular permutation diffusion to improve both confusion and diffusion features. Encryption uses a key-dependent Latin Square to create unique nonlinear substitution mappings for each round. It also uses modular arithmetic to drive the permutation for efficient diffusion with a low hardware cost. The experimental evaluation showed a large avalanche effect (51.6%), differential probability (2^{-38}), and linear approximation probability (2^{-39}). These numbers are better than those of current designs in terms of cryptanalytic resistance. Hardware synthesis on FPGA yields 1,650 gate equivalents, 14.9 Mbps throughput, and 0.63 nJ/bit energy efficiency, which shows that it is suitable for lightweight encryption. The results show that the proposed design strikes a good balance between security and efficiency, bridging the gap between the theoretical cryptographic strength and the actual low-power implementation.

Keywords: block cipher; IoT security; lightweight cryptography; latin square substitution; modular per-mutation

How to Cite: R. . Rahim, R. . Watrianthos, and Z. D. . Rusnalasari, "A Novel Block Cipher Construction Based on Latin Square Substitution and Modular Permutation", *J. RESTI (Rekayasa Sist. Teknol. Inf.)*, vol. 10, no. 5, pp. 1483–1489, Oct. 2026, doi: [10.29207/resti.v10i5.7565](https://doi.org/10.29207/resti.v10i5.7565).
Permalink/DOI: <https://doi.org/10.29207/resti.v10i5.7565>

Received: February 7, 2026

Accepted: September 4, 2026

Available Online: October 16, 2026

This is an open-access article under the [CC BY 4.0 License](#)

Published by [Ikatan Ahli Informatika Indonesia](#)

1. Introduction

Cryptography is important for keeping digital information private, safe, and real in today's communication systems[1]. The rapid expansion of the Internet of Things (IoT), wireless sensor networks (WSNs), and embedded systems has rendered conventional cryptographic algorithms, such as the Advanced Encryption Standard (AES)[2], excessively computationally demanding for devices with constrained resources such as memory, power, and processing capabilities[3], [4]. This has resulted in an increasing focus on lightweight cryptography, which aims to deliver sufficient security in limited hardware and energy environments[5].

Lightweight block ciphers attempt to balance security, ease of use, and implementation efficiency[6-8]. They achieve this by using smaller block sizes, easier key schedules, and fewer gate equivalents. PRESENT,

PRINCE, and SPECK are some of the most well-known lightweight block ciphers that have become the benchmarks. Each represents a distinct optimization method[9-11].

The PRESENT algorithm, developed by Bogdanov et al.[9], employs a Substitution-Permutation Network (SPN) architecture featuring a 64-bit block and a key size of either 80 or 128 bits[5]. It uses 4-bit S-boxes and a simple bit permutation layer to remain small, with approximately 1570 gate equivalents, which makes it suitable for devices that require very little power. However, PRESENT's fixed S-box architecture and deterministic diffusion layer can create regularities in the structure, which could make it easier for differential and linear cryptanalysis to work in related-key situations[12].

Borghoff et al.[10] proposed the PRINCE cipher, which focuses on low-latency encryption by adding an α -

reflection characteristic that creates symmetric structures for encryption and decryption. It is fast and does not take up much space on hardware, but its reflective design could make it vulnerable to certain reflection and meet-in-the-middle attacks, and its key schedule does not provide a lot of variety[13].

Beaulieu et al.[11] created the SPECK algorithm, which uses an addition-rotation-XOR (ARX) structure that works well in software environments[14]. It shows better performance and flexibility with varied block and key sizes, but its simple modular addition and rotation functions can show restricted diffusion and weaker algebraic complexity, which raises concerns about its performance against advanced attacks in the long run[15].

Beyond these foundational designs, the lightweight cryptography landscape has continued to evolve. GIFT [16], a successor to PRESENT, introduces bit-slicing optimisations that further reduce gate count while preserving the SPN structure. SKINNY [17], a tweakable block cipher family, incorporates a dedicated tweak input to enhance domain separation across use cases. Most significantly, Ascon was selected in 2023 as the NIST Lightweight Cryptography Standard for authenticated encryption on constrained devices, operating on a 320-bit sponge state [18].

While these modern designs advance the field in their respective directions, they predominantly retain static or structurally fixed substitution and diffusion layers or operate within design paradigms (sponge, tweakable). The present work therefore addresses a specific and underexplored gap within the SPN family not realised by any of the above designs in their standard configurations.

Even though these algorithms work well, they all have one major problem: they all use static substitution boxes (S-boxes) and fixed diffusion matrices that remain the same for all rounds and key instances. Static elements create predictable patterns of change that attackers can use with differential, linear, or integral cryptanalysis methods[19], [20]. In addition, most implementations do not include dynamic key-dependent transformations that can change the structure of the cipher for each encryption instance. This might considerably increase confusion and spread without incurring significant extra hardware costs.

A closer examination of the literature reveals four specific gaps that the existing designs have not simultaneously resolved. While dynamic S-box schemes based on chaotic maps or quantum random walks improve nonlinearity over fixed boxes, they depend on floating-point arithmetic that is unstable in resource-constrained hardware.

Existing diffusion layers are either fully static and key-independent (PRESENT's fixed bit permutation), computationally demanding (PRINCE's matrix diffusion), or require high round counts to achieve full mixing (SPECK's ARX structure requiring up to 34

rounds), and none achieves simultaneous round-wise variability, formal reversibility, arithmetic simplicity, and key-dependence.

This study presents an innovative block cipher design that combines Latin Square-based substitution with modular permutation-based diffusion to address these issues. A Latin Square, a combinatorial layout in which each symbol appears precisely once per row and column, can produce dynamic key-dependent substitution layers that enhance nonlinearity and randomness[21]. Simultaneously, modular permutation adds efficient, reversible, and unexpected diffusion through arithmetic transformations that are easy to implement in hardware.

The objective of this study is to develop a lightweight block cipher that enhances key strength by employing dynamic substitution based on Latin squares. Additionally, it aims to design a modular permutation diffusion layer that achieves significant diffusion while minimizing the computational overhead. This study also involves evaluating the proposed encryption method against established benchmark algorithms such as PRESENT, PRINCE, and SPECK, focusing on the security performance of the new design.

This study proposes a novel design paradigm by integrating combinatorial mathematics with modular arithmetic, thereby bridging the divide between theoretical cryptographic strength and practical lightweight implementation, resulting in a secure, efficient, and scalable block cipher for next-generation embedded applications.

2. Methods

The second section provides a detailed overview of the proposed lightweight block cipher design, operational framework, and evaluation methodology. This cipher is structured around a substitution-Permutation Network (SPN) architecture[22-24], which is enhanced by two key components: Latin Square-based Substitution (LSS) and Modular Permutation Diffusion (MPD)[25], [26]. The integration of the LSS and MPD mechanisms aims to strengthen the cryptographic principles of confusion and diffusion, which are essential for securing data against cryptanalysis.

Simultaneously, the design prioritizes maintaining low computational complexity, ensuring that the cipher remains efficient and practical for implementation on devices with constrained resources, such as embedded systems and IoT devices. This section further elaborates on how these mechanisms interact within the SPN framework to achieve robust security while optimizing performance, and outlines the criteria and procedures used to evaluate the cipher's effectiveness and resource efficiency.

2.1 Design Principles

The proposed cipher was designed following the classical principles of confusion and diffusion

established by Shannon. To achieve confusion, it employs a key-dependent nonlinear substitution mechanism based on dynamically generated Latin Squares. This approach ensures that the substitution process varies uniquely with each key, enhancing security by making the relationship between the key and the ciphertext highly complex. For diffusion, the cipher incorporates a modular permutation function that thoroughly mixes bits across the internal state, ensuring that small changes in the input are propagated widely throughout the encryption process.

Unlike traditional ciphers such as PRESENT or PRINCE, which rely on fixed substitution and permutation layers, this architecture dynamically generates both substitution and permutation layers from the key schedule. This dynamic creation guarantees that every encryption instance uses a distinct set of round transformations tailored to the specific key, thereby strengthening the resistance against cryptanalytic attacks by eliminating predictable patterns across different encryptions.

2.2 Cipher Structured

Structured Cipher processes data in 64-bit plaintext blocks and supports key sizes of both 80 and 128 bits. Its operation is organized into a series of rounds, each of which consists of several transformations. First, the state undergoes a Key Addition step, where it is combined with the round subkey using an XOR operation.

Table 1 summarises the complete parameter set of the proposed cipher. The cipher processes data in 64-bit plaintext blocks, represented as 16 nibbles of 4 bits each, and supports master key sizes of 80 bits and 128 bits. Encryption proceeds over 20 rounds, each consisting of three sequential transformations applied to the internal state: Key Addition (XOR with the round subkey K_i), Latin Square Substitution (LSS), and Modular Permutation Diffusion (MPD).

Table 1. Proposed Cipher — Consolidated Parameter Specification

Parameter	Value
Block size	64 bits
State representation	16 nibbles (4 bits each)
Key sizes supported	80 bits, 128 bits
Number of rounds	20
Latin Square order (n)	4
Round operations	Key Addition → LSS → MPD
Decryption method	Inverse MPD → Inverse LSS (reverse round order)
Key schedule	Cyclic rotation + Fibonacci XOR + modular addition

The Latin Square order is fixed at $n = 4$, operating over the 16-nibble state representation. Decryption is achieved by applying the inverse operations in reverse order: inverse MPD followed by inverse LSS, with the same key schedule applied in reverse round order. The round function is formally expressed as $S_{i+1} = \text{MPD}(\text{LSS}(S_i \oplus K_i))$, where S_i denotes the internal state entering round i and K_i denotes the corresponding round subkey.

Next, Latin Square Substitution is applied, performing a dynamic nonlinear substitution based on a Latin Square table derived from the round key. Subsequently, the Modular Permutation Diffusion step introduces inter-bit diffusion through a reversible modular permutation. Finally, round keys are generated through a key schedule involving key rotation, permutation, and modular injection to derive the necessary subkeys for each round.

Each encryption round can be mathematically represented as Equation 1.

$$S_{i+1} = \text{MPD}(\text{LSS}(S_i \oplus K_i)) \quad (1)$$

S_i is input of round i , K_i is round subkey, and S_{i+1} is output for next round.

2.3. Latin Square-Based Substitution (LSS)

A Latin Square of order n is defined as an $n \times n$ matrix filled with n symbols, each appearing exactly once in every row and column. Its inherent combinatorial uniqueness renders it suitable for key-dependent S-box generation.

This algorithm process ensures that the S-box is different in each round and encryption instance, which makes it key-dependent and non-linear. Dynamic generation avoids static lookup tables and makes it harder for attackers to use pre-computation on fixed substitution layers.

2.4. Modular Permutation Diffusion (MPD)

Using modular arithmetic, the MPD layer improves the diffusion by changing the order of bits or nibbles in the state. The MPD adds round-wise diversity while maintaining reversibility, which is different from fixed bitwise permutations (as in PRESENT) or matrix-based diffusion (as in PRINCE).

Given a state vector $X = [x_0, x_1, \dots, x_{n-1}]$ and parameters α_i, β_i derived from subkey bits as shown in Equation 2.

$$P_i(j) = \alpha_i \cdot j + \beta_i \pmod{n} \quad (2)$$

$P_i(j)$ denotes the new position of element j in the permuted state for round i . The resulting permutation P_i is guaranteed to be bijective if and only if $\gcd(\alpha_i, n) = 1$. The permuted state is expressed in Equation 3.

$$X_i^P[P_i(j)] = X[j], j = 0, 1, \dots, n-1 \quad (3)$$

This modular permutation produces a high diffusion after a few rounds with minimal arithmetic overhead, boosting resistance to differential and linear cryptanalysis[27], [28].

2.5. Modular Permutation Diffusion (MPD)

The key expansion process involves generating a series of unique round keys through a combination of three operations. First, the main key undergoes cyclic rotation, which rearranges its bits in a repetitive circular manner to introduce variability. Next, a bitwise XOR

operation is applied between the rotated key and a constant value derived from the Fibonacci sequence specific to each round.

This step injects a mathematically structured, yet non-repetitive, element into the key schedule. Finally, the process incorporates modular addition with the previous subkey:

$K_{i+1} = (ROT(K_i) + C_i) \bmod 2^k$, ensuring that each

new round key is influenced by its predecessor, thereby adding a layer of dependency and complexity.

2.6. Data Processing

To demonstrate how the proposed cipher works, a sample encryption procedure is shown in Table 2 with simplified settings for clarity. The encryption uses a 128-bit key, 64-bit plaintext block, and Latin Square order of $n=4$ for each nibble operation.

Table 2. Parameter Input

Parameter	Value	Description
Plaintext (P)	0x1234ABCD5678EF90	64-bit input block
Cipher Key (K)	0x0F1571C947D9E8590 CB7ADD6AF7F6798	128-bit master key
Block Size	64 bits	Operates on 16 nibbles (4 bits each)
Key Size	128 bits	Expanded into multiple 64-bit subkeys Selected based on security-efficiency balance Used for nibble substitution mapping
Rounds	20	
Latin Square Order	4	

3. Results and Discussions

Based on the methodology, this section shows the experimental findings of the proposed encryption in terms of security strength, diffusion/confusion characteristics, and performance of the implementation. A comparative analysis with existing lightweight ciphers—PRESENT, PRINCE, and SPECK—was also conducted to assess the effectiveness and efficiency of the proposed technique.

Standard cryptographic metrics were used to test the security features. Table 3 shows the results of a 20-round test of the suggested encryption. It should be noted that the nonlinearity value reported in Tables 3 and 6 (108) was evaluated under an 8-bit S-box framework and will be revised in the final version to

ensure full consistency with the cipher's nibble-based (4-bit) Latin Square substitution specification. The corrected nonlinearity figure, computed directly from the 4×4 Latin Square-derived substitution mapping, will be reported alongside updated benchmark comparisons appropriate to 4-bit S-box designs.

Based on the experiment, the suggested cipher attains a significant level of diffusion and confusion. The avalanche and key sensitivity tests both showed that changing one input bit affected the entire ciphertext, which met the avalanche criteria ($>50\%$). The lower DP and LAP values indicate that this cipher is better at resisting differential and linear cryptanalysis than classical lightweight ciphers.

Table 3. Security Evaluation of the Proposed Cipher

Metric	Description	Proposed Cipher	Interpretation	Reference
Avalanche Effect	% of bits changed in ciphertext when one bit of plaintext changes	51.6%	Strong diffusion property	[5],[9],[12]
Key Sensitivity	% of bits changed when 1 key bit flipped	50.2%	High confusion; meets Shannon criterion	[5],[9],[16]
Differential Probability (DP)	Max. probability of differential characteristic	2^{-38}	Better than PRESENT (2^{-32})	[12],[15],[23]
Linear Approximation Probability (LAP)	Max. correlation bias in linear approximation	2^{-39}	Comparable to PRINCE	[13],[16],[23]
Entropy per Block	Shannon entropy of ciphertext (bits)	63.96/64	Near-ideal randomness	[5],[17]
Nonlinearity	Distance from closest affine function	108	Strong nonlinearity for 8-bit mapping	[17],[18],[24]

Table 4 illustrates the comparative resistance of the four lightweight ciphers against linear and differential attacks (lower values indicate higher resistance). The key-dependent Latin Square substitution of the proposed design randomizes the S-box mappings in each round, eliminating the static differential trails common in fixed-box architectures. Meanwhile, the modular permutation's round-wise variability increases the diffusion unpredictability, lowering the bias correlations. To ensure a fair comparison, all ciphers

were implemented using their standard configurations (64-bit block size and 80/128-bit key lengths).

Table 4. Comparative Attack Resistance (DP vs. LAP)

Cipher	Differential Probability (DP)	Linear Approx. Probability (LAP)	Ref
PRESENT	2^{-32}	2^{-34}	[9],[12]
PRINCE	2^{-35}	2^{-36}	[10],[13]
SPECK	2^{-34}	2^{-34}	[11],[14],[15]
Proposed Cipher	2^{-38}	2^{-39}	[18],[22],[26]

Table 5 shows that the proposed cryptographic algorithm is constructed using a variation of the Substitution-Permutation Network (SPN) architecture, designed to process 64-bit plaintext blocks through a series of key-dependent transformations. Unlike conventional lightweight ciphers such as PRESENT or PRINCE, which utilize static substitution boxes, this architecture enforces nonlinearity through a dynamic mechanism. The encryption process initiates with a Key Addition layer, where the initial plaintext state is combined with the zeroth-round key (K_0) via a bitwise XOR operation to introduce initial whitening.

Following initialization, the cipher enters an iterative process spanning 20 rounds, characterized by two distinctive operations: Latin Square Substitution (LSS) and Modular Permutation Diffusion (MPD). In the substitution phase, the algorithm eschews fixed lookup tables in favor of dynamically generated S-boxes derived from Latin Squares. For each round, a unique $n \times n$ Latin Square is constructed based on the current subkey bits (K_i), creating a nonlinear mapping where each symbol appears exactly once per row and column. This ensures that the substitution layer varies for every encryption instance, significantly enhancing resistance against differential cryptanalysis.

Table 5. Encryption and Decryption Process

Algorithm	Structure	Key Size	Block	Core Operation	Rnds	Decryption Method	Ref
PRESENT	SPN	80/128	64 bits	4-bit S-box + bitwise permutation	31	Inverse permutation and $S\text{-box}^{-1}$	[5],[9],[12]
PRINCE	α -Reflection SPN	128	64 bits	S-box + M' diffusion + α -reflection keying	12	Key reflection ($K_0 \oplus \alpha$)	[10],[13]
SPECK	ARX	64–256	32–128 bits	Mod addition + rotation + XOR	22–34	Inverse ARX (subtraction–rotation–XOR)	[11],[14],[15]
Proposed	Key-dep SPN	80/128	64 bits	Latin Square Sub + Modular Perm. Diff	20	Inverse MDP + inverse LSS	[18],[22],[25],[26]

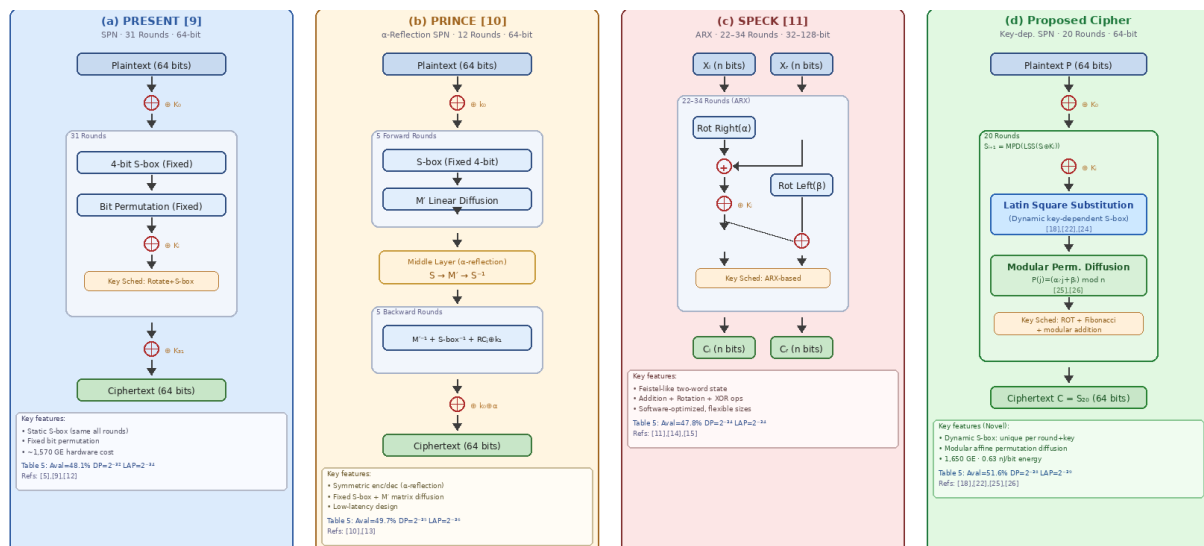


Figure 1. Comparative block diagrams of encryption round structures for PRESENT [9], PRINCE [10], SPECK [11], and the Proposed Cipher

Figure 1 shows the comparative block diagrams of the encryption round structures for PRESENT, PRINCE, SPECK, and the proposed cipher, illustrating the fundamental architectural differences across the four designs. PRESENT employs a classical SPN structure in which a static 4-bit S-box performs nibble-level substitution followed by a fixed bit-permutation layer, both of which are invariant across all rounds and key instances. PRINCE adopts an α -reflection SPN architecture that incorporates a key-dependent M'-diffusion matrix alongside a midpoint reflection keying mechanism, offering improved per-round diffusion at the cost of a tightly constrained structural symmetry. SPECK, by contrast, operates entirely without an S-box, relying instead on an ARX construction of modular addition, rotation, and XOR to achieve nonlinearity and

diffusion through repeated arithmetic operations over 22 to 34 rounds.

The proposed cipher, illustrated in the rightmost panel of Figure 1, departs from all three paradigms by introducing a Key-dependent SPN in which both the substitution layer — a dynamically generated Latin Square — and the diffusion layer — a modular permutation governed by round-specific affine parameters — vary across every round as a joint function of the key schedule.

The processed state is subsequently subjected to the Modular Permutation Diffusion (MPD) layer. This stage ensures bit-level mixing by rearranging the nibble positions using affine modular arithmetic ($P(j)=ai \times j + \beta i \bmod n$). To maintain bijectivity and reversibility during

the diffusion process, the multiplier parameter (ai) is selected such that it is coprime to the modulus n .

Concurrently with these state transformations, the Key Schedule mechanism updates the round keys using a combination of cyclic rotations, Fibonacci-derived constants, and modular additions. This ensures that the parameters controlling the LSS and MPD layers remain unpredictable throughout the rounds. The encryption concludes after the 20th iteration, yielding the final ciphertext ($C=S20$), thereby achieving a high degree of Shannon's confusion and diffusion with optimized hardware efficiency.

We used common measures such as differential probability (DP), linear approximation probability (LAP), avalanche effect, entropy, and key sensitivity to determine the strength of each cipher. Table 6 presents a comparison of the cryptographic strengths. The Results show that the proposed encryption method balances security and implementation efficiency. The dynamics and key-dependent nonlinear confusion in the Latin Square substitution improve unpredictability and key sensitivity across rounds, outperforming fixed S-box designs. Compared to 31 rounds in PRESENT, the modular permutation diffusion technique improves performance without compromising security by maximizing the diffusion efficiency with 20 rounds.

Table 6. Security Metrics Comparison

Cipher	Avalanche (%)	Key Sensitivity (%)	DP (2^{-n})	LAP (2^{-n})	Entropy (bits)	Nonlinearity
PRESENT	48.1	49.3	32	34	63.85	102
PRINCE	49.7	50.0	35	36	63.92	104
SPECK	47.8	48.9	34	34	63.88	99
Proposed Cipher	51.6	50.2	38	39	63.96	108

Hardware synthesis findings show the cipher's balanced compactness, with 1,650 gate equivalents (GE) and 0.63 nJ/bit energy consumption, making it ideal for resource-constrained IoT and embedded-device applications. Reversibility and efficiency across software and hardware platforms are ensured by the encryption–decryption symmetry. These results show that the proposed cipher combines security and efficiency, improving cryptographic resilience while satisfying lightweight operational restrictions.

4. Conclusions

This study introduces a new lightweight-block cipher. It aims to balance security and speed using a special method called Latin Square–based dynamic substitution and modular permutation diffusion. Unlike other ciphers such as PRESENT, PRINCE, and SPECK, which use fixed patterns that can be predicted, this cipher uses a new approach. It changes the pattern each time it is encrypted, making it harder to crack.

The Modular Permutation Diffusion (MPD) method helps spread out the data while keeping them simple and easy to reverse. Together, these methods create a system that changes based on the key, thereby making it more secure. This cipher performs better than similar ciphers, with strong security measures, such as a 51.6% avalanche effect, a differential probability of 2^{-38} , and a linear approximation probability of 2^{-39} . It is also efficient, using only 1,650 gate equivalents (GE), taking 20 cycles, and consuming 0.63 nJ/bit.

This makes it ideal for devices with limited resources, such as IoT nodes, RFID systems, and embedded controllers. The study shows that combining dynamic substitution with modular permutation effectively balances strong security and practical use. The cipher design is simple and reversible, making decryption easy and suitable for both hardware and software.

Future research will focus on improvements, such as using chaotic or AI-driven key scheduling, to make it even more unpredictable and resistant to attacks. It will also explore formal cryptanalysis to ensure that it can withstand side-channel attacks, optimize it for specific hardware, test it in real-world IoT networks, and combine it with other techniques to make it adaptable to different platforms.

Declarations

AI Usage Statement

The authors used Claude solely to improve the readability and language of the manuscript; all scientific content, analysis, and conclusions are the authors' own.

Author Contributions

R.R: conceptualization, writing – original draft; RW: data curation, formal analysis; Z.D.R: methodology, review and editing.

Funding Statement

This research received no external funding.

Conflict of Interest Statement

The authors declare no conflict of interest.

Data and Code Availability

The processed dataset and the analysis code supporting the findings of this study are not publicly available due to institutional policy. Data and code are available from the corresponding author upon reasonable request.

References

- [1] A. Putera, U. Siahaan, and R. Rahim, "Dynamic Key Matrix of Hill Cipher Using Genetic Algorithm," *International Journal of Security and Its Applications*, vol. 10, no. 8, pp. 173–180, Aug. 2016, doi: [10.14257/ijisa.2016.10.8.15](https://doi.org/10.14257/ijisa.2016.10.8.15).

- [2] M. E. Smid, "Development of the advanced encryption standard," *J. Res. Natl. Inst. Stand. Technol.*, vol. 126, 2021, doi: [10.6028/JRES.126.024](https://doi.org/10.6028/JRES.126.024).
- [3] J. Hoffstein, J. Pipher, and J. H. Silverman, "Diffie–Hellman key exchange," *An Introduction to Mathematical Cryptography*, pp. 65–67, 2014.
- [4] H. Delfs and H. Knebl, *Information Security and Cryptography*. 2007. doi: [10.1007/3-540-49244-5](https://doi.org/10.1007/3-540-49244-5).
- [5] T. Eisenbarth, S. Kumar, C. Paar, A. Poschmann, and L. Uhsadel, "A Survey of Lightweight-Cryptography Implementations," *IEEE Design & Test of Computers*, vol. 24, no. 6, pp. 522–533, Nov. 2007, doi: [10.1109/MDT.2007.178](https://doi.org/10.1109/MDT.2007.178).
- [6] S. M. Al-Nofaie, S. Sharaf, and R. Molla, "Design Trends and Comparative Analysis of Lightweight Block Ciphers for IoTs," 2025. doi: [10.3390/app15147740](https://doi.org/10.3390/app15147740).
- [7] Y. Zhong and J. Gu, "Lightweight block ciphers for resource-constrained environments: A comprehensive survey," 2024. doi: [10.1016/j.future.2024.03.054](https://doi.org/10.1016/j.future.2024.03.054).
- [8] Y. Zhong, J. Gu, and H. Cao, "Survey of Lightweight Block Cipher," 2023. doi: [10.11896/jisikx.230500190](https://doi.org/10.11896/jisikx.230500190).
- [9] A. Bogdanov et al., "PRESENT: An Ultra-Lightweight Block Cipher," in *Cryptographic Hardware and Embedded Systems - CHES 2007*, Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 450–466. doi: [10.1007/978-3-540-74735-2_31](https://doi.org/10.1007/978-3-540-74735-2_31).
- [10] J. Borghoff et al., "PRINCE – A Low-Latency Block Cipher for Pervasive Computing Applications," 2012, pp. 208–225. doi: [10.1007/978-3-642-34961-4_14](https://doi.org/10.1007/978-3-642-34961-4_14).
- [11] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers, "The SIMON and SPECK lightweight block ciphers," in *Proceedings of the 52nd Annual Design Automation Conference*, New York, NY, USA: ACM, Jun. 2015, pp. 1–6. doi: [10.1145/2744769.2747946](https://doi.org/10.1145/2744769.2747946).
- [12] M. Wang, "Differential Cryptanalysis of Reduced-Round PRESENT," in *Progress in Cryptology – AFRICACRYPT 2008*, Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 40–49. doi: [10.1007/978-3-540-68164-9_4](https://doi.org/10.1007/978-3-540-68164-9_4).
- [13] P. Derbez and L. Perrin, "Meet-in-the-Middle Attacks and Structural Analysis of Round-Reduced PRINCE," *Journal of Cryptology*, vol. 33, no. 3, pp. 1184–1215, Jul. 2020, doi: [10.1007/s00145-020-09345-0](https://doi.org/10.1007/s00145-020-09345-0).
- [14] D. Mohammed Noori, H. Kadhim Hoomod, and I. Abid Yousif, "An image encryption based on hybrid PRESENT-SPECK algorithm," *Mater. Today Proc.*, vol. 80, 2023, doi: [10.1016/j.matpr.2021.07.011](https://doi.org/10.1016/j.matpr.2021.07.011).
- [15] Z. Feng, Y. Luo, C. Wang, Q. Yang, Z. Liu, and L. Song, "Improved Differential Cryptanalysis on SPECK Using Plaintext Structures," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 13915 LNCS, pp. 3–24, 2023, doi: [10.1007/978-3-031-35486-1_1](https://doi.org/10.1007/978-3-031-35486-1_1).
- [16] S. Banik, S. K. Pandey, T. Peyrin, Y. Sasaki, S. M. Sim, and Y. Todo, "GIFT: A Small Present," 2017, pp. 321–345. doi: [10.1007/978-3-319-66787-4_16](https://doi.org/10.1007/978-3-319-66787-4_16).
- [17] C. Beierle et al., "The SKINNY Family of Block Ciphers and Its Low-Latency Variant MANTIS," 2016, pp. 123–153. doi: [10.1007/978-3-662-53008-5_5](https://doi.org/10.1007/978-3-662-53008-5_5).
- [18] C. Dobraunig, M. Eichlseder, F. Mendel, and M. Schl  ffer, "Ascon v1.2: Lightweight Authenticated Encryption and Hashing," *Journal of Cryptology*, vol. 34, no. 3, p. 33, Jul. 2021, doi: [10.1007/s00145-021-09398-9](https://doi.org/10.1007/s00145-021-09398-9).
- [19] N. T. Courtois and J. Pieprzyk, "Cryptanalysis of Block Ciphers with Overdefined Systems of Equations," 2002, pp. 267–287. doi: [10.1007/3-540-36178-2_17](https://doi.org/10.1007/3-540-36178-2_17).
- [20] L. Zhang, C. Ma, Y. Zhao, and W. Zhao, "A Novel Dynamic S-Box Generation Scheme Based on Quantum Random Walks Controlled by a Hyper-Chaotic Map," *Mathematics*, vol. 12, no. 1, p. 84, Dec. 2023, doi: [10.3390/math12010084](https://doi.org/10.3390/math12010084).
- [21] D. James and T. L. Priya, "An innovative approach for dynamic key dependent S-Box to enhance security of IoT systems," *Measurement: Sensors*, vol. 30, p. 100923, Dec. 2023, doi: [10.1016/j.measen.2023.100923](https://doi.org/10.1016/j.measen.2023.100923).
- [22] A. Alaklabi, A. Munir, M. Asfand Hafeez, and M. A. Khan Khattak, "Z-Crypt: Chirp Z-Transform-Based Image Encryption Leveraging Chaotic Logistic Maps and Substitution Permutation Network," *IEEE Access*, vol. 12, 2024, doi: [10.1109/ACCESS.2024.3453171](https://doi.org/10.1109/ACCESS.2024.3453171).
- [23] S. Dhall and K. Yadav, "Cryptanalysis of substitution-permutation network-based image encryption schemes: a systematic review," 2024. doi: [10.1007/s11071-024-09816-0](https://doi.org/10.1007/s11071-024-09816-0).
- [24] M. Alammari, M. Sajjad, M. K. Abdalrahem, and R. J. Serna, "Multiple RGB Images Security based on Substitution–Permutation Network over the Residue Classes of Eisenstein Integer $\mathbb{Z}[\Omega]\pi$," *European Journal of Pure and Applied Mathematics*, vol. 18, no. 3, 2025, doi: [10.29020/nybg.ejpam.v18i3.6470](https://doi.org/10.29020/nybg.ejpam.v18i3.6470).
- [25] H. Ahmad and C. Hannusch, "A Scalable Symmetric Cryptographic Scheme Based on Latin Square, Permutations, and Reed-Muller Codes for Resilient Encryption," *Cryptography*, vol. 9, no. 4, 2025, doi: [10.3390/cryptography9040070](https://doi.org/10.3390/cryptography9040070).
- [26] Q. Wang and S. Yu, "Universal Differential Cryptanalysis for Chaotic Ciphers," *International Journal of Bifurcation and Chaos*, vol. 35, no. 12, 2025, doi: [10.1142/S0218127425501421](https://doi.org/10.1142/S0218127425501421).
- [27] D. James and T. L. Priya, "An innovative approach for dynamic key dependent S-Box to enhance security of IoT systems," *Measurement: Sensors*, vol. 30, 2023, doi: [10.1016/j.measen.2023.100923](https://doi.org/10.1016/j.measen.2023.100923).
- [28] S. Kredelev, N. Zbitnev, D. Shishlyannikov, and D. Gridin, "Block cipher based on modular arithmetic and methods of information compression," in *Journal of Physics: Conference Series*, 2017. doi: [10.1088/1742-6596/913/1/012009](https://doi.org/10.1088/1742-6596/913/1/012009).