computers

MDPI

Review

Lightweight Encryption Algorithms for IoT

Cláudio Silva [1], Nelson Tenório [2] and Jorge Bernardino [1,*]

[1] Coimbra Institute of Engineering, Polytechnic University of Coimbra, Rua da Misericórdia, Lagar dos Cortiços, São Martinho do Bispo, 3045-093 Coimbra, Portugal

[2] Cesumar Institute of Science, Technology and Innovation, Cesumar University, Avenida Guedner, 1610, B.7, Jardim Aclimação, Maringá 87013-100, Paraná, Brazil

* Correspondence: jorge@isec.pt

Abstract

The exponential growth of the Internet of Things (IoT) has increased the demand for robust security solutions that are tailored to devices with limited resources. This paper presents a systematic review of recent literature on lightweight encryption algorithms designed to meet this challenge. Through an analysis of 22 distinct ciphers, the study identifies the main algorithms proposed and catalogues the key metrics used for their evaluation. The most common performance criteria are execution speed, memory usage, and energy consumption, while security is predominantly assessed using techniques such as differential and linear cryptanalysis, alongside statistical tests such as the avalanche effect. However, the most critical finding is the profound lack of standardized frameworks for both performance benchmarking and security validation. This methodological fragmentation severely hinders objective, cross-study comparisons, making evidence-based algorithm selection a significant challenge and impeding the development of verifiably secure IoT systems.

Keywords: Internet of Things; IoT cryptography; lightweight encryption algorithms

1. Introduction

The Internet of Things (IoT) has firmly established itself as one of today's most prominent and important technological areas. It is characterized by the massive proliferation of interconnected devices, with an estimated number of 20.1 billion devices by the end of 2025 and 39.6 billion by the end of 2033 [1]. Due to its multiple benefits and decreasing cost, IoT technology has found a place in critical domains such as transportation, logistics, medicine, and Industry 4.0. It also contributes to automation and everyday comfort in domestic environments [2,3].

The concept of an IoT device is broad and sometimes hard to define. It ranges from personal items, such as smartwatches, smartphones, and computers, to domestic equipment, such as light bulbs and household appliances like refrigerators, washing machines, and air fryers. IoT devices also include various types of sensors (temperature, humidity, movement, and air quality), surveillance cameras, drones, vehicles and even urban infrastructures such as traffic lights and intelligent waste management systems [4].

The widespread adoption of IoT devices has prompted numerous problems. One of the biggest and most immediate challenges is physical security [5]. Many IoT devices are designed to be installed in places that are easily accessible and without constant surveillance, such as in agricultural fields, streetlights or even inside homes. Physical access to a device enables attackers to extract cryptographic keys and other sensitive data directly from memory, make hardware or firmware changes, install malicious software, and conduct

check for updates

Academic Editor: Paolo Bellavista

Received: 20 October 2025

Revised: 19 November 2025

Accepted: 20 November 2025

Published: 21 November 2025

Citation: Silva, C.; Tenório, N.; Bernardino, J. Lightweight Encryption Algorithms for IoT. *Computers* **2025**, *14*, 505. https://doi.org/10.3390/computers14120505

side-channel attacks, which analyze power consumption or electromagnetic radiation to infer cryptographic secrets [6]. Another fundamental challenge lies in the limited computing resources [2]. Most IoT devices operate with severe restrictions on battery life, processing power, memory and storage capacity. Traditional cryptographic algorithms, such as AES, can be too demanding in terms of computing and energy consumption, which creates a difficult choice between robust security and the operational viability of the device. Furthermore, the eventual arrival of quantum computers capable of breaking many of the algorithms currently in use represents an existential threat to IoT security. The transition to Post-Quantum Cryptography (PQC), which is resistant to quantum attacks, imposes new performance and implementation challenges [7]. Thus, robust security remains a complex and actively researched challenge [8].

This review focuses on a key issue in IoT security: how to apply lightweight cryptography to protect data sent by devices with limited resources. Given these limitations, selecting efficient algorithms is crucial. This paper therefore provides a comprehensive overview of lightweight cryptography and its applicability to such environments. The main objective is to analyze and compare different algorithms, to identify those that offer the optimal balance of robust security and processing, memory, and energy efficiency for such devices.

The rest of this paper is structured as follows: Section 2 details the research methodology, and Section 3 presents the ciphers referenced in the selected papers, providing a high-level description of each cipher. Section 4 focuses on the quality analysis of the selected papers and formulation of answers to the research questions. Section 5 discusses the limitations of this study, and Section 6 suggests ways to advance the field in the future. Finally, Section 7 presents the conclusions of this research.

2. Research Methodology

This section provides a detailed description of the methodology adopted for this review. The study follows Kitchenham's six-step methodology [9], which is widely disseminated in scientific articles. The steps are as follows: defining the research questions, selecting the data sources, creating the search string, defining the precondition criteria, applying the inclusion and exclusion criteria and, finally, extracting the data. Additionally, a new step has been added: semantic analysis.

This review was conducted in compliance with the PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) guidelines, ensuring transparency and reproducibility throughout the process.

2.1. Research Questions

The questions were formulated to structure the review and to ensure that the selected studies directly contribute to the research objective. The guiding RQs for this systematic review are summarized as follows:

RQ1: What are the main lightweight encryption algorithms proposed in the scientific literature for IoT environments?

RQ2: What are the key performance evaluation criteria commonly used to assess and compare lightweight encryption algorithms?

RQ3: What are the main security evaluation techniques used to analyze the robustness of lightweight encryption algorithms?

2.2. Data Sources

The articles selected for this review were sourced from four major academic and scientific databases:

- ACM Digital Library
- IEEE Digital Library
- Science Direct
- Springer Link

2.3. Search String

The search expression used was: ((“IoT” OR “Internet of Things”) AND “encryption algorithms” AND “lightweight cryptography”); -Review, -Survey, -Blockchain were also added to narrow the search, since it was intended to find algorithms adapted for a general utilization.

2.4. Preconditions

Preconditions are essential for filtering out articles published more than five years ago and selecting only the most recent ones. Sometimes, more than half of the articles found are eliminated. The following preconditions are used:

- English is the required writing language for all studies.
- Studies must be published after 2020.

2.5. Semantic Pre-Analysis

To efficiently manage the large volume of literature was employed the tool Orange Data Mining (version 3.39), an open-source data mining and machine learning platform, widely recognized for its intuitive visual programming interface based on workflow components [10]. In this study, Orange was used to perform a semantic clustering of the retrieved articles to aid in identifying relevant studies. This was done in two steps: (i) vectorization of textual content, which include titles, abstracts, and keywords; and (ii) clustering algorithms to identify thematically related clusters of publications. This semantic analysis improved the selection of relevant articles for in-depth review, filtering out studies that did not meaningfully contribute to answering the RQs.

We adopted a two-step procedure to identify publication clusters and determine which articles should be considered for preliminary reading. As shown in Figure 1, the first step employs Orange data mining widgets to classify the clusters and extract the relevant articles for review. Specifically, the ‘Corpus Reading’ widget ingests a “.xlsx” file containing entries organized by keyword, title, and abstract. The ‘Select Columns’ widget then designates the keywords, titles, and abstracts as variables for subsequent semantic analysis. Following this, the ‘Document Embedding’ widget is used for text vectorization, and the ‘Distance Calculation’ widget computes the semantic distance between text entries using the cosine similarity method. The ‘Hierarchical Clustering’ widget then groups thematically similar documents based on their vector representations. Finally, a word cloud visualization highlights the most frequent terms within each cluster. In the second step (Figure 1), articles were filtered by identifying the two most frequent terms in each cluster: ‘secure’ and ‘lightweight’ in Cluster 1 and ‘secure’ and ‘IoT’ in Cluster 2. These terms were subsequently applied in the ‘Select Row’ widget (see ‘Terms 1–3′ in Figure 1) to refine the document selection further. Ultimately, this process yielded a corpus of 212 publications, which were included in the records screened for this study.

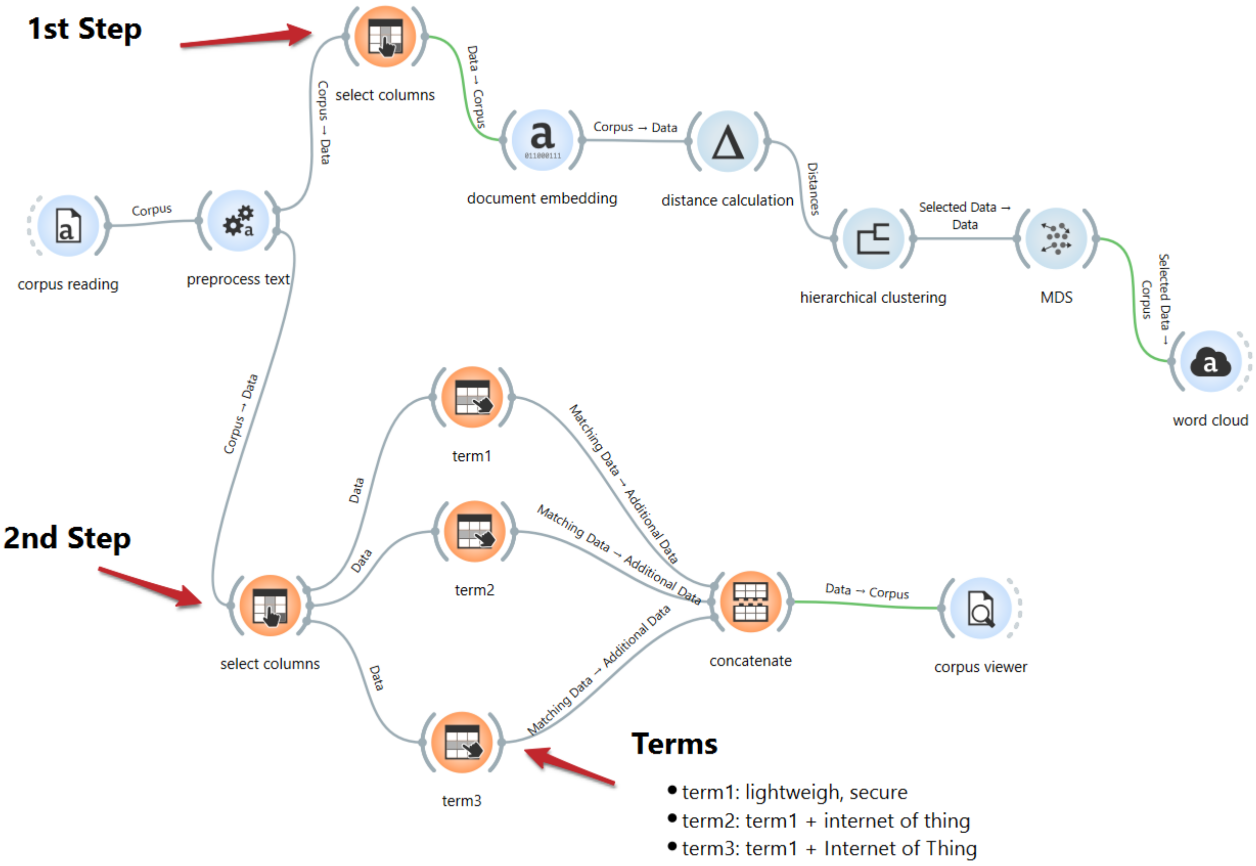

Figure 1. Orange widget to determine records screened.

2.6. Inclusion and Exclusion Criteria

The inclusion and exclusion criteria were used to filter articles based on their content. The following criteria were used:

- The algorithm has an overly narrow focus.
- The article did not answer any of the research questions.
- The article was not available (due to institutional limitations, for example).

2.7. Data Extraction

The article selection process began with the identification of 457 records from databases, along with two additional records, one identified through citation searching and another through manual search, while refining the research questions. After removing 247 ineligible records identified by automation tools during the Semantic Pre-Analysis, 212 records remained for screening. Following the title and abstract analysis, 156 records were excluded due to their low relevance, leaving 56 records to be retrieved. Six of these records were unavailable due to institutional limitations, resulting in 50 records assessed for eligibility. Based on the inclusion and exclusion criteria, 28 records were excluded due to narrow focus or lack of relevance to the research questions. Twenty-two studies were ultimately included in the final review. Figure 2 illustrates the process used for this review.

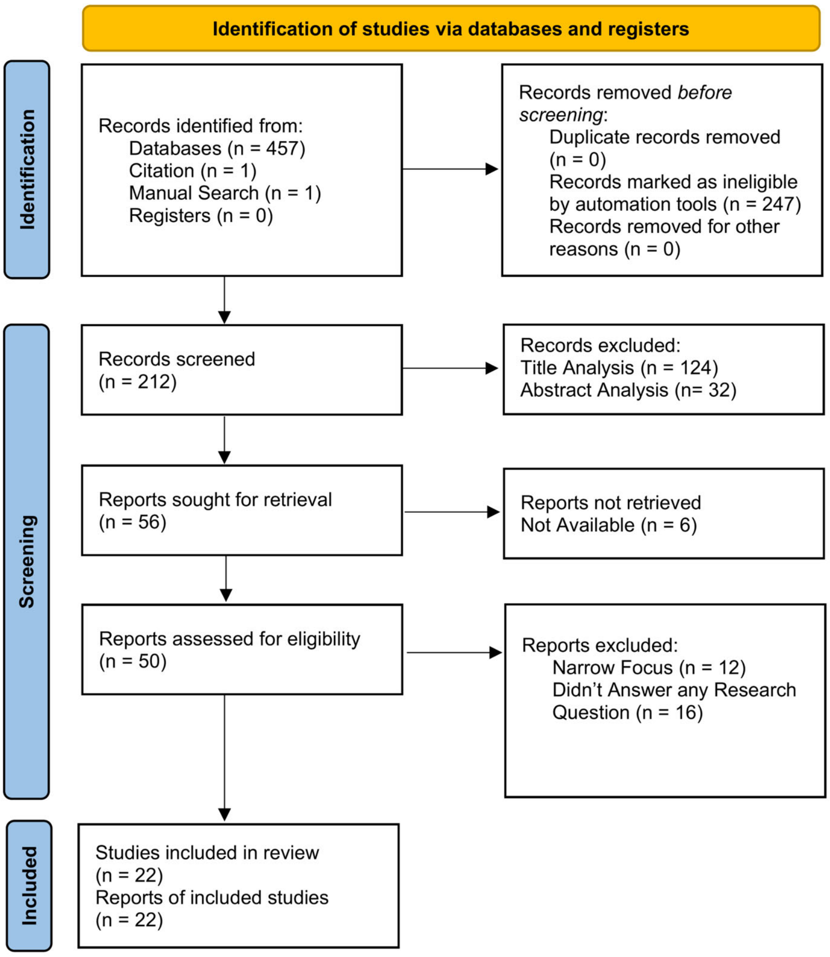
Identification of studies via databases and registers
Identification
Records identified from:
Databases (n = 457)
Citation (n = 1)
Manual Search (n = 1)
Registers (n = 0)
Records removed before screening:
Duplicate records removed (n = 0)
Records marked as ineligible by automation tools (n = 247)
Records removed for other reasons (n = 0)
Screening
Records screened (n = 212)
Records excluded:
Title Analysis (n = 124)
Abstract Analysis (n= 32)
Reports sought for retrieval (n = 56)
Reports not retrieved
Not Available (n = 6)
Reports assessed for eligibility (n = 50)
Reports excluded:
Narrow Focus (n = 12)
Didn't Answer any Research Question (n = 16)
Included
Studies included in review (n = 22)
Reports of included studies (n = 22)

Figure 2. Literature Review Process.

3. Ciphers

This section provides a concise overview of the lightweight block ciphers proposed in the selected literature. The primary aim is to describe the main architectural contributions with high-level descriptions provided in Table 1.

A Substitution-Permutation Network (SPN) is a structure in which S-boxes perform substitutions to create confusion and hide the relationship with the key in each round. The P-boxes then permute the data to create diffusion and spread the influence of each bit. A Feistel Network (FN) is a structure that divides data into two blocks. In each round, a function is applied to one block, and the result modifies the other block, typically through an XOR operation. The blocks are then swapped, and this process is repeated for multiple rounds.

A Generalized Feistel Network (GFN) is similar to a Feistel Network, but instead of dividing the data into just two blocks, it divides it into multiple blocks (usually a power of two) and applies the function to each pair of blocks. An Addition-Rotation-XOR (ARX) structure is based on three simple and efficient operations applied to the data: addition, bit rotation, and the XOR operation.

Some ciphers employ less-common designs. These include the Lai–Massey scheme, which takes inspiration from FN and modifies both halves of the block each round, and the Type-1 EGFN, which is a variant of the GFN. These structures are uncommon and not widely documented.

Table 1. High-level Description of the Ciphers.

Ciphers	Description
SPNRX [11]	The SPNRX cipher distinguishes itself with a hybrid design, merging a modified ARX structure with a SPN to enhance diffusion speed while minimizing computational load. SPNRX operates on 64-bit blocks with a 128-bit key over 16 rounds and features a novel key schedule based on matrix transformation and P-box permutation, aiming for both security and hardware efficiency.
QLW [12]	The QLW cipher is distinguished by its unique hybrid design, which merges a type-III GFS with Lai–Massey principles to achieve rapid diffusion and high efficiency. QLW operates on 64-bit blocks with a 128-bit key over 19 rounds. Key innovations include a 4-bit S-box optimized via a genetic algorithm and a dynamic round constant derived from the key schedule.
GFSPX [13]	The GFSPX cipher introduces a hybrid architecture that combines a 4-branch GFS with elements from a SPN. This design aims to overcome the slow diffusion of traditional Feistel ciphers by using two different round functions: one with hardware-efficient ARX like operations and another with a strong SPN structure using PRESENT S-boxes. The algorithm operates on 64-bit data blocks with a 128-bit key over 20 rounds, providing robust security for resource-constrained IoT applications.
MBRISI [14]	The MBRISI cipher is a lightweight, Feistel-based block cipher designed for area efficiency by uniquely combining ARX operations inspired by the BRIGHT cipher family and a modified SIMON structure. It operates on 32-bit plaintext blocks with a 64-bit key over 10 rounds. Its most distinct feature is a novel key generation algorithm that fuses techniques from the Secure IoT (SIT) algorithm, a Modified Fibonacci sequence, and a Scrambling algorithm.
DRcipher [15]	The DRcipher is a lightweight cipher for IoT built on a 4-branch GFN. Its most distinct quality is a pseudo-random dynamic number of encryption rounds, which is determined by the primary key to balance security and efficiency. The cipher processes 64-bit blocks with either a 96-bit or 128-bit key. Its round function integrates two custom 4×4 S-boxes and a novel negative feedback mechanism.
D-PRESENT [16]	D-PRESENT is an enhancement of the ISO-standardized lightweight cipher PRESENT, specifically designed to address vulnerabilities associated with its static S-box. It retains the original 31-round SPN structure, operating on 64-bit blocks with either an 80-bit or 124-bit key. The central innovation is a dynamic, key-dependent S-box mechanism where one of 16 available S-boxes is selected by the key and then shifted in each round to increase resistance to attacks.
Verma et al. [17]	This optimized version of the SIMON lightweight block cipher modifies the SIMON64/128 variant to enhance performance for IoT applications. The design streamlines the round function by reducing a circular bit shift and consolidating two bitwise operations into a single AND operation, thus lowering the computational overhead.
ALLPC [18]	The ALLPC cipher is based on a novel Extended Type-1 GFN. This 8-branch GFN enhances the slow diffusion of traditional Feistel designs by incorporating an additional linear layer. The cipher operates on 64-bit blocks with a 128-bit key over 25 rounds.
Hafsa et al. [19]	This lightweight symmetric block cipher is designed for high-speed, real-time applications and features a complex single-round architecture. The algorithm operates on 128-bit blocks, with its round function incorporating a novel "Mix-data" transformation, a random permutation, and 16 parallel S-boxes. Key generation is distinct, using a PRNG based on a combination of the Lorenz and Lui chaotic systems to produce a key stream from a 128-bit secret key.

Table 1. *Cont.*

Ciphers	Description
Chatterjee et al. [20]	This modified PRESENT cipher alters the original PRESENT-80 algorithm to improve its security. The design introduces two primary changes: a new key update schedule incorporating the delta function from the TEA cipher and an additional processing layer inserted between the S-box and permutation layer in each round. The authors claim these modifications are significant enough to reduce the required rounds from 31 to 25 while improving security.
ECLBC [21]	The ECLBC cipher is designed to provide both data security and data reliability, its core innovation is an integrated error detection and correction mechanism that uses a linear block code to expand the internal state, allowing the receiver to correct errors from noisy channels. The cipher uses a 40-round SPN architecture with a nonlinear layer based on the ARX principles of the SIMON cipher, operating on 32-bit or 64-bit blocks.
NLW-AES [22]	The NLW-AES is a direct modification of the standard AES algorithm designed to improve performance on IoT devices, the sole architectural change is a drastic reduction in the number of encryption rounds, from the standard 10 down to just 3.
ULC [23]	The Ultra-Lightweight Method (ULM) is a design methodology that synthesizes Bitslice, Wide-Trail Strategy, and compact design principles for optimizing ciphers. The paper presents a concrete instance, the Ultra-Lightweight Cryptosystem (ULC), an SPN-based cipher that processes 64-bit blocks with an 80-bit key over 15 rounds.
Ghorashi et al. [24]	This optimization of the Klein lightweight block cipher improves efficiency and addresses known security vulnerabilities; the work identifies the MixNibble algorithm as a performance bottleneck in the Klein-64 design. The proposed modification replaces this component with a new structure consisting of three sequential S-box layers interleaved with key-mixing XOR operations.
PMSEA [25]	The PMSEA is a lightweight algorithm designed around the concept of a multi-purpose key to reduce power consumption. In this asymmetric scheme, the private key is not a shared secret but is dynamically generated from the plaintext itself, based on the data's statistical properties. This data-dependent key is used in a three-round encryption process.
Alluhaidan et al. [26]	This lightweight symmetric block cipher utilizes a 5-round modified Feistel architecture that integrates both SPN and genetic algorithm principles. The algorithm operates on 64-bit blocks with a 64-bit key, featuring a key schedule that uses a "P-function" and "non-linear bit shuffling".
PREXTEA [27] TRIXTEA [27]	PREXTEA and TRIXTEA are proposed to address the weak key scheduling of the XTEA algorithm. Both variants retain the efficient 64-bit Feistel round function of XTEA but replace its key schedule with a more robust mechanism: PREXTEA adopts the key schedule from the PRESENT block cipher, while TRIXTEA uses the one from the TRIVIUM stream cipher.
Vimalkumar et al. [28]	This modified Lightweight 128-bit AES algorithm is intended to reduce the resource consumption of standard AES for IoT devices. This variant introduces several major structural changes: the number of rounds is reduced from 10 to 6, the SubBytes operation is performed only once, and a new S-box and "Shift Row Column" operation are used, with a ZigZag pre-processing step added to compensate for the loss of diffusion.
Vaz et al. [29]	This cipher is lightweight version of AES that optimizes its most resource-intensive stages. The design retains the standard 10-round, 128-bit block structure but modifies two core components: the SubBytes stage is re-engineered to use a single, 16-byte involutive S-box to save memory, and the MixColumns stage is implemented with computationally lighter operations.
Chia Ni et al. [30]	This lightweight block cipher was specifically tailored for the ESP32 microcontroller, utilizing a Feistel Network to align with the platform's 32-bit architecture. The algorithm operates on a 64-bit block with a 64-bit key and is built with custom components, including unique S-boxes, P-boxes, and a full-state row permutation for diffusion.

Table 1. *Cont.*

Ciphers	Description
Bhagya S et al. [31]	This lightweight block cipher was specifically proposed to address the well-known key scheduling vulnerabilities of the original TEA cipher. While retaining TEA's 64-bit block size, 128-bit key, and Feistel structure, the core innovation is a dynamic round key scheduling mechanism that makes the round keys dependent on the plaintext data being encrypted.
3D RECTANGLE [32]	This cipher is version of the RECTANGLE block cipher designed to improve its cryptographic security. While retaining RECTANGLE's 64-bit block size, 128-bit key, and 25-round SPN structure, the core innovation is a 3DBitRotation function, this function enhances diffusion by conceptually mapping the 64-bit state into a 4 × 4 × 4 cube and rotating each slice differently.

4. Qualitative Analysis of Selected Papers

This section provides an in-depth qualitative analysis of the 22 studies selected for this review. The analysis is structured to systematically answer the three research questions established in the methodology. First, we address RQ1 by identifying and describing the main lightweight algorithms proposed in the literature. Next, we analyze the performance (RQ2) and security (RQ3) evaluation criteria applied in these studies. Finally, we discuss the methodological gaps and trends observed across the field.

4.1. Algorithm Architectures and Implementation Trends (RQ1)

In addressing RQ1 (What are the main lightweight encryption algorithms proposed in the scientific literature for IoT environments?), this systematic review cataloged 22 distinct lightweight encryption algorithms proposed for IoT security. As presented in Table 2, these ciphers display a wide variety of design approaches, but they collectively highlight a significant trend regarding the compromise between cryptographic strength and implementation feasibility.

Most of the algorithms use 128-bit cryptographic keys, considered the minimum standard recommended by international standards, such as NIST SP 800-57, is 112. However, in the context of IoT, using keys of this size can be impractical for devices with extremely limited resources. In these situations, smaller keys are needed to enable the algorithm to run, even though this choice significantly reduces the level of security. It is therefore a compromise between cryptographic robustness and the feasibility of implementation, which must be carefully assessed in light of the risk associated with the use case.

In addition, several of the algorithms analyzed are only available in the form of hardware implementations, namely through logic gates or Field-Programmable Gate Arrays (FPGAs). These solutions have considerable performance and energy efficiency advantages. However, they can limit the IoT ecosystem because they require specialized hardware that is not easily compatible with widely used platforms, such as Arduino, Raspberry Pi, and ESP32. For these types of devices, software-implemented solutions are generally more practical, accessible, and portable.

Many authors have chosen to prototype their algorithms in Python due to its simplicity, high readability, and ecosystem of libraries. However, for real-world applications, low-level languages such as C and C++ are more suitable because they allow for optimization in terms of memory access and efficient use of hardware resources, resulting in significant performance gains.

Table 2. Comparison of the Algorithms.

Ciphers	Key Size (bit)	Block Size (bit)	Number of Rounds	Structure Type	Reference Ciphers	Implementation	Coding Language
SPNRX [11]	128	64	16	ARX SPN	-	HW [1] SW [2]	Python
QLW [12]	128	64	19	GFS Lai-Massey	-	HW [1]	-
GFSPX [13]	128	64	20	ARX GFS SPN	-	HW [1] SW [2]	C
MBRISI [14]	64	32	10	ARX FR	BRIGHT SIMON	HW [1] SW [2]	MATLAB(R2018b)
DRcipher [15]	96 128	64	Dynamic	GFN	-	HW [1] SW [2]	-
D-PRESENT [16]	80 124	64	16	SPN	PRESENST	SW [2]	Python 3.7
Verma et al. [17]	128	64	44	AXR	SIMON	SW [2]	Python
ALLPC [18]	128	64	25	Type-1 EGFNs	-	HW [1] SW [2]	-
Hafsa et al. [19]	128	128	1	SPN	-	HW [1] SW [2]	C/C++ MATLAB R2018b
Chatterjee et al. [20]	80 128	64	25	SPN	PRESENT TEA	SW [2]	C
ECLBC [21]	64 128	32 64	40	ARX SPN	SIMON	HW [1]	-
NLW-AES [22]	128	128	3	SPN	AES	SW [2]	-
ULC [23]	80	64	15	SPN	RECTANGLE	SW [2]	C
Ghorashi et al. [24]	64 80 96	64	12 16 20	SPN	Klein	SW [1]	Python 3
PMSEA [25]	-	-	3	-	-	SW [2]	Python 3.9
Alluhaidan et al. [26]	64	64	5	FN SPN	-	SW [2]	C
PREXTEA [27]	128	64	32	FN	PRESENT XTEA	SW [2]	Python
TRIXTEA [27]	128	64	32	FN	TRIVIUM XTEA	SW [2]	Python
Vimalkumar et al. [28]	128	128	6	SPN	AES	HW [1]	-
Vaz et al. [29]	128	128	10	SPN	AES	SW [2]	C/C++
Chia Ni et al. [30]	64	64	4	FN	AES DES	SW [2]	-
Bhagya S et al. [31]	128	64	64	FN	TEA	SW [1]	-
3D RECTANGLE [32]	80 128	64	25	SPN	RECTANGLE	SW [1]	C++

[1] Hardware Implementation. [2] Software Implementation.

4.2. Analysis of Performance Metrics (RQ2)

Following the architectural review, the analysis now addresses RQ2 (What are the key performance evaluation criteria commonly used to assess and compare lightweight encryption algorithms), which focuses on identifying the key performance criteria used to evaluate these ciphers. Table 3 summarizes the performance metrics used across the selected studies to evaluate lightweight encryption algorithms in IoT contexts. The analysis reveals significant heterogeneity in how performance is measured and reported, which complicates direct comparisons and weakens the generalizability of findings. Despite the diversity of metrics, three main categories—speed, memory usage, and energy consumption—emerge as dominant, aligning partially with the research question RQ2. However, several gaps remain with respect to methodological rigor and standardization.

Table 3. Performance Metrics.

Ciphers	Speed [1]	RAM	ROM	Energy	CPU Usage	CPU Time	Clock Cycles	GEs	NIST IR 8114	ISO/IEC 29192-1
SPNRX [11]	✓	✓						✓		
QLW [12]				✓				✓		
GFSPX [13]	✓						✓	✓		
MBRISI [14]				✓						
[15]	✓							✓		
D-PRESENT [16]	✓									
Verma et al. [17]	✓									
ALLPC [18]	✓	✓		✓				✓		
Hafsa et al. [19]	✓			✓						
Chatterjee et al. [20]										
ECLBC [21]				✓				✓		
NLW-AES [22]	✓	✓		✓	✓					
ULC [23]	✓	✓	✓				✓			
Ghorashi et al. [24]	✓	✓			✓					
PMSEA [25]	✓		✓		✓	✓	✓			
Alluhaidan et al. [26]	✓	✓	✓	✓			✓			
PREXTEA [27]	✓	✓	✓	✓						
TRIXTEA [27]	✓	✓	✓	✓						
Vimalkumar et al. [28]	✓	✓		✓						
Vaz et al. [29]	✓	✓								
Chia Ni et al. [30]	✓									
Bhagya S et al. [31]	✓	✓								
3D RECTANGLE [32]	✓									

[1] Includes Throughput and Encryption/Decryption Time.

Execution Speed was the most widely adopted metric, reported in 18 studies, including [11,13,15–19,22–32]. These studies typically evaluated either encryption/decryption time or overall throughput. For example, SPNRX [11], GFSPX [13], and DRcipher [15] quantified speed using encryption latency or block throughput in hardware or software prototypes, while algorithms such as ULC [23] and 3D RECTANGLE [32] emphasized optimization of round functions to minimize processing time. Despite the broad use of speed as a metric, few papers standardized the hardware or conditions under which measurements were taken, limiting cross-study comparability. Furthermore, none of the works provided a normalized performance index across platforms with significant limitations for practical IoT deployment.

Memory usage, particularly RAM consumption, was addressed in 10 articles, including [11,18,22–24,26–29,31]. In most cases, memory requirements were inferred from the algorithm's structural design or implementation footprint. For instance, ALLPC [18] and NLW-AES [22] acknowledged the importance of minimizing memory in constrained environments but provided limited empirical data. Similarly, Vaz et al. [29] and Ghorashi et al. [24] discussed memory reduction strategies (e.g., S-box compression or substitution layer optimizations) but did not present consistent or detailed benchmarks, revealing a gap in methodological precision.

Energy and Power Consumption were reported in 9 studies, namely [12,14,18,19,21,22,26–28]. Papers such as QLW [12], Hafsa et al. [19], and ECLBC [21] evaluated energy efficiency in hardware simulations or referred to expected energy gains based on structural simplifications. However, empirical validation of power savings was often absent, and only a minority of works (e.g., MBRISI [14]) attempted to correlate energy efficiency with algorithmic features such as key scheduling or round design. The reliance on simulated rather than physical measurements limit the ecological validity of these results.

A smaller number of articles explored CPU-specific metrics, such as CPU usage [22,24,25] and clock cycles [13,23,25,26]. These metrics are particularly valuable for evaluating software-based encryption in microcontroller platforms (e.g., ESP32, ARM Cortex-M) yet were inconsistently applied. Chia Ni et al. [30], despite targeting a specific architecture, omitted low-level performance profiling, representing a missed opportunity for architecture-aware benchmarking.

Additionally, ROM usage [23,25–27] and CPU time [25] were addressed in only a few papers, often without detail on measurement methodology. These secondary metrics are crucial for understanding the total memory footprint and runtime behavior of an algorithm in real-world conditions but were generally underreported.

Therefore, while RQ2 identifies speed, memory, and energy as core performance criteria, the findings suggest that most studies lack a consistent performance evaluation framework. The absence of standard reference hardware, uniform test datasets, or benchmarking protocols severely hampers reproducibility. This aligns with the gap identified in RQ3, namely, that empirical comparisons remain fragmented and often incomparable due to inconsistent methodologies.

Moreover, none of the papers proposed a unified trade-off model between performance and security, which would be essential for developers making practical design decisions. Considering this, there is a relevant need for future works to develop standardized benchmarking environments and protocols, as well as multi-objective optimization models that formally consider performance–security trade-offs in lightweight cryptography.

This fragmentation is evident in Table 3, which shows that none of the 22 studies used the same metric. While 18 papers reported on Speed, only 10 measured RAM, 9 measured Energy, and just 6 reported on hardware GEs (Gate Equivalents). The inconsistent application of these metrics, which are recommended by international standards such as ISO/IEC 29192-1 [33] and the NIST IR 8114 report [34], is the core issue.

Therefore, the critical gap, as confirmed by Table 3, is not a failure to identify the correct metrics. Rather, the gap lies in the profound lack of a standardized methodology to ensure that all proposed ciphers are evaluated and compared using the same complete set of standard criteria. This corroborates the central finding of the review that this methodological fragmentation severely impedes objective, cross-study comparisons.

4.3. Analysis of Security Evaluation Techniques (RQ3)

The final analytical component addresses RQ3 (What are the main security evaluation techniques used to analyze the robustness of lightweight encryption algorithms?), which investigates the security evaluation techniques used to validate the robustness of the lightweight ciphers.

Table 4 presents the security evaluation techniques applied across the reviewed studies, reflecting a wide range of both theoretical and empirical approaches. These methods are essential for addressing RQ3. Although the variety of applied metrics is commendable, their inconsistent usage and limited depth reveal significant methodological gaps in the literature.

Table 4. Security Metrics.

Ciphers	Fault Attack Analysis	Side-Channel Analysis	Linear Cryptanalysis [1]	Differential Cryptanalysis	Algebraic Attack	Key Schedule Attack	NIST Statistical Test	Other Statistical and Diffusion Tests [2]
SPNRX [11]			✓	✓	✓			✓
QLW [12]			✓	✓	✓	✓		✓
GFSPX [13]			✓	✓	✓	✓		✓
MBRISI [14]							✓	✓
DRcipher [15]		✓	✓	✓	✓			✓
D-PRESENT [16]								✓
Verma et al. [17]				✓				✓
ALLPC [18]			✓	✓				
Hafsa et al. [19]				✓				✓
Chatterjee et al. [20]								✓
ECLBC [21]			✓	✓				
NLW-AES [22]							✓	✓
ULC [23]			✓	✓				
Ghorashi et al. [24]								
PMSEA [25]								
Alluhaidan et al. [26]						✓		✓
PREXTEA [27]			✓					✓
TRIXTEA [27]			✓					✓
Vimalkumar et al. [28]								✓
Vaz et al. [29]							✓	✓
Chia Ni et al. [30]								✓
Bhagya S et al. [31]				✓				✓
3D RECTANGLE [32]							✓	✓

[1] Includes Impossible Differential Cryptanalysis and Related-Key Differential Cryptanalysis. [2] Includes Avalanche Effect, Key Sensitivity, Histogram, Entropy, or Correlation analysis.

Firstly, it is important to distinguish between tests that measure statistical properties and those that evaluate true resilience against cryptanalytic attacks. Tests such as the avalanche effect, histogram analysis, and entropy measurement are valuable for evaluating statistical randomness and diffusion. However, they do not constitute proof of security against dedicated cryptanalytic attacks by themselves.

4.3.1. Cryptanalytic Resilience Tests

This category includes methods that assess the cipher's theoretical strength and resilience against known, direct cryptanalytic attacks.

Hence, a total of 10 studies employed 'differential cryptanalysis', including [11–13,15,17–19,21,23,31]. This method, which analyzes how small differences in input affect the output ciphertext, demonstrated structural resistance to differential attacks. For instance, SPNRX [11], QLW [12], and GFSPX [13] presented structured evaluations, while Verma et al. [17] and Hafsa et al. [19] presented the technique without in-depth analysis.

'Linear cryptanalysis,' in turn, was presented in 8 articles: [11–13,15,18,21,23,27], typically to assess the correlation between plain text, ciphertext, and key bits. However, only a few studies quantified the strength of linear approximations, and none compared their results against established thresholds or baselines.

Considering 'algebraic attacks' were presented in 4 studies ([11–13,15]), focusing on how the cipher structure may allow transformation into solvable algebraic equations. While the studies mention resistance, none conducted symbolic computations or used tools such as SAT solvers, limiting the depth of this analysis.

Other forms of theoretical cryptanalysis were sparsely such as 'integral attacks' [12,21]; key schedule attacks [12,13,26]; Meet-in-the-middle attacks [15], Zero-correlation attacks [18]; and Rotational-XOR and XSL attacks [11]. Those techniques were generally cited rather than empirically applied, and most papers lacked formal quantification of complexity or resistance margins.

4.3.2. Statistical and Diffusion Tests

In contrast to direct cryptanalysis, this group of tests evaluates the statistical properties and diffusion characteristics of the cipher's output in order to measure qualities such as randomness and data scrambling.

'Avalanche effect', the most frequently reported empirical measure in the literature. It appeared in 11 articles: [11–16,28–32]. This test evaluates whether small input changes produce widespread changes in the output. While studies such as MBRISI [14] and 3D RECTANGLE [32] reported favorable results, few presented round-by-round analyses or statistical thresholds, limiting comparability.

'Key sensitivity' was analyzed in only 4 studies: [11,14,26,32]. These works demonstrate that minor changes in the key produce significantly different cyphertexts but often rely on single-case experiments rather than systematic tests over multiple plain texts and keys.

Histogram analysis and correlation tests were both applied in 7 articles each: [11,14,17,19,20,22,26]. These visual and statistical tools are often used in image encryption validation but lack cryptographic rigor. The absence of confidence intervals or test reproducibility limits their reliability for security validation.

Entropy measurement was performed in 7 studies: [11,14,17,19,26,27,31]. While high entropy values are generally desirable, most articles report single entropy scores without detailing block size, sample count, or deviation from the ideal entropy (typically 8 bits), which reduces analytical strength.

NIST statistical tests, considered a gold standard for randomness validation, were only mentioned in 4 studies: [14,22,29,32]. Although these papers claim that their algorithms passed the NIST suite, they often do not specify which individual tests were performed or the pass rate thresholds, thus limiting reproducibility. Other less commonly used metrics include 'Non-homogeneity (Chi-Square) and N-gram analysis' [20]; 'Frequency and run tests' [27]; 'Bit error test' [32]. These metrics were introduced in isolated fashion and lacked comparative or baseline reporting.

While many articles cite a range of evaluation methods, the literature review reveals a lack of standardized and systematic approaches to security analysis, directly reflecting a critical gap in the response to RQ3. Thus, only a few articles combine formal cryptanalysis with empirical testing, and those that do often provide incomplete data or insufficient methodological detail. Moreover, there is no common framework for selecting or reporting metrics.

The advanced cryptanalysis scenarios such as known-plaintext or adaptive chosen-ciphertext attacks are absent, despite their practical relevance in IoT contexts. Most empirical results are presented in isolation, without statistical benchmarks, distributions, or variance indicators. This fragmentation significantly hinders comparability across studies and undermines confidence in the claimed robustness of many proposed algorithms.

4.4. Analysis of Security Validation Gaps

A deeper examination of the security evaluation techniques used in the reviewed studies reveals a significant and recurring limitation: although most papers present theoretical analyses or statistical randomness tests, they do not adequately address implementation-level attacks, particularly Side-Channel Attacks (SCAs) and Fault-Injection Attacks (FIAs). This gap is especially critical in IoT environments, where adversaries often have physical access to the device and can exploit its constrained hardware characteristics.

Side-channel attacks exploit unintentional information leakage from the physical implementation of a cryptographic algorithm, rather than mathematical weaknesses in the algorithm itself. Leakage can include power consumption, electromagnetic (EM) radiation, timing variations, cache behavior, or acoustic emissions. SCAs are widely recognized as one of the most practical and devastating classes of attacks on lightweight cryptography because lightweight ciphers typically operate on simple hardware with limited shielding or countermeasures.

The most common categories of SCAs include [35]:

- Power Analysis Attacks [6]: Such as Simple Power Analysis (SPA) and Differential Power Analysis (DPA), which infer key-dependent operations by measuring variations in power consumption during encryption.
- Electromagnetic (EM) Analysis [36]: Which detects EM emissions from the processor or microcontroller while performing cryptographic operations.
- Timing Attacks [37]: Which exploit execution time variations caused by data-dependent branching or table look-ups.
- Cache-Based Attacks [38]: More relevant for software implementations using lookup tables, where adversaries exploit memory access patterns.
- Template Attacks [39]: Highly precise attacks combining statistical modeling and observed leakage to recover secret keys.

Despite the practical relevance of these attacks in real-world IoT devices, Table 4 clearly shows that only one of the reviewed studies performed any form of side-channel evaluation, and even in that case, the analysis was limited to time measurements, due to the number of dynamic rounds and negative feedback impact, rather than comprehensive leakage assessment or resistance testing.

In addition to SCAs, fault-injection attacks constitute another major threat to lightweight cryptography. In these attacks, adversaries deliberately introduce faults, through voltage disturbances, clock manipulation, laser injection, or electromagnetic interference, to disrupt internal computations [40]. Differential Fault Analysis (DFA) [41] takes advantage of the resulting faulty outputs to recover keys or other sensitive information, often requiring only a small number of observations. However, as highlighted in Table 4, none of the reviewed studies performed experiments or simulations on fault-injection resistance, nor did they assess structural robustness against such attacks at a theoretical level.

This absence of practical, implementation-level validation significantly weakens the claimed security of many algorithms. Although several studies assert resistance to classical cryptanalysis (e.g., differential, linear, algebraic attacks), these assurances do not extend to scenarios where the algorithm is physically deployed on low-cost hardware, the very environment in which IoT devices operate. Lightweight ciphers implemented on microcontrollers, FPGAs, or ASICs may exhibit data-dependent leakage channels that remain undetectable in purely mathematical analyses.

Furthermore, none of the reviewed works implemented widely accepted countermeasures such as masking, hiding, threshold implementations, or noise-based defenses. Nor did they evaluate standard SCA leakage indicators (e.g., Test Vector Leakage Assessment [42]), which are commonly recommended in contemporary hardware security assessments. This lack of rigorous methodology prevents a realistic evaluation of whether the proposed ciphers could withstand an adversary with even modest physical access.

Overall, this systematic review identifies a profound disconnect between theoretical security evaluation and real-world adversarial models. While statistical diffusion tests and classical cryptanalysis remain important, they are insufficient for evaluating encryption algorithms intended for insecure, physically exposed environments. Strengthening the security guarantees of lightweight cryptography requires systematic and standardized integration of implementation-level evaluation frameworks, including power/EM leakage testing, DFA resistance experiments, and architecture-specific leakage modeling.

This gap reinforces the importance of the future-work directions proposed in Section 6, particularly the need for standardized evaluation frameworks and AI-assisted testing methodologies that can simulate and analyze modern attack vectors. Bridging this gap is essential to ensure the viability of lightweight ciphers in practical IoT deployments, where side-channel and fault-based threats represent some of the most realistic and effective lines of attack.

In summary, this qualitative analysis, grounded in the algorithms identified for RQ1, has successfully addressed RQ2 and RQ3 by identifying the principal performance and security metrics used to evaluate lightweight ciphers. The most critical finding, however, is not the metrics themselves but the profound lack of a standardized framework for applying them. The inconsistent methodologies in measuring performance and the fragmented application of security tests prevent objective, cross-study comparisons of the proposed algorithms. This makes it exceedingly difficult to make informed, evidence-based decisions when selecting an algorithm, as the claimed performance and security levels are not directly comparable. Therefore, while many solutions are proposed, the absence of a unified benchmarking protocol remains a significant barrier to advancing the field and ensuring robust security in practical IoT deployments.

5. Study Limitations

While this systematic review was conducted with a rigorous and transparent methodology, it is important to acknowledge certain limitations that may influence the scope and generalizability of the findings:

- Publication and Database Bias: The search was confined to four major academic databases: ACM Digital Library, IEEE Digital Library, Science Direct, and Springer Link. Consequently, relevant studies or algorithm proposals published in other databases, specialized conference proceedings, or as technical reports may have been omitted from this review.
- Semantic Pre-Analysis Filtering: The use of Orange Data Mining for semantic clustering was a crucial step for managing the large volume of initial results; however, like any automated tool, the process is dependent on the specific vectorization and clustering algorithms used. There is a possibility that this automated pre-filtering step could have inadvertently excluded some relevant articles before the manual screening phase.

- Subjectivity in Manual Screening: The final step in the article selection process involved a manual review of 210 articles based on their abstracts and the defined inclusion and exclusion criteria. This human screening, which narrowed the selection to the final 22 papers, is inherently subjective. While necessary, this process relies on the researcher's interpretation, and there is a possibility that this judgment could have led to the unintentional exclusion of some relevant studies.

6. Future Research Directions

In this section, we present potential future research directions for those who intend to advance in this field.

- Development of Lightweight Stream Ciphers: Our survey revealed a lack of cryptographic algorithms based on stream ciphers architecture. Stream ciphers are known for their simplicity and high performance, making them theoretically ideal for use with devices that have limited resources. Although a few notable designs exist, such as Lizard [43]. There is a significant research opportunity designing, implementing and analyzing novel stream ciphers that are specifically adapted for these environments. This would not only fill a gap in the current field but could also lead to new paradigms in high-performance, low-resource-usage encryption.
- Establishment of a Standardized Security Evaluation Framework: One of the major challenges identified during our research is the lack of standardization in the security analysis. Different studies apply various types of security tests, which makes it extremely difficult to make direct, objective comparisons between algorithms. Recent reviews of the NIST LWC finalists indicate that evaluation procedures, particularly for side-channel and fault resistance, vary considerably across studies, with several candidates lacking detailed or practical analyses [44].

  Future research should focus on developing a standardized security evaluation framework. This framework should define a mandatory minimum suite of cryptanalytic and implementation-level tests that all new and existing algorithms must undergo. A preliminary structure could include:

  - Core cryptanalytic evaluations: Differential, linear, and algebraic attacks.
  - Implementation of security tests: Standardized methodologies for side-channel (e.g., power, timing) and fault-injection resistance.
  - Reporting guidelines: A common format for documenting test conditions, metrics, and outcomes to ensure reproducibility.
- Creation of a Unified Performance Benchmarking Protocol: Similarly to the inconsistencies in security testing, we identified a lack of standardized protocols for performance evaluation. Experimental metrics are often measured on different hardware platforms and under varying conditions, which makes comparative analysis unreliable. Future research could focus on developing a preliminary unified benchmarking framework consisting of:

  - Reference hardware profiles: Standardized specifications for representative IoT devices (e.g., microcontrollers and low-power sensors) to ensure comparable test environments.
  - Standardized metrics: Core metrics including throughput, latency, energy consumption, and memory footprint. A good starting point for defining metrics could be ISO/IEC 29192, which identifies relevant performance variables such as speed, energy consumption, and memory usage.
  - Measurement methodology: Clear guidelines for timing, repetitions, and averaging to reduce variability across experiments.

    - ○ Reporting format: A structured template for recording results, including both raw data and normalized metrics.

  Existing initiatives, such as the NIST Lightweight Cryptography Performance Benchmarking Project, highlight the importance of this effort but also demonstrate the current absence of a universal methodology [45]. The development of an open-source benchmarking toolkit based on such a protocol would enable researchers to consistently evaluate the practical viability of algorithms for IoT applications.

- Investigation of Lightweight PQC for IoT: The advent of quantum computing poses a threat to most conventional public-key algorithms, underscoring the importance of studying post-quantum cryptography for ensuring the security of the IoT. However, many NIST-selected PQC schemes [46], such as CRYSTALS-Kyber [47] and CRYSTALS-Dilithium [48], are computationally demanding for constrained devices. Future research should therefore explore lightweight adaptations and hardware-optimized implementations of these standards, as well as alternative low-footprint candidates such as SABER [49], FrodoKEM [50], and SPHINCS+ [51], all of which have been tested in embedded contexts. Integrating these schemes efficiently into IoT systems while maintaining acceptable energy, memory, and latency profiles is a key challenge for the future.
- AI-Driven Cryptanalysis and Security Validation: Advances in artificial intelligence have introduced new opportunities and risks to the field of cryptography. Machine-learning models can exploit side-channel leakages and optimize differential fault analysis to recover secret keys from lightweight ciphers more efficiently [52]. Such AI-assisted cryptanalysis poses a particular threat to IoT devices with limited physical protection. Conversely, AI can also strengthen defences by enabling automated security-evaluation frameworks that simulate modern attacks and improve the reproducibility of testing. Future research should integrate these AI-based methods to enhance both offensive understanding and defensive resilience in lightweight cryptographic designs.
- Application of Frameworks for a Comparative Benchmark Study: A necessary next step following the creation of the standardized security and performance frameworks would be to conduct a comprehensive benchmark study. This new research should systematically apply these unified protocols to the algorithms cataloged in this review. Such a study would enable an objective cross-study comparison, which is currently hindered by methodological fragmentation. The results would provide a concrete, evidence-based ranking of existing proposals, enabling readers to make informed, practical decisions about which lightweight cipher offers the optimal balance of security and efficiency for their specific IoT applications.

7. Conclusions

Based on the analysis carried out, this study presents the main challenges to ensuring the security of IoT devices through lightweight cryptography. The massive proliferation of IoT devices, many with limited computing resources, requires encryption solutions that balance robustness and efficiency. The aim is not to compromise the usability of the devices while ensuring their security. This systematic review identified and analyzed the most recent lightweight algorithm proposals, answering fundamental questions about their characteristics, performance evaluation metrics and security.

In response to the research questions, it was concluded that the main algorithms are SPNRX, MBRISI, QLW and GFSPX, among others. It was also concluded that the main performance criteria to consider are execution speed, memory and energy consumption. In terms of security, it was found that methods such as linear cryptanalysis, differential cryptanalysis, algebraic attacks, avalanche effect, histogram analysis and the NIST statistical tests are the most commonly used for evaluating algorithms.

Although the analysis pointed out several challenges, such as the lack of standardization in the methodology for evaluating performance and security. This makes it difficult to objectively compare the various algorithms proposed.

Looking ahead, this research highlights the urgent need to improve security and performance criteria, as well as the importance of exploring new areas such as lightweight PQC. This is essential for ensuring the long-term security of IoT ecosystems against quantum computing threats. Progress in these areas is key to building a secure and reliable system to support continued expansion.

Author Contributions: Conceptualization, J.B.; methodology, J.B. and N.T.; software, C.S. and N.T.; validation, J.B. and N.T.; formal analysis, J.B. and N.T.; investigation, J.B., N.T. and C.S.; resources, J.B. and N.T.; data curation, C.S. and N.T.; writing—original draft preparation, C.S.; writing—review and editing, C.S. and N.T.; supervision, J.B. and N.T.; project administration, J.B.; funding acquisition, J.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Number of Internet of Things (IoT) Connections Worldwide from 2022 to 2023, with Forecasts from 2024 to 2033 (in Billions). Statista. Available online: https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/ (accessed on 1 May 2025).
2. Atzori, L.; Iera, A.; Morabito, G. The Internet of Things: A Survey. *Comput. Netw.* **2010**, *54*, 2787–2805. [CrossRef]
3. Chataut, R.; Phoummalayvane, A.; Akl, R. Unleashing the Power of IoT: A Comprehensive Review of IoT Applications and Future Prospects in Healthcare, Agriculture, Smart Homes, Smart Cities, and Industry 4.0. *Sensors* **2023**, *23*, 7194. [CrossRef]
4. Madakam, S.; Ramaswamy, R.; Tripathi, S. Internet of Things (IoT): A Literature Review. *J. Comput. Commun.* **2015**, *3*, 164–173. [CrossRef]
5. Sadkhan, S.B.; Salman, A.O. A Survey on Lightweight-Cryptography Status and Future Challenges. In Proceedings of the International Conference on Advances in Sustainable Engineering and Applications, ICASEA 2018-Proceedings, Kut, Iraq, 14–15 March 2018; pp. 105–108. [CrossRef]
6. Kocher, P.; Jaffe, J.; Jun, B. Differential Power Analysis. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Berlin/Heidelberg, Germany, 1999; Volume 1666, pp. 388–397. [CrossRef]
7. Bernstein, D.J.; Lange, T. Post-quantum cryptography. *Nature* **2017**, *549*, 188–194. [CrossRef]
8. Santos, R.J.; Bernardino, J.; Vieira, M. A Survey on Data Security in Data Warehousing: Issues, Challenges and Opportunities. In Proceedings of the EUROCON 2011-International Conference on Computer as a Tool-Joint with Conftele, Lisbon, Portugal, 27–29 April 2011. [CrossRef]
9. Kitchenham, B. Procedures for Performing Systematic Reviews. *Keele UK Keele Univ.* **2004**, *33*, 1–26.
10. Demšar, J.; Curk, T.; Erjavec, A.; Gorup, Č.; Hočevar, T.; Milutinovič, M.; Možina, M.; Polajnar, M.; Toplak, M.; Starič, A.; et al. Orange: Data Mining Toolbox in Python. *J. Mach. Learn. Res.* **2013**, *14*, 2349–2353.
11. Zhao, G.; Chen, H.; Wang, J. A Lightweight Block Encryption Algorithm for Narrowband Internet of Things. *Peer Peer Netw. Appl.* **2023**, *16*, 2775–2793. [CrossRef]
12. Yue, X.; Li, L.; Li, Q.; Xiang, J.; Hu, Z. QLW: A Lightweight Block Cipher with High Diffusion. *J. Supercomput.* **2025**, *81*, 224. [CrossRef]
13. Zhang, X.; Shao, C.; Li, T.; Yuan, Y.; Wang, C. GFSPX: An Efficient Lightweight Block Cipher for Resource-Constrained IoT Nodes. *J. Supercomput.* **2024**, *80*, 25256–25282. [CrossRef]
14. Poojary, A.; Kiran Kumar, V.G.; Nagesh, H.R. FPGA Implementation Novel Lightweight MBRISI Cipher. *J. Ambient. Intell. Humaniz. Comput.* **2023**, *14*, 11625–11637. [CrossRef]

15. Kuang, J.; Cao, X.; Li, S.; Li, L. DRcipher: A Pseudo-Random Dynamic Round Lightweight Block Cipher. *J. King Saud Univ. Comput. Inf. Sci.* **2024**, *36*, 101928. [CrossRef]
16. Labio, R.D.; Festijo, E.D. D-PRESENT: A Lightweight Block Cipher with Dynamic Key-Dependent Substitution Boxes. In Proceedings of the 2020 International Conference on Advanced Computer Science and Information Systems, ICACSIS, Depok, Indonesia, 17–18 October 2020; pp. 27–32. [CrossRef]
17. Verma, A.; Thokchom, S. An Optimized SIMON Lightweight Image Encryption Algorithm for Internet of Things: Balancing Performance and Security. In Proceedings of the 1st International Conference on Pioneering Developments in Computer Science and Digital Technologies, IC2SDT 2024-Proceedings, Delhi, India, 2–4 August 2024; pp. 592–597. [CrossRef]
18. Cheng, J.; Guo, S.; He, J. ALLPC: A Lightweight Block Cipher Based on Generalized Feistel Networks for IoT. In Proceedings of the IEEE International Performance, Computing, and Communications Conference 2021, Austin, TX, USA, 28–30 October 2021. [CrossRef]
19. Hafsa, A.; Gafsi, M.; Machhout, M. A Lightweight and Robust Block Cipher Algorithm for Real-Time Applications. *Signal Image Video Process.* **2024**, *18*, 1609–1624. [CrossRef]
20. Chatterjee, R.; Chakraborty, R. A Modified Lightweight PRESENT Cipher for IoT Security. In Proceedings of the 2020 International Conference on Computer Science, Engineering and Applications, ICCSEA, Online, 1–4 July 2020. [CrossRef]
21. Guo, Y.; Liu, W.; Chen, W.; Yan, Q.; Lu, Y. ECLBC: A Lightweight Block Cipher with Error Detection and Correction Mechanisms. *IEEE Internet Things J.* **2024**, *11*, 21727–21740. [CrossRef]
22. Qabajeh, L.; Tahboub, R.; Abujoodeh, M. A New Lightweight AES for IoT. In Proceedings of the 2023 International Conference on Information Technology: Cybersecurity Challenges for Sustainable Cities, ICIT 2023-Proceeding, Amman, Jordan, 9–10 August 2023; pp. 397–404. [CrossRef]
23. Sliman, L.; Omrani, T.; Tari, Z.; Samhat, A.E.; Rhouma, R. Towards an Ultra Lightweight Block Ciphers for Internet of Things. *J. Inf. Secur. Appl.* **2021**, *61*, 102897. [CrossRef]
24. Ghorashi, S.R.; Zia, T.; Jiang, Y. Optimisation of Lightweight Klein Encryption Algorithm with 3 S-Box. In Proceedings of the 2020 IEEE International Conference on Pervasive Computing and Communications Workshops, PerCom Workshops, Austin, TX, USA, 23–27 March 2020. [CrossRef]
25. Longwani, P.C.; Mendonca, I.; Aritsugi, M. A Lightweight Cryptographic Algorithm with a Multi-Purpose Encipher Key for IoT. In Proceedings of the 10th International Japan-Africa Conference on Electronics, Communications, and Computations, JAC-ECC, Alexandria, Egypt, 19–20 December 2022; pp. 15–20. [CrossRef]
26. Alluhaidan, A.S.D.; Prabu, P. End-to-End Encryption in Resource-Constrained IoT Device. *IEEE Access* **2023**, *11*, 70040–70051. [CrossRef]
27. Chaturvedi, S.P.; Mukherjee, R.; Kumar, S.; Yadav, A. Revolutionizing XTEA: Unveiling PREXTEA and TRIXTEA-Enhanced Efficiency and Security in Internet of Things. *IEEE Internet Things J.* **2025**, *12*, 3971–3979. [CrossRef]
28. Vimalkumar, J.; Babu, H.R.; Bhaskar, M. FPGA Implementation of Modified Lightweight 128-Bit AES Algorithm for IoT Applications. In Proceedings of the 2023 IEEE International Symposium on Smart Electronic Systems, iSES, Ahmedabad, India, 18–20 December 2023; pp. 306–309. [CrossRef]
29. Vaz, Y.S.; Mattos, J.C.B.; Soares, R.I. Improving an Ultra Lightweight AES for IoT Applications. In Proceedings of the 2023 IEEE World Forum on Internet of Things: The Blue Planet: A Marriage of Sea and Space, WF-IoT, Aveiro, Portugal, 12–27 October 2023. [CrossRef]
30. Ni, L.C.; Ali, S.; Rashid, R.A. Design of Cryptography Algorithm for Data Security of a IoT System. In Proceedings of the Conference Proceedings-IEEE International Conference on Advanced Telecommunication and Networking Technologies: Empowering Telecommunication Technologies for Sustainable Future, ATNT, Johor Bahru, Malaysia, 9–10 September 2024. [CrossRef]
31. Bhagya, S.; Jain, K.; Krishnan, P. Securing IoT Devices with Enhanced Tiny Encryption Algorithm. In Proceedings of the 3rd International Conference on Automation, Computing and Renewable Systems, ICACRS 2024-Proceedings, Pudukkottai, India, 4–6 December 2024; pp. 700–705. [CrossRef]
32. Zakaria, A.A.; Azni, A.H.; Ridzuan, F.; Zakaria, N.H.; Daud, M. Extended Rectangle Algorithm Using 3D Bit Rotation to Propose a New Lightweight Block Cipher for IoT. *IEEE Access* **2020**, *8*, 198646–198658. [CrossRef]
33. *ISO/IEC 29192-1:2012*; Information Technology—Security Techniques—Lightweight Cryptography—Part 1: General. International Organization for Standardization: Geneva, Switzerland, 2012.
34. Dang, Q.H. *Report on Lightweight Cryptography*; Internal Report (NISTIR) 8114; NIST Interagency: Gaithersburg, MD, US, 2016.
35. Spreitzer, R.; Moonsamy, V.; Korak, T.; Mangard, S. Systematic Classification of Side-Channel Attacks: A Case Study for Mobile Devices. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 465–488. [CrossRef]
36. Gandolfi, K.; Mourtel, C.; Olivier, F. Electromagnetic Analysis: Concrete Results. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Berlin/Heidelberg, Germany, 2001; Volume 2162, pp. 251–261. [CrossRef]

37. Kocher, P.C. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Berlin/Heidelberg, Germany, 1996; Volume 1109, pp. 104–113. [CrossRef]
38. Osvik, D.A.; Shamir, A.; Tromer, E. Cache Attacks and Countermeasures: The Case of AES. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; LNCS; Springer: Berlin/Heidelberg, Germany, 2006; Volume 3960, pp. 1–20. [CrossRef]
39. Chari, S.; Rao, J.R.; Rohatgi, P. Template Attacks. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Berlin/Heidelberg, Germany, 2003; Volume 2523, pp. 13–28. [CrossRef]
40. Bar-El, H.; Choukri, H.; Naccache, D.; Tunstall, M.; Whelan, C. The Sorcerer's Apprentice Guide to Fault Attacks. *Proc. IEEE* **2006**, *94*, 370–382. [CrossRef]
41. Biham, E.; Shamir, A. Differential Fault Analysis of Secret Key Cryptosystems. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Berlin/Heidelberg, Germany, 1997; Volume 1294, pp. 513–525. [CrossRef]
42. Schneider, T.; Moradi, A. Leakage Assessment Methodology a Clear Roadmap for Side-Channel Evaluations. *Lect. Notes Comput. Sci.* **2015**, *9293*, 495–513. [CrossRef]
43. Hamann, M.; Krause, M.; Meier, W. LIZARD—A Lightweight Stream Cipher for Power-Constrained Devices. *IACR Trans. Symmetric Cryptol.* **2017**, *2017*, 45–79. [CrossRef]
44. Madushan, H.; Salam, I.; Alawatugoda, J. A Review of the NIST Lightweight Cryptography Finalists and Their Fault Analyses. *Electronics* **2022**, *11*, 4199. [CrossRef]
45. Lightweight Cryptography | CSRC. Available online: https://csrc.nist.gov/projects/lightweight-cryptography/performance-benchmarking (accessed on 5 November 2025).
46. Alagic, G.; Apon, D.; Cooper, D.; Dang, Q.; Dang, T.; Kelsey, J.; Lichtinger, J.; Liu, Y.-K.; Miller, C.; Moody, D.; et al. *Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process*; NIST Interagency/Internal Report (NISTIR) 8413-upd1; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2022.
47. Jati, A.; Gupta, N.; Chattopadhyay, A.; Sanadhya, S.K. A Configurable Crystals-Kyber Hardware Implementation with Side-Channel Protection. *ACM Trans. Embed. Comput. Syst.* **2024**, *23*, 1–25. [CrossRef]
48. CRYSTALS-Dilithium: Resources. Available online: https://pq-crystals.org/dilithium/resources.shtml (accessed on 10 November 2025).
49. Wang, B.; Gu, X.; Yang, Y. Saber on ESP32. In Proceedings of the International Conference on Applied Cryptography and Network Security, Virtual, 19–22 October 2020.
50. Bos, J.W.; Bronchain, O.; Custers, F.; Renes, J.; Verbakel, D.; van Vredendaal, C. Enabling FrodoKEM on Embedded Devices. *Cryptol. Eprint Arch.* **2023**, *2023*, 3.
51. Magyari, A.; Chen, Y. Optimizing SPHINCS+ for Low-Power Devices. *Electronics* **2025**, *14*, 3460. [CrossRef]
52. Hameed, F.; Alkhzaimi, H. Deep Learning-Based Profiling Side-Channel Attacks in SPECK Cipher. *Sci. Rep.* **2025**, *15*, 26149. [CrossRef] [PubMed]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.