Research on Block Cipher Technology

Yuanhao Yin

School of Computer Science and Technology, Soochow University, Suzhou, China

Abstract. With the rapid advancement of information technology, Humanity will enter the era of the Internet of Things, with billions of IoT devices connected to the network, and network attacks targeting user privacy, network environment, etc. will continue to grow, data security faces growing challenges such as leakage, tampering, and forgery. Block ciphers play a vital role in modern cryptography by encrypting fixed-size data blocks with secret keys, thereby ensuring confidentiality, integrity, and authenticity. This paper provides a comprehensive review of the evolution of block ciphers, tracing developments from the Data Encryption Standard (DES) to the Advanced Encryption Standard (AES). It examines key structural frameworks, including Substitution-Permutation Networks (SPN), Feistel Networks, and Add-Rotate-XOR (ARX) constructions, and evaluates their performance across various application scenarios. In addition, the paper discusses emerging research directions such as post-quantum cryptography, lightweight cipher design for resource-constrained environments, hybrid structural innovations, and automated cryptanalysis techniques, offering valuable insights into the future development of secure encryption algorithms.

1 Introduction

From the birth of computers to the deep integration of internet technology into people's lives, with the development of information technology, computer technology has been widely applied in fields such as science and technology, military, healthcare, and everyday life. People's lives have become more convenient, military technology has advanced, and researchers have entered the new era of the Internet of Things. However, the security issues that come with this development cannot be ignored.

Unlike traditional networked systems such as servers and smartphones, IoT devices—including wireless sensors and implantable medical instruments—operate under severe resource constraints. These limitations include minimal memory capacity, reduced processing power, restricted energy supply, compact physical form factors, and increased exposure to potential security threats. In addition, with the widespread application of information technology, data has become one of the most valuable resources in modern society. However, data transmission and storage often face numerous security threats, such as data tampering, information leakage, and others. As the cornerstone of information security, cryptographic technology effectively addresses these challenges. Through encryption algorithms, cryptography ensures data confidentiality, preventing sensitive

2227406099@stu.suda.edu.cn

information from being stolen; through digital signatures and message authentication codes, it verifies data integrity and authenticity, preventing data from being tampered with or forged; through identity authentication mechanisms, cryptography ensures the trustworthiness of the identities of communication parties, preventing malicious attackers from impersonating legitimate users. Therefore, cryptographic technology is not only the core means of protecting data security but also the key support for building a trustworthy information society. In the era of the Internet of Things, the importance of cryptographic technology is increasingly prominent, and its research and application are of significant importance for safeguarding information security and promoting technological development.

Among the various cryptographic technologies, block ciphers, as a core component of symmetric encryption, play a crucial role. Block ciphers divide plaintext data into fixed-length blocks and use a key to encrypt them, generating ciphertext, thereby ensuring data confidentiality. Due to their efficiency and security, block ciphers are widely applied in various information security scenarios, such as network communication encryption, data storage protection, and hardware security module design. From the classic Data Encryption Standard (DES) [1] to the widely used Advanced Encryption Standard (AES) [2], the design and analysis of block ciphers have always been a focal point of cryptographic research. However, with the advancement of computing power and the emergence of new attack techniques, block ciphers also face new challenges, such as side-channel attacks, quantum computing threats, and more. Therefore, in-depth research on the design principles, security analysis, and optimization of block ciphers in practical applications is of significant importance for advancing the development of cryptographic technology.

This paper introduces the current research status of block cipher algorithms, as well as the risks and challenges they face, and summarizes the latest developments in this field. Additionally, this paper classifies and summarizes some advanced block cipher algorithms. Relevant evaluation metrics are specified to compare and analyze the performance and security of the algorithms. Finally, the paper discusses the possible future research directions in the field of block ciphers.

2 Development and Principles of Block Ciphers

2.1 Historical Development

The research in modern cryptography originated in the 1970s, with the Data Encryption Standard (DES) being the first encryption algorithm designed by IBM and adopted by the United States National Bureau of Standards as the first federal data encryption standard. However, due to various limitations, early cryptographic algorithms were only suitable for specific scenarios, such as KeeLoq, which was designed for remote keyless systems. Although DES laid the foundation for modern block ciphers, its relatively short key length led to its gradual obsolescence. In the 1990s, as the weaknesses of DES were discovered, other block ciphers were developed to improve upon it, including IDEA, Blowfish, Twofish, and others. In the early 21st century, the Rijndael algorithm emerged as the winner, becoming the AES standard, which is now widely used globally and continues to be regarded as secure and reliable.

Entering the 21st century, with the rapid development of technology, the pursuit of cryptographic algorithms that offer higher security and occupy less physical space has become a critical issue. This led to the emergence of research and algorithms focused on lightweight cryptography. The classic PRESENT cipher [3], which occupies a small chip area, fully considers differential and linear analysis, providing sufficient security while also

significantly improving the operation speed. In recent years, research on block ciphers has become more diverse, with significant advancements in areas such as lightweight design, resistance to quantum attacks, hardware optimization, and security analysis. The GIFT cipher [4], introduced in 2017, combines hardware efficiency and security, becoming one of the representatives of lightweight cryptography. Additionally, the rise of quantum computing poses a threat to traditional block ciphers, leading to a focus on developing block ciphers resistant to quantum attacks. As a result, research on post-quantum cryptography has made notable progress. The lattice-based cryptosystem has achieved significant advancements in security, computational efficiency, and key size, especially in the study of problems like NTRU, LWE, RLWE, and MLWE. The CRYSTALS-Kyber algorithm was selected by NIST as one of the post-quantum cryptography standard algorithms, showcasing its potential to resist quantum attacks [5].

2.2 Operating Principle

2.2.1 Basic Concepts

The block cipher divides the plaintext into fixed-length blocks (such as 64 bits or 128 bits) during the encryption process. Each block is encrypted independently, which improves processing efficiency and enhances data control. At the same time, the key length is a key factor in determining the encryption strength. Common key lengths include 128 bits, 192 bits, and 256 bits. The longer the key, the stronger the resistance to brute-force attacks, but it may also increase the computational resources required.

2.2.2 Encryption Process

As shown in Figure 1 and Figure 2. In the encryption process of a block cipher, the plaintext block is first subjected to an initial permutation, which rearranges the data to enhance randomness. This is followed by the core round function stage, which typically includes several key steps: First, multiple subkeys are derived from the main key using a subkey generation mechanism, and one subkey is used in each round. Then, non-linear components such as S-boxes are used for confusion operations, increasing the complexity of the algorithm. Next, permutation or linear transformations are applied to achieve diffusion, making the relationship between plaintext and ciphertext more complex. The entire round function is repeated for multiple rounds (such as 10, 12, or 14 rounds), each using a different subkey. At the end of the encryption process, a final permutation is performed to generate the ciphertext output.

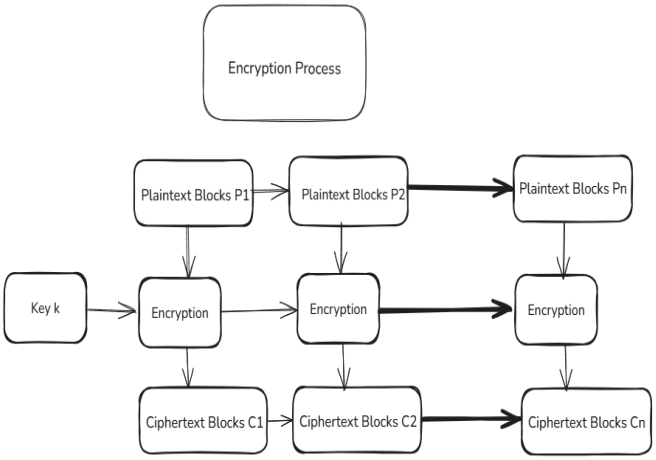

Figure 1. Block Cipher Encryption Process (Picture credit: Original)

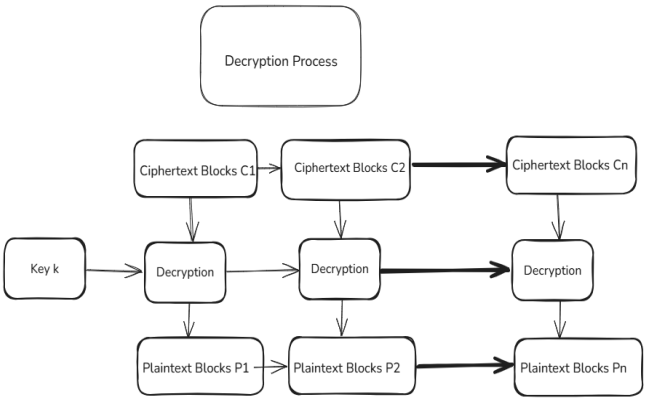

Figure 2. Block Cipher Decryption Process (Picture credit: Original)

2.2.3 Operation Mode

The actual message length is generally greater than the block size of the block cipher. The block cipher divides the message into fixed-length data blocks for processing one block at a time. Many different block processing methods have been designed, known as block cipher operation modes. These are typically combinations of basic cipher modules, feedback mechanisms, and some simple operations.
Table 1 presents the operating modes of block ciphers.

Table 1. Block Cipher Operation Modes

Mode	Describe	Typical Applications
Electronic Codebook (ECB)	Encrypting the plaintext blocks separately with the same key.	Secure transmission of a single piece of data (such as an encryption key).
Cipher Block	Cipher Block	General-purpose

Chaining (CBC)	Chaining.	packet-oriented transmission authentication.
Cipher Feedback (CFB)	"Process j bits at a time, using the previous ciphertext block as input to the encryption algorithm. A pseudorandom number is generated and XORed with the ciphertext to form the input for the next block.	General-purpose block-oriented transmission authentication.
Output Feedback (OFB)	Output Feedback.	Transmission of data streams over noisy channels (such as satellite communication).
Counter (CTR)	Each plaintext block is XORed with an encrypted counter. The counter is incremented for each subsequent block.	General-purpose block-oriented transmission with high-speed requirements.

3 Key Algorithm Description

The mainstream block cipher algorithms can be classified into the following five types based on their internal structure: Substitution-Permutation Network (SPN) structure, Feistel Network (FN) structure, General Feistel Network (GFN) structure, Add-Rotate-XOR (ARX) structure, and Non-Linear Feedback Shift Register (NLFSR) structure. As shown in Figure 3.

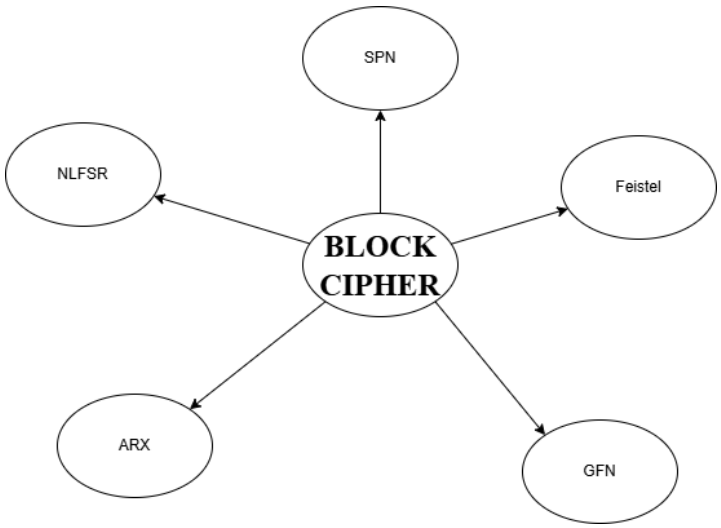

Figure 3 Main Categories of Block Ciphers (Picture credit: Original)

Table 2 presents the key algorithms of block cipher-related structures.

Table 2. Classical Algorithms

Algorithm Structure	Algorithms
SPN	AES、PRESENT、LED、Midori、SKINNY、GIFT、KLEIN、RAIN
Feistel	DES、SIMON、TWINE、LEA、MISTY1、Camellia、SMS4
General Feistel Network	CLEFIA、RC5、RC6、SEED、CAST-256、ARIA、FOX
ARX	SPECK、CHACHA、SALSA20、BLAKE、Threefish
NLFSR	GRAIN、TRIVIUM、MICKEY、ACORN、F-FCSR、ESTREAM

3.1 SPN

SPN is a symmetric cipher structure that was indirectly described by Shannon in his paper through the concepts of "diffusion" and "confusion." Diffusion disperses the statistical properties of the plaintext into the ciphertext through permutation operations, ensuring that local statistical features affect the global structure. Confusion, on the other hand, makes the relationship between the key and ciphertext more complex through substitution operations, preventing direct inference. Shannon's theoretical core is the alternating use of substitution and permutation operations, combined with multiple rounds of iteration and key layering, to achieve sufficient mixing of statistical properties and complex key relationships. This concept directly guided the design of modern symmetric ciphers and became the prototype of the SPN structure.

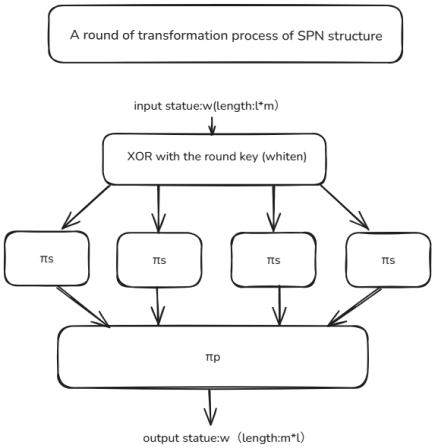

Figure 4. One Round of SPN Transformation (Picture credit: Original)

As shown in Figure 4, the encryption process of this network consists of n rounds. For each round, the input is whitened (XORed with a subkey to eliminate the statistical features of the plaintext). The plaintext is divided into m blocks, each of length l, and substitution is performed according to a specific rule. The substitution rule is called an S-box. After substitution, a permutation operation is applied, and the rule for this permutation is stored in a P-box, which then results in the output state.

Some algorithms belonging to this structure are as follows:

AES [2], proposed by NIST to replace DES [1], is a significant milestone in modern cryptography. Tan Lin [6] improved the full-round related key cube attack and rectangle attack on AES-192, proposing the most probable 10-round related key cube distinguisher, and based on this distinguisher, optimized the time complexity of the full-round AES-192 related key cube attack. Additionally, the optimal rectangle attack results for full-round AES-192 in the chosen plaintext model were provided.

PRESENT is an ultra-lightweight block cipher algorithm that adopts the SPN structure [6]. It is designed for resource-constrained environments, using 64-bit blocks and a 31-round SPN structure, with a 4-bit S-box and bit-level permutation. It supports 64-bit block length and 80 or 128-bit key lengths. J.G. Pandey and others detailed the implementation of the PRESENT algorithm architecture on the Xilinx Virtex-5 FPGA and experimentally validated its performance [7]. Compared to existing architectures, the PRESENT architecture shows significant improvements in FPGA resource usage, latency, and throughput. Specifically, the PRESENT-80 architecture reduced FPGA slice usage by 20.3% and increased throughput by 25.4%; the PRESENT-128 architecture reduced FPGA slice usage by 20.7%, latency by 27.7%, and increased throughput by 69.1%.

LED is a lightweight block cipher that adopts a 64-bit block size and a 32/48-round SPN structure, making it suitable for resource-constrained environments. Research by Guo J [8] and others showed that the LED algorithm successfully achieves an ultra-compact design in hardware (minimizing silicon area) while maintaining reasonable performance in software implementations. It also provides security proofs for single-key and related-key attacks. Through security analysis based on an AES-like structure, LED was shown to have high security margins in both the single-key and related-key models, with the complexity of differential and linear attacks being proven to be much higher than that of exhaustive search attacks. LED uses keyless scheduling (direct reuse of the master key), simplifying hardware implementation, while theoretical analysis proves its security, especially its robustness under related-key attacks [8].

Serpent uses a 128-bit block size and a 32-round SPN structure [9], with 8 4-bit S-boxes used in each round. As one of the candidate algorithms for the Advanced Encryption Standard (AES), Serpent, though not selected as the AES standard, is widely recognized for its high security and conservative design philosophy. As an SPN-based block cipher, it performs an initial permutation on 128-bit plaintext, rearranging the bit order to optimize subsequent bitslice implementations. A total of 32 rounds are performed, and the final output undergoes an inverse initial permutation to generate the 128-bit ciphertext. Ross Anderson found that Serpent's design provides strong security guarantees against various known attacks, such as high-order differential attacks [9], truncated differential attacks, and related-key attacks.

3.2 Feistel

The Feistel cipher structure is a symmetric structure widely used in the design of block ciphers. It was proposed by the American scientist Lawrence Feistel in the 1970s and has been extensively applied in the DES (Data Encryption Standard) algorithm.

The core concept of the Feistel cipher involves structuring the encryption process into two primary operations: substitution and permutation. During the substitution phase, specific parts of the plaintext are transformed into new values using a round function, which processes a portion of the input and produces corresponding output segments. This step introduces non-linearity and enhances confusion. The permutation phase then reorders the elements of the data sequence without altering their actual values—no data is inserted, removed, or replaced—only the positions are shuffled to improve diffusion.

During encryption, the input is a plaintext block of size 2w and a key K. The plaintext is divided into two halves, L0 and R0. After performing n rounds of iteration, the two halves are merged together to produce the ciphertext block. The input for the i-th round iteration is based on the function of the previous round's output:

$$\mathrm{L_i} = R_{i-1} \quad (1)$$

$$R_i = L_{i-1} \oplus F(R_{i-1}, K_i) \quad (2)$$

In this case, K_i is the subkey used in the i-th round, derived from the encryption key K. Generally, the subkeys for each round are different from each other and also different from the key K.

Algorithms belonging to this structure are as follows:

DES [1] is a symmetric-key block cipher algorithm designed by IBM in the 1970s based on the Feistel structure. In 1977, it was adopted as a federal standard by the National Institute of Standards and Technology (NIST's predecessor). Despite its short key length (56 bits), which makes it insecure today, DES remains a milestone in the history of cryptography and has had a profound influence on the development of modern encryption algorithms.

3DES (Triple Data Encryption Algorithm) is an enhanced version of DES that applies DES multiple times to increase security. Due to the 56-bit key of DES being vulnerable to brute-force attacks, 3DES uses two or three different keys to perform three iterations of the DES operation, effectively increasing the key length to 112 or 168 bits (with actual security strengths of 112 bits and 168 bits, respectively). Cooper D. Ad [10] and others discovered that using the Grover algorithm could reduce the key search complexity of 3DES, and quantum computers may pose a threat to its long-term security.

SMS4 is a lightweight block cipher algorithm designed by China [11], primarily used for data encryption in communication systems such as Wireless Local Area Networks (WLAN) and Wireless Metropolitan Area Networks (WMAN). Zhao Yanmin [12] proposed an improved differential attack method, utilizing matrix techniques introduced by Achiya Bar-On and others, combining the nonlinear and linear layers of the SMS4 algorithm to successfully perform key recovery attacks on the 23-round SMS4 algorithm. This method effectively reduced the storage complexity of the attack process by using two auxiliary matrices (Matrix A and Matrix B).

3.3 Generalized Feistel

The Generalized Feistel structure is an extension and generalization of the traditional Feistel structure, designed to provide higher flexibility and stronger security. Zheng Y and others proposed several types of transformations (Type-1, Type-2, Type-3, and Generalized Type-2 transformations), and demonstrated the effectiveness of these transformations in cryptography.

In the Generalized Feistel structure, data is divided into multiple branches (e.g., k branches, each with n bits), whereas the traditional Feistel structure typically divides the data into two branches (left and right halves). The Generalized Feistel structure allows the use of multiple different round functions in each round (for example, in the Type-2 transformation, each odd-numbered branch uses a separate round function), while the traditional Feistel structure generally applies a single round function per round. The Generalized Type-2 transformation, under specific conditions (such as when k is even and ρ

is odd), is considered "optimal," meaning it can achieve security with the fewest rounds. There are many differences between the Generalized Feistel structure and the traditional Feistel structure:

- The traditional Feistel structure divides data into two parts (such as the 32-bit left and right halves in DES).
- The Generalized Feistel structure supports more branches and is suitable for larger block sizes.
- The traditional Feistel structure applies a round function to only one branch in each round, and then simply XORs it with other branches.
- The Generalized Feistel structure (such as Type-2) can apply round functions in parallel on multiple branches, improving parallelism and flexibility.
- The traditional Feistel structure requires at least three rounds to prove security (such as in the Luby-Rackoff theorem).
- The Generalized Feistel structure (like Generalized Type-2) can prove security after k+1 rounds, and increasing the number of rounds (e.g., k+2 rounds) can resist stronger attacks (like chosen plaintext/ciphertext attacks).
- The traditional Feistel structure is designed for fixed block sizes (e.g., 64 bits).
- The Generalized Feistel structure can flexibly adapt to different block sizes and security requirements.

Algorithms belonging to this structure are as follows:

FOX is a new block cipher family based on the Lai-Massey structure [13], which includes both 64-bit and 128-bit block sizes and supports variable rounds and key lengths (up to 256 bits). FOX is highly resistant to linear cryptanalysis and differential cryptanalysis. Sun Ying[14] analyzed the overall structure of FOX128 and proposed a 4-round impossible differential distinguisher that does not rely on the specific properties of the round function (F-function). This distinguisher is applicable to the FOX128 algorithm and does not depend on the specific implementation details of the F-function.

CAST-256 [15] is one of the candidate algorithms for AES, using a Generalized Feistel Structure (Type-1). It divides the data into 4 sub-blocks, and each round applies a nonlinear transformation to two of these sub-blocks. CAST-256 shows high hardware complexity and moderate encryption speed in FPGA implementations, and its design is better suited for software implementations. It is recommended that future cryptographic algorithms simplify the operations required for hardware implementations to enhance efficiency.

Clefia is a lightweight block cipher designed by Sony [16], using a 4-branch Generalized Feistel Structure (Type-2). It supports 128-bit block size and key lengths of 128, 192, or 256 bits. Bogdanov et al. first introduced a 14-round zero-correlation attack. Later research optimized the attack strategy, reducing data complexity. To date, no effective zero-correlation attack has been found for full-round (18/22/26 rounds) Clefia, indicating that Clefia has strong resistance to zero-correlation analysis.

DBST employs a 4-branch GFN structure variant [17], with a block length of 128 bits, a key length of 64 bits, and 32 rounds of iteration. This variant significantly improves diffusion while maintaining the consistency of the Feistel structure. DBST uses bit-slicing technology to dynamically associate the S-box with the key, enhancing the algorithm's security. In hardware implementations, DBST occupies a similar area to SKINNY-128/128, demonstrating its lightweight characteristics.

3.4 ARX

ARX-type cryptographic algorithms refer to symmetric algorithms such as block ciphers, stream ciphers, hash functions, message authentication codes (MACs), and authenticated encryption functions, designed using basic operations such as modular addition (Addition),

rotation (Rotation), and XOR (Exclusive OR). Due to their advantages in terms of efficiency, security, and ease of implementation, ARX-type algorithms have been widely adopted, with several international symmetric cipher standards utilizing the ARX design.

ARX requires only simple operations, making it suitable for both hardware and software implementations, especially in resource-constrained environments (such as IoT devices). Through multiple rounds of iteration, the combination of modular addition and shifting can provide sufficient non-linearity and diffusion while maintaining fixed operation timing, thus reducing the risk of side-channel leakage (compared to S-box-based designs).

Table 3 compares the differences between ARX and Feistel/SPN.

Table 3 ARX vs Feistel/SPN Comparison

	ARX	Feistel	SPN
Core Operations	Addition, Shifting, XOR	Block Splitting+S-box/Permutation	S-box+Permutation Layer.
Source of Non-linearity	Modular Addition	S-box	S-box
Implementation Complexity	Extremely Low	Moderate	High

SPECK belongs to the ARX (Add-Rotate-XOR) structure [18], relying entirely on three operations—modular addition (Addition), rotation (Rotation), and XOR (Exclusive OR)—to construct the encryption process, without the need for S-boxes or permutation tables. This design was proposed by the United States National Security Agency (NSA) in 2013, aiming to provide an efficient and secure encryption solution for resource-constrained devices (such as IoT and embedded systems).

Threefish is an encryption algorithm that is part of the Skein hash function family. It was designed by the developers of the Skein project and is an improvement of Luffa. Threefish is primarily used for encryption and hashing applications. Its design goals are security, efficiency, and scalability. Recent research shows that by only encrypting the most significant bits of image pixels, encryption time and data volume can be significantly reduced while maintaining a high level of encryption quality [19]. However, the algorithm still has room for improvement in resistance to differential analysis, and future work can further optimize the algorithm to enhance its resistance to differential attacks.

LEA is a block cipher algorithm based on the ARX structure (Addition-Rotation-XOR) [20], proposed by South Korean cryptography experts in 2013, and listed as a national standard in Korea (KS X 3246). It is designed for efficient implementation (software/hardware) and is suitable for resource-constrained environments like IoT, mobile devices, and embedded systems. Its key lengths are 128, 192, and 256 bits, and the full-round LEA (24/28/32 rounds) currently has no known practical attacks.

3.5 NLFSR

The Nonlinear Feedback Shift Register (NLFSR) extends the traditional LFSR architecture by incorporating a nonlinear feedback mechanism, thereby improving the cryptographic strength of various algorithms. It finds broad application in lightweight block ciphers, stream cipher systems, and hashing techniques.

An NLFSR typically comprises a sequence of shift registers along with a nonlinear function that governs state transitions, which are updated according to specific feedback rules.

$$s_{t+n} = f(s_t, s_{t+1}, ..., s_{t+n-1}) \oplus c_t \tag{3}$$

In this case:

- st represents the state of the register at time t.
- f is a non-linear Boolean function (such as combinations of AND, OR, XOR).
- ct could be a round key or a constant.

Due to the stronger non-linearity provided by NLFSR, some block ciphers use NLFSR or its variants as part of the round function or key expansion.

The linear feedback of LFSR is vulnerable to algebraic attacks (such as the Berlekamp-Massey algorithm), while NLFSR, by introducing non-linear functions (such as AND, multiplication), can significantly improve security by breaking the linear recursive structure.

NLFSR needs to satisfy high algebraic complexity and balance to avoid linear approximations and statistical biases. This idea has been adopted in subsequent ciphers (such as Grain and KATAN), where a mixed structure of NLFSR and LFSR is used to enhance diffusion. NLFSR has potential in lightweight cryptography but its security is highly dependent on specific implementation and design details. Due to its simplicity in hardware implementation and low resource usage, NLFSR is often used in resource-constrained devices (such as RFID tags and sensor networks). However, the NLFSR structure has certain security weaknesses, such as small internal states that are susceptible to attacks (such as Guess-and-Determine attacks), and some algorithms have been quickly broken due to design flaws (such as A5/1 and A5/2).

The KATAN algorithm is inspired by KeeLoq [21][22]. It offers three block sizes: 32-bit, 48-bit, and 64-bit, with a fixed key length of 80 bits. It requires 254 rounds of iteration and is specifically designed for hardware implementation. The core of KATAN uses two Non-Linear Feedback Shift Registers (NLFSRs), and its key scheduling algorithm is based on Linear Feedback Shift Registers (LFSRs).

KTANTAN shares many similarities with KATAN, such as block size, key length, and the number of iterations [21]. To reduce the complexity of the gate circuit, KTANTAN adopts a round function structure based on NLFSRs. However, unlike KATAN, KTANTAN uses hardcoded encryption keys, and only two bits from the key are used in each round of encryption.

Halka is a lightweight block cipher tailored for environments with limited computational resources [23]. It adopts a 64-bit block and key size with 32 encryption rounds, utilizing a compact 4-bit S-box and a linear diffusion layer to achieve a favorable trade-off between efficiency and security. Suitable for IoT devices, wireless sensors, and RFID tags, Halka offers a practical encryption option. Its key scheduling resembles that of PRESENT, but with an 8-bit S-box in use. Hardware-wise, Halka occupies around 1475 GE, slightly less than PRESENT-80.

4 Performance and Security Analysis

4.1 Performance Evaluation Metrics

Security Metrics: The ability to resist known attacks (such as differential attacks, linear attacks, related-key attacks, side-channel attacks, etc.). This is achieved through confusion (S-box design) and diffusion (permutation layer), which implement the avalanche effect (where a small change in input causes a significant change in output). Additionally, the key length should be sufficiently large (e.g., AES-128/192/256) to resist brute-force attacks.

Efficiency Metrics: The amount of data processed per unit of time (e.g., MB/s), usually tested through software/hardware implementations. The number of clock cycles required for a single encryption or decryption operation (especially important for real-time systems). Parallelization capabilities (e.g., whether the block cipher mode supports parallel processing).

Implementation Cost: The number of gates in ASIC or FPGA implementations, memory usage, etc. The code size in software implementations (important for embedded systems). Suitability for low-power devices (such as IoT terminals).

Flexibility: The speed and memory requirements of the key scheduling algorithm. Whether variable block lengths are supported (e.g., 64-bit, 128-bit). Whether common operation modes are supported (e.g., CBC, GCM, CTR).

Other Metrics: Whether certified by authoritative bodies like NIST (e.g., AES, SM4). Whether the algorithm is open-source or has patent restrictions. Whether it has countermeasures against timing attacks, power analysis, etc. (e.g., masking techniques).

4.2 Performance Analysis of Some Algorithms

Zhong Yue and others summarized the performance evaluations of various block cipher algorithms after software and hardware implementations [24]. Their conclusions are as Table 4 and Table 5:

Table 4. Hardware Implementation Comparison

Algorithm	Throughput	Area Occupancy	Energy Efficiency Performance	Energy Consumption Performance
SPECK	——	High	Low	Low
SIMON	High	High	Low	Low
CHAM	High	High	——	Low
PRIDE	Higher	Moderate	Low	——
mCrypton (D)	Low	Low	High	Moderate
PRESENT	Low	Low	——	High
KTANTAN	Extremely Low	Extremely Low	Extremely High	High
PRESENT-GRP	——	——	——	Extremely High
Hummingbird-2 (D)	——	——	Extremely Low	——

Table 5. Software Implementation Comparison

Algorithm	Throughput	Area Occupancy	Energy Efficiency Performance	Memory Usage
SPECK	High	High	Low	Low
SIMON	High	High	Low	Low
CHAM	High	High	——	Low
PRIDE	Higher	Moderate	Low	——
mCrypton (D)	Low	Low	High	Moderate
PRESENT	Low	Low	——	High
KTANTAN	Extremely Low	Extremely Low	Extremely High	High
PRESENT-GRP	——	——	——	Extremely High
Hummingbird-2 (D)	——	——	Extremely Low	——

In terms of hardware implementation for lightweight block ciphers, Piccolo shows the best overall performance, with advantages of small area, low energy consumption, and high efficiency. Algorithms such as RECTANGLE, SCENERY, LBlock, and DBST achieve a

good balance between area, efficiency, and energy consumption. On the other hand, PRINCE, mCrypton, and CLEFIA are more suitable for scenarios with high throughput requirements, but their hardware resource usage is relatively large.

In terms of software implementation, SPECK, SIMON, and PRIDE excel in throughput, efficiency, and energy consumption, making them particularly suitable for resource-constrained embedded devices. CHAM performs excellently in terms of throughput and memory usage. In contrast, KTANTAN (D) and PRESENT show weaker performance and are not ideal for low-power or real-time application scenarios.

5 Future Research Directions

Looking to the future, the field of block ciphers will continue to evolve in several key directions, laying a solid foundation for security in the digital world. With the advent of the quantum computing era, the development of block cipher algorithms resistant to quantum attacks is becoming urgent. Researchers will explore new mathematical structures and algorithmic principles to ensure that, even under the immense computational power of quantum computers, block ciphers can still protect data confidentiality. At the same time, the integration of artificial intelligence and block ciphers will open up new paths. Machine learning algorithms can be used to optimize key management, enabling smarter key generation, distribution, and updates to enhance the overall security of cryptographic systems. Deep learning models are also expected to help detect anomalous patterns in encrypted data, enabling the timely identification of potential attacks.

With the rapid development of the Internet of Things (IoT) and Industrial Internet, lightweight block cipher algorithms will enter a golden period of growth. Designing efficient yet secure lightweight cryptographic algorithms for resource-constrained IoT devices to meet the strict requirements of low power consumption and low cost is key to ensuring IoT security. This not only requires meticulous algorithm design but also the seamless integration with existing IoT architectures to ensure the security of data transmission and storage between devices.

The standardization process will continue to progress, with global industry organizations and research teams collaborating to develop stricter, unified block cipher standards. This will promote interoperability between different systems and devices, reduce security risks caused by standard differences, and drive the widespread application of block cipher technologies across various fields. As emerging technologies continue to emerge, the field of block ciphers will continue to innovate and break through, providing a solid foundation for building a secure and trustworthy digital ecosystem.

6 Conclusion

This paper provides a systematic review of the development history, basic principles, structural classifications, and typical algorithms of block ciphers. It also conducts a multi-dimensional analysis of their performance and security from both software and hardware implementation perspectives. The research findings show that block ciphers of different structures have their own advantages in terms of security, efficiency, and resource usage, with lightweight algorithms having broad application prospects in resource-constrained environments such as the Internet of Things. In the face of the potential threats posed by quantum computing, the design of quantum-resistant algorithms is becoming an increasingly important research focus. In the future, block ciphers will continue to evolve in the directions of lightweight, intelligent, and standardized design, providing a solid foundation for building a secure and trustworthy information society.

References

1. Diffie, W., Hellman, M. E.: 'Special feature exhaustive cryptanalysis of the NBS data encryption standard.' Computer, 1977, 10(6): 74-84.
2. Bertoni, G., Breveglieri, L., Fragneto, P., et al.: 'Efficient software implementation of AES on 32-bit platforms.' Cryptographic Hardware and Embedded Systems-CHES 2002: 4th International Workshop, Redwood Shores, CA, USA, August 13–15, 2002. Springer Berlin Heidelberg, 2003: 159-171.
3. Bogdanov, A., Knudsen, L. R., Leander, G., et al.: 'PRESENT: An ultra-lightweight block cipher.' Cryptographic Hardware and Embedded Systems-CHES 2007: 9th International Workshop, Vienna, Austria, September 10–13, 2007. Springer Berlin Heidelberg, 2007: 450-466.
4. Banik, S., Pandey, S. K., Peyrin, T., et al.: 'GIFT: A small present: Towards reaching the limit of lightweight encryption.' Cryptographic Hardware and Embedded Systems–CHES 2017: 19th International Conference, Taipei, Taiwan, September 25–28, 2017. Springer International Publishing, 2017: 321-345.
5. Wu, K., Hu, X.: 'A Survey on Post-Quantum Cryptography Technologies.' Computer Science, 2025, 52(02): 8–19.
6. Tan, L., Zeng, X., Liu, J.: 'Related-Key Boomerang and Rectangle Attacks on AES-192.' Journal of Cryptologic Research (Chinese and English), 2024, 11(05): 1018–1028.
7. Pandey, J. G., Goel, T., Karmakar, A.: 'An efficient VLSI architecture for PRESENT block cipher and its FPGA implementation.' VLSI Design and Test: 21st International Symposium, VDAT 2017, Roorkee, India, June 29–July 2, 2017. Springer Singapore, 2017: 270-278.
8. Guo, J., Peyrin, T., Poschmann, A., et al.: 'The LED block cipher.' Cryptographic Hardware and Embedded Systems–CHES 2011: 13th International Workshop, Nara, Japan, September 28–October 1, 2011. Springer Berlin Heidelberg, 2011: 326-341.
9. Anderson, R., Biham, E., Knudsen, L.: 'Serpent: A proposal for the advanced encryption standard.' NIST AES Proposal, 1998, 174: 1-23.
10. Cooper, D. A., Apon, D. C., Dang, Q. H., et al.: 'Recommendation for stateful hash-based signature schemes.' NIST Special Publication, 2020, 800(208): 800-208.
11. Liu, F., Ji, W., Hu, L., et al.: 'Analysis of the SMS4 block cipher.' Australasian Conference on Information Security and Privacy. Springer Berlin Heidelberg, 2007: 158-170.
12. Zhao, Y., Liu, Y., Wang, M.: 'Differential Attacks on the Improved SMS4 Cipher Algorithm.' Journal of Software, 2018, 29(09): 2821–2828.
13. Junod, P., Vaudenay, S.: 'FOX: A new family of block ciphers.' Selected Areas in Cryptography: 11th International Workshop, SAC 2004, Waterloo, Canada, August 9–10, 2004. Springer Berlin Heidelberg, 2005: 114-129.
14. Sun, Y.: 'Four-Round Impossible Differential Distinguishers for FOX128.' Journal of China Academy of Electronics and Information Technology, 2022, 17(12): 1190–1196.
15. Riaz, M., Heys, H. M.: 'The FPGA implementation of the RC6 and CAST-256 encryption algorithms.' Engineering Solutions for the Next Millennium. 1999 IEEE Canadian Conference on Electrical and Computer Engineering. IEEE, 1999, 1: 367-372.
16. Katagi, M., Moriai, S.: 'The 128-bit blockcipher CLEFIA.' IACR. 2011.

17. Yan, L., Li, L., Guo, Y.: 'DBST: A Lightweight Block Cipher Based on Dynamic S-box.' Frontiers of Computer Science, 2023, 17(3): 173805.
18. Beaulieu, R., Shors, D., Smith, J., et al.: 'The SIMON and SPECK lightweight block ciphers.' Proceedings of the 52nd Annual Design Automation Conference, 2015: 1-6.
19. Nori, A. S., Abdulmajeed, A. O.: 'Design and implementation of Threefish cipher algorithm in PNG file.' Sustainable Engineering and Innovation, 2021, 3(2): 79-91.
20. Hong, D., Lee, J. K., Kim, D. C., et al.: 'LEA: A 128-bit block cipher for fast encryption on common processors.' International Workshop on Information Security Applications. Springer International Publishing, 2013: 3-27.
21. De Canniere, C., Dunkelman, O., Knežević, M.: 'KATAN and KTANTAN—a family of small and efficient hardware-oriented block ciphers.' International Workshop on Cryptographic Hardware and Embedded Systems. Springer Berlin Heidelberg, 2009: 272-288.
22. Indesteege, S., Keller, N., Dunkelman, O., et al.: 'A practical attack on KeeLoq.' Advances in Cryptology–EUROCRYPT 2008: 27th Annual International Conference, Istanbul, Turkey, April 13–17, 2008. Springer Berlin Heidelberg, 2008: 1-18.
23. Das, S.: 'Halka: A lightweight, software friendly block cipher using ultra-lightweight 8-bit S-box.' Cryptology ePrint Archive, 2014.
24. Zhong, Y., Gu, J., Cao, H.: 'A Survey of Lightweight Block Cipher Algorithms.' Computer Science, 2023, 50(09): 3–15.