ISSN: 3078-5367 **DOI:** https://doi.org/10.69513/jncs.v3.i1.a8

Al-Noor Journal for Information Technology and Cybersecurity

https://jncs.alnoor.edu.iq/

Lightweight Encryption: A Comprehensive Review of Design Evolution and Architectural Extensions

[1]**Ola Zaid Abd al-majid** and [2]**Yaseen Hikmat Ismaiel**

[1]Department of Computer Science/ University of Mosul/ Mosul - Iraq

Article information

Article history:
Received: 8-4-2026
Revised: 1-5-2026
Accepted: 14-6-2026

Keywords:
Lightweight encryption
Software
Hardware
Authenticated encryption
Standard encryption algorithms
Proposed lightweight encryption algorithms
NIST.

Correspondence:

Ola Zaid Abd al-majid
Email:
ola.24csp36@student.uomosul.edu.iq

Abstract

Identifying abandoned accounts is crucial as they can be breeding grounds for fraud or used as fake accounts, and addressing platform disorder is an important part of sustainable development , Their accumulation causes chaos .This research aims to use the C4.5 algorithm to build a decision tree due to its speed and ability to manage categorical and numerical inputs in identifying features to classify Facebook accounts as active, abandoned, or deactivated by determining the number of days since the last interaction on the account. Data was obtained from a website and manually from other sources. The results of the decision tree demonstrated the discrimination capabilities derived from machine learning. The research presented a solution for recycling or deleting accounts to mitigate the damage caused by the accumulation of abandoned and deactivated accounts on the Facebook platform. The biggest obstacle for the researcher was the sample, as Facebook accounts are subject to privacy laws, and previous literature and studies have not addressed the treatment and recycling of these accounts. The results were satisfactory in the diagnostic process, providing a reliable method to ensure the safety and sustainability of interactions on this platform, which includes billions of users around the world.

1. Introduction:

Lightweight encryption techniques represent a significant research trend, as they are fundamental to protecting the integrity and confidentiality of data in resource-constrained digital systems. Algorithms are designed to support these limited environments and their specific needs, relying on fewer cycles, simplified operations, and uncomplicated calculations, among other features that make them lightweight without compromising security[1]. Numerous algorithms have emerged, some of which have been selected as standard. With the rapid development of IoT devices, the need for algorithms that support these advancements has increased [2]. This has led to the emergence of many proposed algorithms that not only provide encryption or improve upon previous algorithms but also add reliability to enhance security and meet the demands of today's technological revolution[3][4].

2. Early algorithms for Lightweight Encryption:

Early lightweight encryption algorithms offer a direct solution to the security challenges posed by resource-constrained environments, such as the Internet of Things, wireless sensor networks, and RFID tags. These algorithms focus on balancing performance and security by relying on simple calculations, XOR logic, transpositions, permutations, and simplified structures[5]. Many of them were developed through formal initiatives by international standards organizations to standardize security solutions for embedded systems[2]. The following sections will describe these algorithms according to their type: Block Cipher, Stream Cipher, and Hash Faction[4].

ISSN: 3078-5367 **DOI:** https://doi.org/10.69513/jncs.v3.i1.a8

A. Block Cipher :

Lightweight block encryption algorithms were developed to operate in resource-constrained environments in terms of hardware, power, and memory. Most algorithms rely on simple architectures to minimize complex computations and achieve hardware and software efficiency[6][7]. We will review some of these algorithms in chronological order of their development.

Hong et al. (2006) proposed the HIGHT algorithm to provide an algorithm that supports low-resource hardware devices, distributed computing devices, and devices that deal with very simple processors. The algorithm was distinguished by its dispensing with S-boxes and the use of simple operations such as XOR, ADD, mod 2^8, and Rotation over 32 rounds, as shown in Figure 1, which made its implementation easier. It was adopted in ISO/IEC 18033-3, but it suffered from the 128-bit key length and the small 64-bit block size in facing block collision attacks [8].

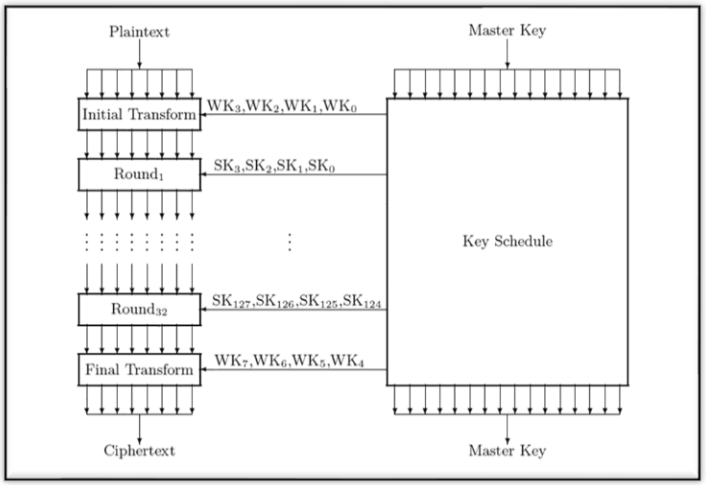

Figure 1. Encryption process of HIGHT[8].

Bogdanov et al. (2007) proposed the PRESENT algorithm, one of the first lightweight encryption algorithms, And as part of ISO/IEC 29192-2 standard, designed to address the encryption problem in devices with strict limitations. Traditional encryption algorithms often fail to achieve acceptable logic gate sizes for such devices. PRESENT tackles high energy and resource consumption by employing an SP-Network architecture with a 4-bit S-boxes, a simple Wiring-based swap layer to reduce computational complexity, and a small key length of 80 or 128 bits, as illustrated in Figure 2. The algorithm demonstrated high hardware efficiency, achieving approximately 1570 gate equivalents, making it the de facto hardware standard for lightweight encryption. It also proved capable of withstanding traditional attacks. However, it suffered from relatively slow programmatic performance compared to newer algorithms. Furthermore, the small 64-bit block size raised concerns about potential future attacks[9].

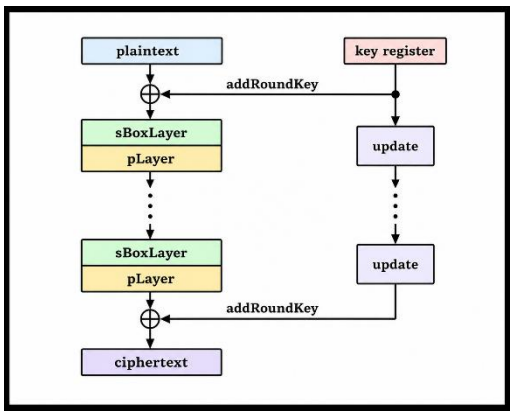

Figure 2. A top-level algorithmic description of present

Taizo Shirai et al. (2007) proposed a new block cipher, CLEFIA, from Sony Labs to address the limitations of lightweight ciphers and to provide flexible key sizes for electronic devices. This approach also addressed the need for new block ciphers based on the latest technologies. The algorithm tackled the number of iterations by reducing the number of rounds and using DSM technology for attack resistance. Based on an extended Feistel architecture, the algorithm offered flexible key lengths of 128, 192, and 256 bits with a block size of 128 bits. The algorithm demonstrated its ability to balance performance and security and exhibited high hardware and software implementation efficiency, leading to its adoption in the ISO/IEC 29192-2 standard. However, it required

ISSN: 3078-5367 **DOI:** https://doi.org/10.69513/jncs.v3.i1.a8

slightly more hardware space than the PRESENT algorithm, limiting its use in small applications [10].

Beaulien et al. (2013) proposed the SIMON algorithm to address the problem of poor algorithm on resource-limited devices, particularly small hardware, and to provide greater flexibility for larger groups of devices. The algorithm's structure is based on a balanced Feistel architecture with simple operations such as AND, XOR, and rotation performance, as illustrated in Figure 3. The algorithm achieved outstanding results in terms of power and space consumption and was characterized by its efficiency and flexibility in terms of block and key sizes and low memory consumption because it does not use S-boxes. However, it suffered from poor performance on processors from a technical standpoint. One of the technical issues related to the algorithm is that the agency adopting and developing the algorithm, the NSA, raised concerns among researchers[11].

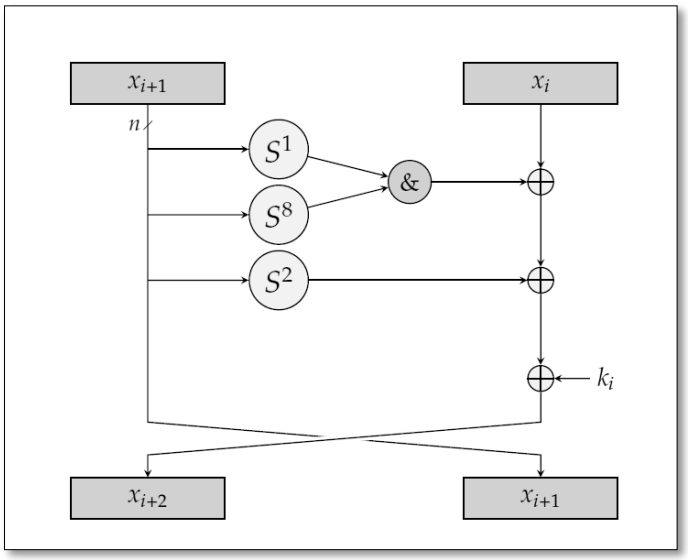

Figure 3. Feistel stepping of the SIMON round function[11].

Beaulien et al. (2013) proposed the SPECK algorithm in conjunction with the SIMON algorithm to address the performance issues of SIMON, as it is considered its sister algorithm. The algorithm relies on the ARX architecture to achieve better processor performance using simple, computationally inexpensive operations, as illustrated in Figure 4. It was distinguished by its superior speed compared to other algorithms, but it faced security concerns from its adopter, the NSA [11].

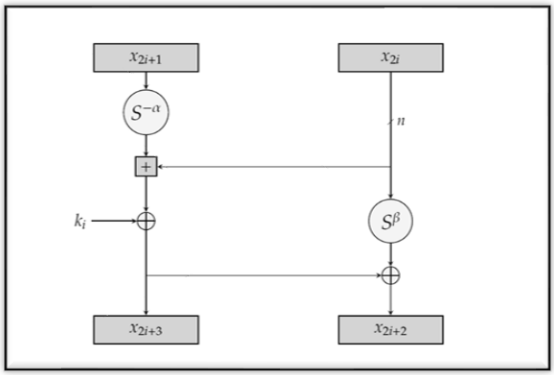

Figure 4. SPECK round function; (x2i+1, x2i) denotes the subcipher after i steps of encryption[11].

B. Stream Cipher:

Lightweight stream encryption algorithms are suitable for restricted environments due to their structural simplicity and high efficiency in resource consumption. However, their structural simplicity has made them sensitive to some attacks. We will discuss these algorithms according to the years of their emergence[12].

Hell et al. (2006) proposed the Grain algorithm for use in hardware environments where the number of gates, memory, and power consumption are very limited, while at the same time a continuous and fast

ISSN: 3078-5367 **DOI:** https://doi.org/10.69513/jncs.v3.i1.a8

data flow is required. This algorithm was distinguished by allowing the user to determine the code speed based on the amount of hardware available, with the integration of LFSR with NFSR to increase complexity and randomness; as in Figure 5, the algorithm achieved outstanding hardware performance results in very small spaces, which qualified it to be one of the designs of choice in eSTREAM. However, it suffered from relative slowness in the initialization state, as feedback registers cannot be applied immediately, and some theoretical attacks appeared on low-cycle versions[13].

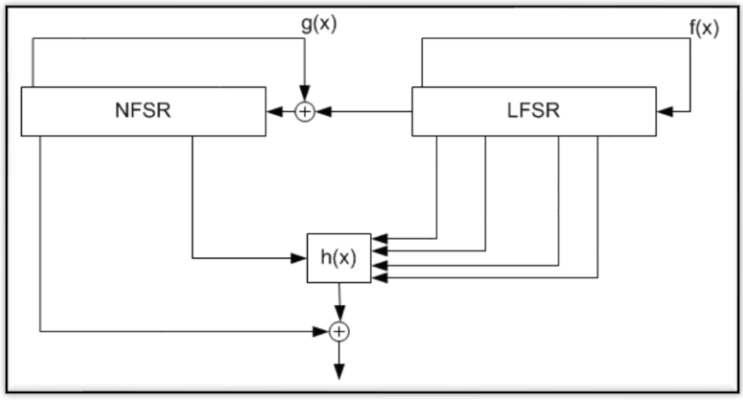

Figure 5: The cipher[13].

de Canniere and Preneel (2006) proposed the TRIVIUM algorithm with the aim of building a flow cipher based on block cipher design principles, i.e., binary assembly ciphers geared towards the hardware. The algorithm was distinguished by its simple and flexible design using only NAND and XOR gates, as shown in Figure 6. It achieved outstanding results and was considered a benchmark for efficiency in its class, as it had a very high flow rate. However, it suffered from a large 288-bit internal state, which leads to increased initialization time and may not be suitable for micro-applications [14].

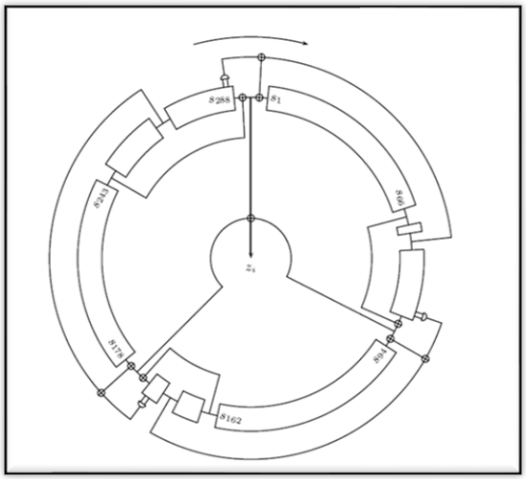

Figure 6: TRIVIUM [14].

Babbage and Dodd (2006) proposed the MICKEY 2.0 algorithm, an improved version of the algorithm that optimized time consumption and resistance to known attacks. The algorithm was designed for devices with low footprint and low power, relying on a combination of LFSR and FSM. It was characterized by the use of irregular clock technology to control register updates, which created complexity between state and output to resist certain types of side attacks. The algorithm achieved outstanding results in terms of security, performance, and good efficiency in resource-limited environments compared to its first version, but it suffered from complexity in the control circuits, which could lead to power consumption[15].

C. Hash Encryption:

It is a key element in lightweight encryption for ensuring data integrity, authentication, and digital signatures in resource-limited environments because it relies on simplified structural designs, simple logical relationships, and limited iterations to reduce the number of computations and implementation

ISSN: 3078-5367 **DOI:** https://doi.org/10.69513/jncs.v3.i1.a8

complexities. We will discuss these algorithms according to their years of emergence [3][1][16].

Guo et al. (2011) proposed the PHOTON family to address the lack of lightweight hashing systems in the research community that could operate in restrictive hardware environments. Based on the Sponge architecture (as shown in Figure 7) and using S-boxes derived from AES but modified for the algorithm's intended purpose, it achieved excellent results as the smallest hardware footprint for a hashing function. However, it suffered from a low processing rate due to the small rate of the architecture, and the Sponge architecture offered a lower level of security than perimage[17].

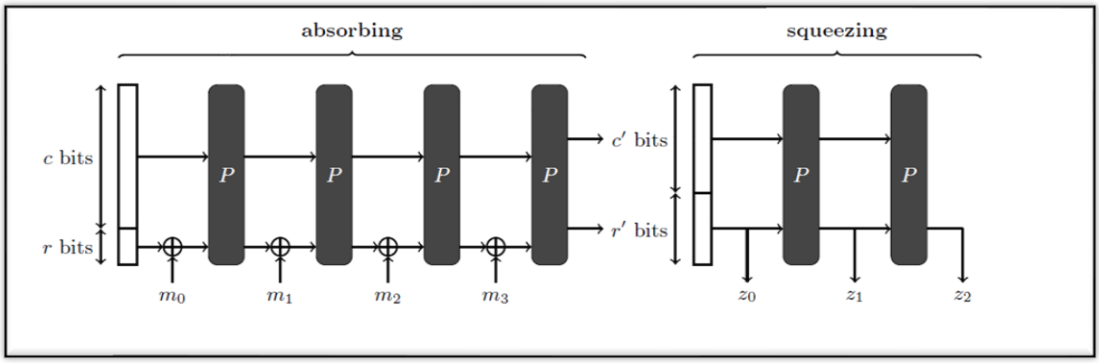

Figure 7: The extended sponge framework, the domain

Bogdanov et al. (2011) proposed the SPONGENT function to reduce the cost of manufacturing RFID cards, which require digital signatures and authentication. The sponge architecture was combined with the present algorithm's compensation box, as shown in Figure 8, which simplified the design. It achieved excellent results, being one of the lightest hash functions in terms of logic gates. However, it suffered from large cycles per block of data, making it slow and unsuitable for real-time applications [18]. Tables 1 and 2 show a comparison between the early algorithms for Lightweight Encryption.

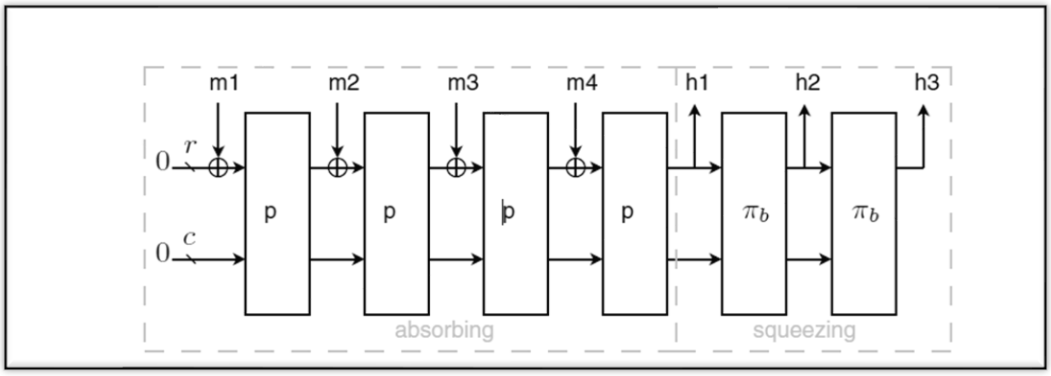

Figure 8: Sponge construction based on a b-bit permutation πb with capacity c bits bits and rate r bits. mi are r-bit message blocks. hi are parts of the hash value[18].

Table 1: Comparison of early algorithms for Lightweight Encryption.

Algorithm	Type	Space (GE)	Block Size (bits)/state	Key Size (bits)	Hardware Efficiency	Software Efficiency	Memory Consumption	Execution Speed	NO. of rounds
HIGHT	Block Cipher	3048 ~	64	128	high	Medium	low	Good	32

ISSN: 3078-5367 **DOI:** https://doi.org/10.69513/jncs.v3.i1.a8

PRESENT	Block Cipher	1570 ~	64	80/128	high	low	low	Slow in software	31
CLEFIA	Block Cipher	~4950	256/192/128	128	Medium - high	Medium	Medium	Good	18/22/26
SIMON	Block Cipher	<1000 low	256-64	32-128	Very high	Medium	Very low	Excellent	Depending on the size of the key and block
SPECK	Block Cipher	~1000	256/64	32-128	high	Very high	Very low	Superior	22-34
Grain V1	Stream Cipher	~1294	1	80	Very high	Medium	Very low	Very high	160
TRIVIUM	Stream Cipher	1800 ~	1	80	Very high	high	low	Very high	160
MICKEY 2.0	Stream Cipher	3180 ~	1	80	high	Medium	Medium	Medium	160
PHOTON	Hash Function	1120 ~	/224/160/128/80 256	-	high	low	low	Medium	12
SPONGENT	Hash Function	~738	8/128/224/160/8 256	-	Very high	low	low	low	It depends on the size of the case.

Table 2: Feature and Challenge of early algorithms for Lightweight Encryption.

Algorithm	Feature	Challenge
HIGHT	Using simple processes and eliminating S-boxes	Limited size - Poor software performance - Weak side protection
PRESENT	Simplicity of design	Slow software due to the use of bit-wise mechanisms
CLEFIA	High efficiency in software and hardware	Structural complexity - hardware and power costs - limited availability in low-volume versions - unsuitable for ultra-small devices
SIMON	Smallest size, flexible and efficient hardware	Software slowness, security concerns, weaknesses in some smaller versions, lack of standardization
SPECK	Software excellence combined with high hardware efficiency	International rejection, political controversy, Backdoor concerns, weaknesses in some smaller versions, lack of standardization
Grain V1	It relies on displacement records - very fast in gear - difficult to break in traditional brute force attacks	Relative slowness in the initialization state

ISSN: 3078-5367 **DOI:** https://doi.org/10.69513/jncs.v3.i1.a8

TRIVIUM	High data flow speeds on small devices - We rely on large conversion registers - Simple architecture - Resistant to known attacks	Larger internal capacity - longer setup time
MICKEY 2.0	It was distinguished by its use of irregular clock technology to control the updating of the recorders, which created complexity between the state and output to resist certain types of lateral attacks.	The complexity of the control circuits, the increased dynamic power consumption, and the sensitivity to IV reuse.
PHOTON	The design of Sponge and the use of S-boxes are derived from	Low processing rate due to the small rate of the structure, and the sponge structure offers a lower level of safety.
SPONGENT	Leveraging PRESENT's features to reduce space	Potential vulnerability to some analytical attacks on early versions, and the difficulty of balancing performance and security.

3. Proposed/Modern Algorithms for Lightweight Encryption:

In recent years, a large number of proposed algorithms have emerged, designed not necessarily as standards but as solutions to meet specific requirements in highly restrictive environments, such as Internet of Things (IoT) systems, low-power embedded devices, and wireless identification cards[19]. Recent research, particularly within the NIST and eSTREAM projects, is moving towards adopting the Authenticated Encryption with Associated Data (AEAD) concept, which integrates encryption and authentication into a single step to reduce processing time[20][3]. Proposed algorithms for lightweight encryption focus on simplifying the algorithm's internal structure and reducing the number of complex calculations. They often rely on lightweight structures such as Feistel, simple addition, rotation, and XOR (ARX) calculations, or linear and nonlinear shift registers in streamlined encryption algorithms. This simplification contributes to lower memory and power consumption and reduces the hardware space required for implementation[2].

Liu et al. (2019) proposed proposed the Loong family of algorithms to solve the space problem in integrated circuits. The algorithm was designed to be reversible, meaning the decryption process is the same as the encryption process. It uses an SPN structure with S-boxes and a diffusion layer designed to be reversible. It was characterized by saving space because in the encryption and decryption process it uses the programming code or the logic circuit, which reduces the cost and memory consumption in small devices. However, it suffered from mathematical constraints that reduced the diffusion efficiency compared to algorithms with non-reversible components, which led to an increase in rounds to ensure security, which affected the speed[21].

Sebastien Riou (2019) proposed the DryGASCON algorithm, an improved version of the GASCON algorithm, to address the problem of side attacks while maintaining the efficiency of the original algorithm. It is based on a sponge structure, as in the original algorithm. The improvement was in handling the internal state and data confidentiality in reducing the physical leakage of information, as the update function was designed to be constant in terms of energy consumption or random, which is difficult to analyze. The algorithm was characterized by higher resistance to energy attacks without the need for complex and costly masking, but it suffered from a slight increase in the size of the circuit and a decrease in speed compared to the original version[22].

researcher Banik et al. (2020) proposed the WARP algorithm to solve the security problem in lightweight algorithms with small 64-bit data blocks. The algorithm's architecture is based on the generalized Feistel network architecture. It deals with a 128-bit data block and a 128-bit key. The algorithm uses 4-bit S-Boxes and switching operations, and consists of 41 rounds to achieve the required security.The algorithm was distinguished by achieving outstanding results, as it was considered the smallest 128-bit block encryption algorithm in terms of size, making it one of the most suitable algorithms for Internet of Things devices. However, due to its GFN architecture and the number of rounds of 41, it suffered from a delay that was higher compared to some algorithms, and this delay is unsuitable for environments that require an immediate response[23].

Kim et al. (2020) proposed the PIPO algorithm to solve the problem of cost and difficulty in using Masking against side-channel attacks in software.

ISSN: 3078-5367 **DOI:** https://doi.org/10.69513/jncs.v3.i1.a8

The algorithm was designed based on the SPN architecture with a BitSlice design and 8-bit S-Boxes to facilitate logical and parallel arithmetic operations. This algorithm supports data with 64-bit blocks and 128/256-bit keys. The algorithm achieved outstanding results in terms of performance against power and speed attacks compared to other standard algorithms, but it suffered from lower hardware efficiency in terms of space compared to algorithms designed specifically for hardware[24].

Daemen et al. (2020) proposed the Subterranean algorithm to provide highly efficient authenticated encryption in very small hardware devices, focusing on reducing the size of the electronic circuit. The algorithm relied on a sponge architecture and an internal state of 257 bits, and it relied on very simple operations. Its operation was at the bit level rather than the word level, so it dispensed with complex S-Boxes and relied on simple logical operations to update the state. It showed outstanding results in terms of very small silicon size and high speed in hardware execution, but it suffered from programming performance because it operates at the bit level, so it is inefficient compared to word-oriented algorithms[25].

Bassam et al. (2021) proposed the SLIM algorithm to address the need for an algorithm that works in the strict environments of medical IoT devices. The algorithm relied on a lightweight Fistel network to handle small 32-bit data blocks and an 80-bit key, and used simple logical operations such as XOR and simple 4x4 s-boxes to reduce computational complexity. The algorithm was characterized by its low power consumption and extremely small footprint, and achieved results resistant to differential and linear attacks within its range of use. However, it suffered from the small 32-bit block size. This meant the algorithm was vulnerable to dictionary attacks if it was encrypted using the same key[26].

G.Li et al.(2021) proposed the Shadow algorithm to solve the power consumption problem in IoT nodes with limited batteries, in addition to solving the problem of arx code defects that spread half a block in a single round. It is from the SIMON-Like family and relies on a simplified Feistel architecture with the use of AND, Rotate, and XOR operations only. It supports 32/64 bit blocks. The key generation table was simplified to reduce the memory required. It obtained results with lower power consumption on microprocessors, which extends the battery life of sensors, but it suffered from security if used for very long periods with the same key[27].

Chen et al. (2022) proposed the SAND algorithm to solve the problem of performance imbalance in software and hardware with low footprints. The algorithm relied on the ARX approach but improved by AND-RX, by replacing the Add operation with AND. This helps to simplify the architecture from the hardware perspective. The algorithm relied on a Feistel architecture with 64/128 bit blocks. The algorithm was characterized by the simplicity of operations, high efficiency in hardware implementation, and very small silicon footprint, with fast program performance on modern processors. However, it suffered from slow propagation speed in the first rounds, which leads to security risks[28].

Gupta et al.(2022) proposed the FUTURE algorithm to improve the propagation speed of lightweight algorithms while minimizing the number of rounds required for security. The algorithm is a block cipher based on a SPN architecture with a block size of 64 bits and a 128-bit key. A propagation array was added to maximize obfuscation. and the algorithm achieved high execution efficiency with the fewest possible rounds. The algorithm successfully balanced security and speed, outperforming some other algorithms in execution speed thanks to the added obfuscation array. However, it suffered from weaknesses in differential analysis resistance due to the low number of rounds[29].

Dobraunig et al.(2023) proposed the ASCON family of algorithms, which were adopted as the basis for lightweight encryption in the 2023 NIST competition to address the lack of a universal standard for lightweight encryption. The algorithm is built on a sponge architecture and uses SPN-based permutations, a 320-bit internal state, and 5-bit s-boxes with a linear diffusion layer.The algorithm is characterized by balanced hardware and software performance, providing reliable encryption and hashing functions. It has shown excellent results in resisting side-resistance attacks, very low power consumption, and a small silicon footprint in embedded applications. However, it has faced challenges in environments that require extremely high speeds[16].

vanzi et al. (2023) proposed a second version of the QARMAv2 algorithm due to the problems suffered by the first version and to improve security for applications that require modifiable encryption, such as memory encryption and pointer authentication. The algorithm was built on the SPN architecture as it is a modifiable block encryption algorithm in which the deployment matrices were redesigned and the s-Boxes were changed to increase resistance against advanced cryptographic analysis. It was characterized by providing a low response time and stronger security limits than its predecessor, but it suffered from being intended to be part of the processor architecture and not for encrypting files or general network communications, which limited its use and specialization to physical and memory applications[30].

ISSN: 3078-5367 **DOI:** https://doi.org/10.69513/jncs.v3.i1.a8

Chen et al. (2023) proposed a development of the standard SAND algorithm by proposing the SAND-2 algorithm to solve the problem of the encryption output of the first four rounds of the original algorithm, in addition to the large number of rounds that indicates a slow propagation process. It did not change the structure of the standard algorithm, which is AND-RX, but it made modifications to the switching function and the key table and reduced the instructions in each round. The algorithm was distinguished by achieving full propagation within 4 rounds and achieved improved results in terms of throughput and less RAM consumption, so it became one of the most suitable algorithms for applications that need to work in real time, but it suffered from more mathematically difficult security than the traditional XOR algorithm[31].

Gupta et al. (2025) proposed the EE-LBC wireless algorithm for solving power sensors in networking. The algorithm's structure is based on a hybrid architecture that helps to use S-Boxes dynamically. It relies on a key-dependent approach to avoid the need for common assistance for a nonlinear benefit. The size is 64 bits and the key is 80 bits. The algorithm stands out as less of a comparison with other algorithms in dynamic social network simulation environments[32]. Tables 3 and 4 show a comparison between modern lightweight encryption algorithms.

Table 3: Structural standards and implementation efficiency for modern lightweight encryption algorithms.

Algorithm	Type	Space (GE)	Block Size (bits)/state	Key Size (bits)	Hardware Efficiency	Software Efficiency	Memory Consumption	Execution Speed	NO. of rounds
Loong	Block Cipher	~1467-1766	64	64/80/128	Very High	High	Very Low	High	16/20/32
DryGASCON	AEAD	~2900 - 03200	-320 640	128-256	High	Very High	Low	High	19-22
WARP	Block Cipher	~800	64 / 128	128	Very High	Medium	Low	Medium	41
PIPO	Block Cipher	~1200-1400	64	128-256	High	Very High	Very Low	High	13
Subterranean 2.0	AEAD	~738	257	128	Very High	Low	Very Low	Very Fast (HW)	1
SLIM	Block Cipher	~635	32	80	Very High	High	Very Low	High	32
Shadow	Block Cipher	~836 - 1688	32-64	64 / 128	High	High	Very Low	High	16-32
SAND	Block Cipher	~2200 3500 -	128/64	128	Medium	High	Low	Very High	32
FUTURE	Block Cipher	~150 - 01700	64	128	Very High	High	Low	Very High	10
Ascon	AEAD / Hash	~3000	320	128	Very High	High (Balanced)	Low	High	8/12

ISSN: 3078-5367 **DOI:** https://doi.org/10.69513/jncs.v3.i1.a8

QARMAv2	TBC	~9650	64-128	128-256	Very High	Medium	Low	Very High	16-20
SAND-2	Block Cipher	~3500-5000	64 / 128	128	High	High	Very Low	Very High	32
EE-LBC	Block Cipher	~259	64	80	Very High	High	Very Low	Medium	31

Table 4: Feature and Challenge of modern lightweight encryption algorithms.

Algorithm	Feature	Challenge
Loong	Involutional structure shares the same circuit for Encryption/Decryption.	Designing strong diffusion is complex in involutional structures.
DryGASCON	Built-in (Dry) resistance to Power Analysis attacks (SCA).	Slightly slower and larger area compared to the standard Ascon.
WARP	Smallest block cipher supporting 128-bit blocks (replaces 64-bit blocks).	High latency due to the large number of rounds (41).
PIPO	Efficient implementation of Higher-Order Masking in software.	Bitslice design consumes more area in dedicated hardware than nibble-based ciphers.
Subterranean 2.0	Ultra-lightweight efficiency for RFID/Smart cards.	Slow and inefficient implementation on standard CPUs (software).
SLIM	Extremely low power consumption (ideal for IoHT/Implants).	Small block size (32-bit) is vulnerable to Sweet32 collision attacks on large data.
Shadow	Optimizes battery life for resource-constrained IoT nodes.	Relies on a slim security margin to achieve efficiency.
SAND	Combines small hardware area with fast software (SIMD) using AND-RX.	Algebraic security analysis is difficult due to the AND-RX logic.
FUTURE	Uses an Optimal Diffusion Matrix for high security in fewer rounds.	Relatively new; requires more independent cryptanalysis.
Ascon	Balanced performance & robust Side-Channel Attack (SCA) resistance.	Not the highest throughput for massive data transfer.
QARMAv2	Ultra-low latency, specifically designed for memory encryption (PAC).	Specialized for hardware hardening; not intended for general-purpose use.
SAND-2	Optimized version for Industrial IoT (IIoT) real-time performance.	Inherits the same algebraic analysis complexities as v1.
EE-LBC	Maximum energy efficiency for Wireless Sensor Networks (WSN).	Reliance on dynamic S-boxes may introduce side-channel vulnerabilities.

ISSN: 3078-5367 **DOI:** https://doi.org/10.69513/jncs.v3.i1.a8

4. Conclusion

In our study, we reviewed lightweight encryption methods, including early and recently proposed methodologies. We conducted an analytical comparison of the algorithms of each type, examining their advantages and disadvantages and identifying their structural frameworks. The second part of the study provided a comprehensive overview of the last years, reviewing proposed algorithms and the optimization mechanisms added to improve their efficiency and adapt them to the continuous growth and development of resource-constrained applications.

In conclusion, we can deduce that the criteria used to evaluate the efficiency of lightweight encryption algorithms are: Number of Rounds, Execution Speed, Memory Consumption, Software Efficiency, Hardware Efficiency, Key Size (bits), Block Size (bits)/State Space (GE). Furthermore, we identified a set of shortcomings and challenges facing the design of lightweight encryption algorithms, most notably security, attack resistance, size, power consumption, and implementation in both hardware and software. Finally, an efficient selection for LWC algorithms must balance between the security requirement and the simplicity and speed of the LWC.

5. Refrence

[1] A. Biryukov and L. Perrin, "State of the Art in Lightweight Symmetric Cryptography," *IACR Cryptol. ePrint Arch.*, pp. 1–55, 2017, [Online]. Available: https://eprint.iacr.org/2017/511

[2] M. Katagi and S. Moriai, "Lightweight Cryptography for the Internet of Things," in *Workshop on Cryptographic Hardware and Embedded Systems (CHES)*, Springer, 2008, pp. 7–10.

[3] A. Encryption, E. O. Functions, K. A. Mckay, J. Kelsey, and K. A. Mckay, "NIST Special Publication 800 NIST SP 800-232 Ascon-Based Lightweight Cryptography Standards for Constrained Devices NIST SP 800-232 Ascon-Based Lightweight Cryptography Standards for Constrained Devices," Gaithersburg, MD, 2023.

[4] T. S. Preview, "INTERNATIONAL STANDARD ISO / IEC Information security — Lightweight Block ciphers iTeh STANDARD iTeh STANDARD PREVIEW," 2019, *International Organization for Standardization, Geneva, Switzerland*.

[5] T. Eisenbarth and L. Uhsadel, "A Survey of Lightweight- Cryptography Implementations," *IEEE Des. Test Comput.*, vol. 24, no. 6, pp. 522–533, 2007, doi: 10.1109/MDT.2007.178.

[6] P. Huynh, "Design and Analysis of Lightweight Encryption Schemes," Université de Lorraine, 2020. [Online]. Available: https://hal.archives-ouvertes.fr/tel-03086269/

[7] H. Brekke, "State of the Art in Lightweight Symmetric Cryptography," *Iahr 2012*, no. Lofgren 2015, pp. 1–11, 2012.

[8] D. Hong *et al.*, "HIGHT : A New Block Cipher Suitable for Low-Resource Device," in *Cryptographic Hardware and Embedded Systems - CHES 2006 (Lecture Notes in Computer Science, vol 4249)*, Springer, Berlin, Heidelberg, 2006, pp. 46–59.

[9] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, and A. Poschmann, "PRESENT : An Ultra-Lightweight Block Cipher," Vienna, Austria: Springer, Berlin, Heidelberg, 2007. doi: 10.1007/978-3-540-74735-2_31.

[10] T. Shirai, K. Shibutani, and T. Akishita, "The 128-Bit Blockcipher CLEFIA (Extended Abstract)," Luxembourg City, Luxembourg: Springer, Berlin, Heidelberg, 2007, pp. 181–195. doi: 10.1007/978-3-540-74619-5_12.

[11] R. Beaulieu, S. Treatman-clark, B. Weeks, and F. Meade, "The SIMON and SPECK lightweight block ciphers," *IACR Cryptol. ePrint Arch.*, 2013, doi: 10.1145/2744769.2747946.

[12] H. Noura and A. Chehab, "Lightweight Stream Cipher Scheme for Resource-Constrained IoT Devices," 2019.

[13] M. Hell, T. Johansson, and W. Meier, "Grain - A Stream Cipher for Constrained Environments," *Int. J. Wirel. Mob. Comput.*, vol. 2, no. 1, pp. 86–93, 2006.

[14] C. De Canni, "Trivium : A Stream Cipher Construction Inspired by Block Cipher Design Principles," Springer, Berlin, Heidelberg, 2006, pp. 171–172.

[15] Steve Babbage and M. Dodd, "The stream cipher MICKEY 2 . 0 Steve Babbage Matthew Dodd," 2006, *ECRYPT Stream Cipher Project (eSTREAM)*.

[16] C. Dobraunig, M. Eichlseder, F. Mendel, and M. Schläffer, "A SCON v1 . 2 : Lightweight Authenticated Encryption and Hashing," *J. Cryptol.*, vol. 34, no. 3, pp. 1–42, 2021, doi: 10.1007/s00145-021-09398-9.

[17] J. Guo, T. Peyrin, and A. Poschmann, "The PHOTON Family of Lightweight Hash Functions," Santa Barbara, CA, USA: Springer, Berlin, Heidelberg, 2011.

[18] A. Bogdanov and M. Kneˇ, "spongent : A Lightweight Hash Function," Nara, Japan: Springer, Berlin, Heidelberg, 2011, pp. 312–325. doi: 10.1007/978-3-642-23951-9_3.

[19] McKay, B. Larry, T. Meltem Sönmez, and M. Nicky, "Report on Lightweight Cryptography," *Natl. Inst. Stand. Technol.*, vol. NISTIR 811, p. 27, 2017, doi: 10.6028/NIST.IR.8114.

[20] K. A. Mckay, L. Bassham, and L. Bassham, *Status Report on the First Round of the NIST Lightweight Cryptography Standardization Process Status Report on the First Round of the NIST Lightweight Cryptography Standardization Process*, (2019).

[21] B. Liu, L. Li, R. Wu, M. Xie, and Q. P. Li, "Loong : A Family of Involutional Lightweight Block Cipher Based on SPN Structure," *IEEE Access*, vol. 7, pp. 136023–136035, 2019, doi: 10.1109/ACCESS.2019.2940330.

[22] L. Cryptography and S. Process, *DryGASCON*. SCITEPRESS — Science and Technology Publications, 2019.

[23] S. Banik, Z. Bao, T. Isobe, and H. Kubo, "WARP : Revisiting GFN for Lightweight 128-bit Block Cipher," *IACR Trans. Symmetric Cryptol.*, pp. 1–35, 2020, doi: 10.46586/tosc.v2020.i2.1-23.

[24] D. Hutchison and J. C. Mitchell, *Information Security and Cryptology – ICISC 2011*.

[25] J. Daemen, P. Maat, C. Massolino, and Y. Rotella, "The Subterranean 2 . 0 cipher suite," *IACR Trans. Symmetric Cryptol.*, 2020, doi: 10.1007/s10623-021-00892-6.

[26] B. W. Aboshosha, R. A. Ramadan, and A. D. Dwivedi, "SLIM : A Lightweight Block Cipher for Internet of Health Things .," *IEEE Access*, pp. 1–11, 2020.

[27] Y. Guo, L. Li, and B. Liu, "Shadow : A Lightweight Block Cipher for IoT," *IEEE Internet Things J.*, vol. 4662, no. XX, pp. 1–10, 2021, doi: 10.1109/JIOT.2021.3064203.

[28] S. Chen *et al.*, "SAND : An AND-RX Feistel Lightweight Block Cipher Supporting S-box-based Security Evaluations," *IACR Trans. Symmetric Cryptol.*, pp. 1–47, 2023, doi: 10.46586/tosc.v2023.i2.1-29.

[29] K. C. Gupta, S. K. Pandey, and S. Samanta, "FUTURE : A Lightweight Block Cipher Using An Optimal Diffusion Matrix," 2022.

[30] R. Avanzi *et al.*, "The QARMAv2 Family of Tweakable Block Ciphers," *IACR Trans. Symmetric Cryptol.*, 2023, doi: 10.46586/tosc.v2023.i3.1-33.

[31] W. Chen, L. Li, Y. Guo, and Y. Huang, "SAND-2 : An optimized implementation of lightweight block cipher," *Integration*, vol. 91, no. August 2022, pp. 23–34, 2023, doi: 10.1016/j.vlsi.2023.02.013.

[32] A. Gupta and T. Sasikala, "Lightweight Block Cipher for Security in Resource-Constrained Network," *Int. J. Electr. Comput. Eng.*, pp. 507–515, 2025, doi: 10.32985/ijeces.16.7.2.