cryptography

MDPI

Article

Compact 8-Bit S-Boxes Based on Multiplication in a Galois Field $GF(2^4)$

Phuc-Phan Duong *, Tuan-Kiet Dang, Trong-Thuc Hoang and Cong-Kha Pham

Department of Computer and Network Engineering, University of Electro-Communications (UEC), 1-5-1 Chofugaoka, Chofu-shi, Tokyo 182-8585, Japan; tuankiet@vlsilab.ee.uec.ac.jp (T.-K.D.); hoangtt@uec.ac.jp (T.-T.H.); phamck@uec.ac.jp (C.-K.P.)

* Correspondence: duongphucphan@vlsilab.ee.uec.ac.jp

Abstract: Substitution boxes (S-Boxes) function as essential nonlinear elements in contemporary cryptographic systems, offering robust protection against cryptanalytic attacks. This study presents a novel technique for generating compact 8-bit S-Boxes based on multiplication in the Galois Field $GF(2^4)$. The goal of this method is to create S-Boxes with low hardware implementation cost while ensuring cryptographic properties. Experimental results indicate that the suggested S-Boxes achieve a nonlinearity value of 112, matching the AES S-Box. They also maintain other cryptographic properties, such as the Bit Independence Criterion (BIC), the Strict Avalanche Criterion (SAC), Differential Approximation Probability, and Linear Approximation Probability, within acceptable security thresholds. Notably, compared to existing studies, the proposed S-Box architecture demonstrates enhanced hardware efficiency, significantly reducing resource utilization in implementations. Specifically, the implementation cost of the S-Box consists of 31 XOR gates, 32 two-input AND gates, 6 two-input OR gates, and 2 MUX21s. Moreover, this work provides a thorough assessment of the S-Box, covering cryptographic properties, side channel attacks, and implementation aspects. Furthermore, the study estimates the quantum resource requirements for implementing the S-Box, including an analysis of CNOT, Toffoli, and NOT gate counts.

Keywords: S-Box; Galois Field; multiplication; hardware efficiency

Academic Editor: Jim Plusquellic

Received: 19 February 2025
Revised: 1 April 2025
Accepted: 2 April 2025
Published: 3 April 2025

Citation: Duong, P.-P.; Dang, T.-K.; Hoang, T.-T.; Pham, C.-K. Compact 8-Bit S-Boxes Based on Multiplication in a Galois Field $GF(2^4)$. *Cryptography* **2025**, *9*, 21. https://doi.org/10.3390/cryptography9020021

1. Introduction

The Substitution box (S-Box) is a core element in block ciphers that is responsible for introducing nonlinearity to obfuscate the link between plaintext, ciphertext, and secret keys [1]. This feature makes encryption algorithms safer against cryptanalysis tools like Differential Cryptanalysis and Linear Cryptanalysis [2]. When an S-Box is deployed in hardware, it can be implemented using either lookup tables LUTs or direct computation via logic circuits. Furthermore, S-Box operations are repeatedly used across multiple encryption rounds. For example, in AES-128 [3], each 128-bit data block undergoes 10 encryption rounds, with 16 S-Box substitutions per round. Consequently, the resource consumption of the S-Box significantly affects the hardware cost and overall performance of the encryption algorithm.

Within the field of embedded systems and IoT devices, where lightweight cryptography with minimal resource consumption is required [4], designing an efficient S-Box from the outset, rather than merely optimizing existing implementations, is crucial. The choice of an 8-bit S-Box is justified as it is the standard size used in many block cipher algorithms. S-Boxes of 4-bit, 5-bit, or smaller could further reduce hardware costs but would significantly degrade nonlinearity, making the cipher more vulnerable to differential and linear attacks. Thus, an 8×8 S-Box provides an optimal balance between security and

hardware efficiency, ensuring compliance with cryptographic criteria while remaining practical for resource-constrained devices. Resource optimization methods using SAT Solvers for implementing small-sized S-Boxes are often effective, as shown in a previous study [5]. However, for larger sizes such as 8-bit, they typically face challenges due to the complexity of optimizing an 8-variable Boolean function. Therefore, if resource optimization is the goal, optimizing from the design stage is a better approach.

Many studies have primarily focused on improving the hardware implementation of existing S-Boxes rather than developing entirely new S-Box architectures that inherently minimize resource usage [6–8]. Some studies focus on optimizing S-Box resources, but the cryptographic criteria parameters are very low [9,10].

The choice of using $GF(2^4)$ for constructing an 8×8 S-Box is driven by both mathematical efficiency and practical implementation considerations. Since $GF(2^4)$ naturally supports arithmetic operations on 4-bit elements, an 8-bit S-Box can be efficiently structured by combining two 4-bit elements, making finite field operations more suitable for this size.

For 4×4 S-Boxes, using $GF(2^4)$ is unnecessary and inefficient, as the field size exceeds the required number of input bits. If a finite field approach is used, $GF(2^2)$ would be a more appropriate choice for 4-bit designs. However, 4×4 S-Boxes generally consume very few hardware resources and their implementation can be efficiently optimized using logic minimization techniques. In contrast, an 8×8 S-Box is a Boolean function of eight variables, making it significantly more complex and challenging to optimize directly in hardware. Therefore, optimizing the S-Box at the design stage is crucial to achieving an efficient implementation.

Meanwhile, applying $GF(2^4)$ to S-Boxes of sizes such as 5×5 or 6×6 is challenging, as these sizes do not naturally align with the structure of the finite fields commonly used in cryptographic implementations. Unlike 8-bit designs, where each byte can be decomposed into two 4-bit elements for structured computation, a 5-bit or 6-bit S-Box would require a different field representation, resulting in increased complexity and reduced efficiency.

Several studies have demonstrated that multiplication in $GF(2^4)$ can be efficiently implemented using logic circuits with a minimal number of gates [11,12]. Moreover, even for implementing existing S-Boxes such as the AES S-Box, researchers often seek to transform operations from $GF(2^8)$ to $GF(2^4)$ to reduce computational costs in hardware. This highlights the potential of $GF(2^4)$ in designing S-Boxes that maintain security while optimizing resource consumption. This is also demonstrated in the study in [13]. The authors of [14] constructed the S-Box on a subset of the multiplicative group in a Galois field smaller than $GF(2^8)$ instead of the entire field to reduce size, but the results are not optimal.

In this research, we propose a novel method for designing an 8-bit S-Box using multiplication in $GF(2^4)$ with modular reduction using an irreducible polynomial. The goal of the study is to propose an S-Box generation architecture that is both hardware-optimized and meets cryptographic criteria effectively.

The following points summarize this paper's main contributions:

- Propose an architecture for generating 8-bit S-Boxes based on multiplication in $GF(2^4)$.
- Experiment with modulo multiplication using various irreducible polynomials and evaluate the generated S-Boxes. The S-Box exhibits significantly lower hardware resource consumption compared to previous studies.
- Conduct a comprehensive and extensive analysis of the S-Box, demonstrating that its cryptographic properties remain robust when compared to related works.
- Assess the hardware implementation efficiency of the selected S-Box and evaluate its resilience to side-channel attacks (SCAs).
- Estimate the number of CNOT, Toffoli, and NOT gates required for implementing the proposed S-Box in a quantum computing environment.

The structure of this paper is outlined as follows. Section 2 introduces the proposed method. Section 3 analyzes the cryptographic properties of the designed S-Box. Section 4 evaluates the hardware resource consumption of the proposed S-Box implementation. Section 5 examines the parameters and experimental results to assess its resistance against SCAs. Finally, the results are outlined in Section 6.

2. Related Works

There are various methods for designing S-Boxes, each with distinct characteristics in terms of security, computational effectiveness, and hardware implementation feasibility.

A common approach to producing S-Boxes is the use of mathematical transformations over finite fields. This method is employed in AES [3], where the S-Box is designed according to the multiplicative inverse operation combined with an affine transformation to ensure nonlinearity and uniform distribution. The advantage of this approach lies in its strong cryptographic properties; however, it requires complex arithmetic operations, making hardware implementation more challenging. A recently popular approach is the use of chaotic maps for S-Box generation [15–20]. Chaotic dynamical systems exhibit high sensitivity to initial conditions, enabling the creation of highly nonlinear and unpredictable S-Boxes. This method is suitable for applications requiring flexibility and dynamic S-Box generation instead of a fixed substitution structure. However, a notable drawback is that, for 8-bit S-Boxes, the achievable nonlinearity remains limited. As reported in [18], only a few cases achieve a nonlinearity value of 112.

Additionally, many other methods have been explored with the goal of finding S-Boxes that satisfy criteria such as high nonlinearity and resistance to linear and differential cryptanalysis [21]. As cryptographic attacks continue to evolve, research and improvements in S-Box generation methods remain a crucial area in cryptographic studies. This section will analyze research findings specifically related to hardware optimization for 8-bit S-Boxes.

The study by Canright et al. [6] proposes a compact AES S-Box design utilizing subfield arithmetic in $GF(2^4)$ and $GF(2^2)$, leading to a reduction in logic gate count. The results demonstrate a minimized S-Box implementation with 195 logic gates for the standalone design and 253 logic gates for the merged design, achieving greater optimization compared to previous approaches.

The studies in [7,8,22,23] have focused on optimizing resource consumption for the AES S-Box. One way to do this is to implement multiplication operations at the logic circuit level to cut down on the number of logic gates needed. The most efficient AES S-Box implementation to date is by Maximov [23], which achieves logic gates of 64 XOR/NOR gates, 23 NAND/OR gates, 4 AND gates, and 6 multiplexers (MUXs) [13,23].

The studies by Rashidi [13,24] use a lightweight 8-bit S-Box design using inversion in $GF(2^8)$ combined with an optimized affine transformation. To reduce hardware resource consumption, the inversion operation is performed in the composite field $GF(2^4)^2$ instead of directly in $GF(2^8)$. A key aspect of this approach is the utilization of resource sharing in the field multiplication over $GF(2^4)$, rewriting inversion equations to minimize computational complexity and integrating computational blocks within the S-Box into a unified structure. This optimization reduces circuit area and latency compared to previous designs while maintaining a level of security equivalent to the AES S-Box. The most optimized S-Box implementation achieves a hardware footprint of 60 XOR/XNOR gates and 80 NAND/NOR gates.

Reference [25] presents a low-cost 8-bit S-Box design using inversion in $GF(2^n)$ combined with an affine transformation. To optimize hardware resources, the authors apply composite field arithmetic over $GF(2^4)^2$, reducing the computational cost of the inversion process. The S-Box construction consists of two main steps: first, performing inversion

in $GF(2^4)^2$ and then applying a low-area affine transformation. However, although the paper does not provide detailed information on the number of logic gates used, the S-Box architecture still requires multiple multiplications, squaring operations, and inversions. As a result, it is not entirely resource-efficient.

Kuznetsov et al. [26] have proposed a way to make 8-bit S-Boxes that are more nonlinear by using the Hill Climbing Algorithm along with a new cost function. The goal is to cut down on the number of times the S-Box search process is run while keeping important cryptographic properties like higher nonlinearity and resistance to cryptanalytic attacks. Although the proposed method significantly improves efficiency, it achieves a maximum nonlinearity of 104, which is still lower than that of some optimally designed S-Boxes. This limitation may impact its robustness against differential and linear cryptanalysis in practical applications.

Reference [27] describes a new way to find S-Box circuits that have the best multiplicative complexity (MC-optimal). It does this by combining the A* pathfinding algorithm with a detailed study of MC computation. This method extends the search space beyond existing tools such as SAT-solvers and LIGHTER, enabling the construction of optimized circuits for 5×5 and 6×6 S-Boxes, including bijective S-Boxes, almost perfect nonlinear (APN) S-Boxes, and certain quadratic permutations. Furthermore, the study establishes new lower bounds on the complexity of multiplicative AES and MISTY S-Boxes. However, while the research provides improved theoretical bounds for the MC of 8×8 S-Boxes, it does not directly apply the MC-optimal search method to construct optimized circuits for them.

There have been many studies on generating 8-bit S-Boxes from smaller S-Boxes using Lai-Massey, Feistel, and MISTY structures [10,28–31]. However, these methods have limitations in cryptographic security and efficiency, as they rely on predefined structures and the properties of small-sizes S-Boxes. In reference [9], the authors describe a new way to make S-Boxes by combining bitwise processes that come from the identity function. This is very different from how things have been done in the past. This technique is articulated as a Markov Decision Process, with reinforcement learning functioning as an appropriate solver. The goal of the study is to train an RL agent to generate S-Boxes that can efficiently apply the masking scheme. Studies [9,10,28–31] focus only on resource optimization, resulting in S-Boxes with rather poor cryptographic properties, i.e.,with nonlinearity less than 100.

In summary, there are various approaches to designing S-Boxes, each optimizing different criteria. However, the most crucial aspect is to design an S-Box that ensures both strong cryptographic properties and hardware efficiency. This research presents an effective solution to achieve this design requirement.

3. Proposed Method

3.1. Proposed Architecture

The proposed method constructs the S-Box by dividing the 8-bit data split into two 4-bit parts, a and b, and applying transformations in $GF(2^4)$. Specifically, the upper part a, consists of bits $a_3a_2a_1a_0$, while the lower part b, consists of bits $b_3b_2b_1b_0$. The output is formed by computing A and B, concatenating them, and applying an XOR operation with $0x01$. This approach reduces computational complexity and optimizes hardware implementation while preserving the essential nonlinear properties of the S-Box.

The data processing within the S-Box occurs in two consecutive steps. First, the value B is computed based on a and b. If b is zero, B retains the value of a. If b is nonzero, a multiplication in $GF(2^4)$ between a and b is performed to generate a new value for B. Once B is determined, the value A is computed in a similar manner. If B is zero, A retains the value of b. If B is nonzero, a multiplication in $GF(2^4)$ between b and B is performed to

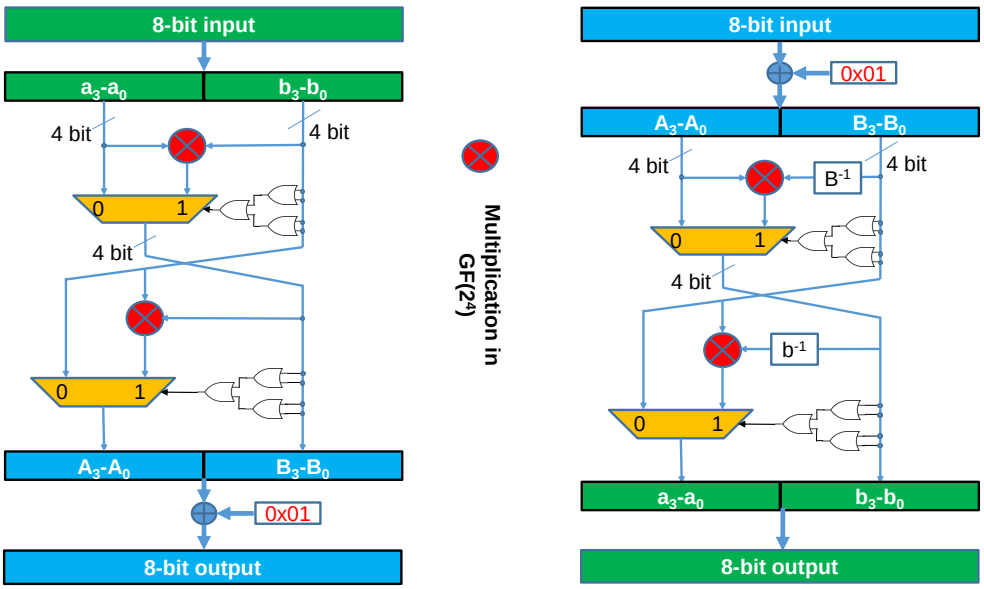
generate a new value for A. These steps are expressed mathematically as Equation (1). And Equation (2) is applied to compute the inverse S-Box.

$$\begin{aligned} B[3:0] &= \begin{cases} a[3:0] & : b[3:0] = 0 \\ a[3:0] \otimes b[3:0] & : b[3:0] \neq 0 \end{cases} \\ A[3:0] &= \begin{cases} b[3:0] & : B[3:0] = 0 \\ b[3:0] \otimes B[3:0] & : B[3:0] \neq 0 \end{cases} \end{aligned} \tag{1}$$

where $\otimes$ denotes multiplication in $GF(2^4)$, performed modulo an irreducible polynomial of degree 4.

$$\begin{aligned} b[3:0] &= \begin{cases} A[3:0] & : B[3:0] = 0 \\ A[3:0] \otimes B^{-1}[3:0] & : B[3:0] \neq 0 \end{cases} \\ a[3:0] &= \begin{cases} B[3:0] & : b[3:0] = 0 \\ B[3:0] \otimes b^{-1}[3:0] & : b[3:0] \neq 0 \end{cases} \end{aligned} \tag{2}$$

Figure 1 illustrates the hardware architecture of the proposed S-Box and invert S-Box. In this design, $GF(2^4)$ multiplications play a key role in generating the transformed values of A and B. However, to avoid unnecessary computations, the system employs multiplexer (MUX) units combined with OR gates to check whether each 4-bit input is zero. Specifically, if an input is zero, the MUX selects the unchanged value; otherwise, the MUX selects the result of the $GF(2^4)$ multiplication.



(a) S-box (b) Invert S-box

Figure 1. Proposed architecture for S-Box construction.

Upon completing the processing steps, the two 4-bit parts, A and B, are concatenated to form the final 8-bit output of the S-Box. By leveraging this data-splitting approach, the proposed architecture not only optimizes the number of operations required in $GF(2^4)$ but also significantly reduces hardware resource consumption.

3.2. S-Box Construction

This section does not focus on the mathematical theory of multiplication in the Galois Field in detail. The study does not explore hardware optimization for implementing these multiplications. Many previous studies have already addressed this issue [11,12]. The

primary objective of this section is to select and utilize multiplication effectively for S-Box construction. To multiply two-term polynomials, the elements in $GF(2^4)$ must be multiplied, which necessitates the use of a fourth-degree irreducible polynomial. Multiplication in $GF(2^4)$ is defined as Equation (3).

$$q(x) = a(x) \otimes b(x) \mod p(x) \tag{3}$$

In this context, $q(x)$, $a(x)$, and $b(x)$ are considered elements of the Galois Field $GF(2^4)$ expressed in polynomial form, as seen in Equation (4).

$$\begin{aligned} q(x) &= \sum_{i=0}^{3} q_i x^i = q_3 x^3 + q_2 x^2 + q_1 x + q_0, \\ a(x) &= \sum_{i=0}^{3} a_i x^i = a_3 x^3 + a_2 x^2 + a_1 x + a_0, \\ b(x) &= \sum_{i=0}^{3} b_i x^i = b_3 x^3 + b_2 x^2 + b_1 x + b_0. \end{aligned} \tag{4}$$

By solving Equation (5), one can find the inverse $a^{-1}(x)$ of an element $a(x) \in GF(2^4)$, which yields Equation (6).

$$a(x) \otimes a^{-1}(x) \mod p(x) = 1 \tag{5}$$

$$q(x) = a(x)^{-1} \mod p(x) \tag{6}$$

The outcomes of the resource evaluation for multiplication operations are comprehensively detailed in Table 1. The results demonstrate that employing either the irreducible polynomial $p_1(x) = x^4 \oplus x \oplus 1$ or $p_2(x) = x^4 \oplus x^3 \oplus 1$ for modular reduction in the multiplication operation yields an identical count of logic operations.

In the computation of the inverse S-Box, an additional inversion operation is required, as illustrated in Part (b) of Figure 1. Therefore, the implementation of this operation is also evaluated in detail in Table 2. The results indicate that the quantity of logical operations remains similar.

Based on these findings, the study will experiment with generating the S-Box using both $p_1(x)$ and $p_2(x)$. The corresponding S-Boxes are generated and evaluated based on key cryptographic properties to determine the most suitable polynomial for the design.

Since each S-Box involves two multiplication operations in $GF(2^4)$, the choice of irreducible polynomials affects both computations. Two different irreducible polynomials are considered for these multiplications, resulting in four distinct S-Box implementations. The selection methods for these four cases are presented in Table 3. Through the evaluation of these four cases, it was observed that the S-Boxes constructed using the same irreducible polynomial for both multiplications exhibited better cryptographic properties than those using different polynomials. The highest nonlinearity achieved was 112. Based on this result, the S-Box generated in Case S-Box 1, presented in Table 4, was selected for detailed evaluation. Note that the calculation results for the inverse S-Box are applied in exactly the same manner, so the results are not presented in this section.

Table 1. Computation of $a(x) \otimes b(x)$ for different irreducible polynomials $p(x)$.

$p(x)$	$q(x) = a(x) \otimes b(x) \mod p(x)$	Logic Gates
$x^4 \oplus x \oplus 1$	$T_1 = a_0 \oplus a_3$, $T_2 = a_2 \oplus a_3$, $q_0 = a_0b_0 \oplus a_2b_2 \oplus a_3b_1 \oplus a_1b_3$, $q_1 = a_1b_0 \oplus T_1b_1 \oplus T_2b_2 \oplus b_3(a_1 \oplus a_2)$, $q_2 = a_1b_1 \oplus a_2b_0 \oplus T_1b_2 \oplus T_2b_3$, $q_3 = a_1b_2 \oplus a_2b_1 \oplus a_3b_0 \oplus T_1b_3$.	15 XOR gates, 16 2-input AND gates
$x^4 \oplus x^3 \oplus 1$	$T_1 = a_3 \oplus a_2$, $T_2 = T_1 \oplus a_1$, $q_0 = T_2b_3 \oplus T_1b_2 \oplus a_3b_1 \oplus a_0b_0$, $q_1 = T_1b_3 \oplus a_3b_2 \oplus a_0b_1 \oplus a_1b_0$, $q_2 = a_3b_3 \oplus a_0b_2 \oplus a_1b_1 \oplus a_0b_3$, $q_3 = (T_2 \oplus a_0)b_3 \oplus T_2b_2 \oplus T_1b_1 \oplus a_3b_0$.	15 XOR gates, 16 2-input AND gates

Table 2. Computation of $a^{-1}(x)$ for different irreducible polynomials $p(x)$.

$p(x)$	$q(x) = a(x)^{-1} \mod p(x)$	Logic Gates
$x^4 \oplus x \oplus 1$	$T_1 = a_1 \oplus a_2 \oplus a_3 \oplus a_1a_2a_3$, $q_0 = T_1 \oplus a_0 \oplus a_0a_2 \oplus a_1a_2 \oplus a_0a_1a_2$, $q_1 = a_3 \oplus a_0a_1 \oplus a_0a_2 \oplus a_1a_2 \oplus a_1a_3 \oplus a_0a_1a_3$, $q_2 = a_2 \oplus a_3 \oplus a_0a_1 \oplus a_0a_2 \oplus a_0a_3 \oplus a_0a_2a_3$, $q_3 = T_1 \oplus a_0a_3 \oplus a_1a_3 \oplus a_2a_3$.	20 XOR gates, 10 2-input AND gates
$x^4 \oplus x^3 \oplus 1$	$T_1 = a_3 \oplus a_0a_3 \oplus a_2a_3 \oplus a_1a_2a_3$, $T_2 = a_1 \oplus a_1a_2 \oplus a_0a_3 \oplus a_0a_2a_3$, $q_0 = T_1 \oplus a_0 \oplus a_0a_1 \oplus a_0a_2a_3$, $q_1 = T_1 \oplus a_2 \oplus a_1a_2 \oplus a_0a_1a_2 \oplus a_0a_1a_3$, $q_2 = T_2 \oplus a_2 \oplus a_0a_1 \oplus a_1a_3 \oplus a_2a_3 \oplus a_0a_1a_2$, $q_3 = T_2 \oplus a_0a_2 \oplus a_0a_1a_3$.	20 XOR gates, 10 2-input AND gates

Table 3. Different S-Box implementations based on polynomial selection in $GF(2^4)$.

Case	$a[3:0] \otimes b[3:0]$	$b[3:0] \otimes B[3:0]$
S-Box 1	$p_1(x) = x^4 \oplus x \oplus 1$	$p_1(x) = x^4 \oplus x \oplus 1$
S-Box 2	$p_1(x) = x^4 \oplus x \oplus 1$	$p_2(x) = x^4 \oplus x^3 \oplus 1$
S-Box 3	$p_2(x) = x^4 \oplus x^3 \oplus 1$	$p_1(x) = x^4 \oplus x \oplus 1$
S-Box 4	$p_2(x) = x^4 \oplus x^3 \oplus 1$	$p_2(x) = x^4 \oplus x^3 \oplus 1$

Table 4. Proposed S-Box.

i/j	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	01	11	21	31	41	51	61	71	81	91	A1	B1	C1	D1	E1	F1
1	00	10	43	52	35	24	77	66	C9	D8	8B	9A	FD	EC	BF	AE
2	03	23	85	A7	69	4B	ED	CF	B2	90	36	14	DA	F8	5E	7C
3	02	32	C7	F4	5D	6E	9B	A8	7A	49	BC	8F	26	15	E0	D3
4	05	45	39	7D	C2	86	FA	BE	57	13	6F	2B	94	D0	AC	E8
5	04	54	7B	2E	F6	A3	8C	D9	9F	CA	E5	B0	68	3D	12	47
6	07	67	BD	DB	AA	CC	16	70	E4	82	58	3E	4F	29	F3	95
7	06	76	FF	88	9E	E9	60	17	2C	5B	D2	A5	B3	C4	4D	3A
8	09	89	62	EA	B7	3F	D4	5C	AD	25	CE	46	1B	93	78	F0
9	08	98	20	B9	83	1A	A2	3B	65	FC	44	DD	E7	7E	C6	5F
A	0B	AB	E6	4C	DF	75	38	92	1E	B4	F9	53	C0	6A	27	8D
B	0A	BA	A4	1F	EB	50	4E	F5	D6	6D	73	C8	3C	87	99	22
C	0D	CD	5A	96	74	B8	2F	E3	FB	37	A0	6C	8E	42	D5	19
D	0C	DC	18	C5	40	9D	59	84	33	EE	2A	F7	72	AF	6B	B6
E	0F	EF	DE	30	1C	F2	C3	2D	48	A6	97	79	55	BB	8A	64
F	0E	FE	9C	63	28	D7	B5	4A	80	7F	1D	E2	A9	56	34	CB

3.3. Comparison

Table 5 compares the number of logic gates used in different studies, where the design of this work utilizes 31 XOR gates, 32 AND gates, 6 OR gates, and 2 MUX21s, without using NAND/NOR gates. This design is relatively simple, with fewer total gates compared to other studies. Notably, this study has 31 XOR gates, far fewer than all others. For example, Zhang [22] uses 154 XOR gates, Canright [6] uses 91 XOR gates, and even the study with the lowest XOR count before this, Rashidi [13], still uses 57 XOR gates. Since XOR gates have a higher resource cost than other logic gates, the significant reduction in XOR usage allows this design to save substantial hardware resources.

Here, we only make comparisons with studies that provide detailed information on the quantity of logic gates used in the S-Box design. The area of logic gates varies across different technologies, leading to corresponding differences in the Gate Equivalent (GE) count. Based on STM 65 nm technology [8], the GE count of this design is calculated to be 115.00, the lowest among all studies compared. Compared to the previous best result in [23] (168.00 GE), this design reduces 53 GEs, equivalent to a 31.5% reduction, demonstrating significant improvements in hardware resource efficiency. This outcome further demonstrates that the resource utilization of the S-Box produced in this research is merely around 60% relative to that in [13].

Table 5. Benchmark comparison of logic gate usage for the proposed S-box and S-boxes from other studies.

Studies	XOR/ XNOR	NAND/ NOR	AND	OR	NOT	MUX21	GE (*)
Canright [6]	91	36	0	0	0	0	218.00
Ueno [7]	87	0	54	0	0	0	241.50
Reyhani [8]	79	41	0	0	0	0	199.00
Rashidi [13]	57	80	0	0	0	0	194.00
Zhang [22]	154	0	36	0	0	8	369.00
Maximov [23]	64	23	4	0	0	6	168.00
Rashidi [24]	76	56	0	0	0	0	208.00
Kuznyechick [32]	90	0	79	28	29	0	342.25
Teng [33]	107	10	38	7	5	8	301.75
Proposed S-box	31	0	32	6	0	2	115.00

* The Gate Equivalent (GE) is evaluated based on STM 65 nm [8], where: XOR/XNOR = 2GEs, AND = 1.25GEs, OR = 1.5GEs, NAND = NOR = 1GE, NOT= 0.75 GEs, MUX21 = 2GEs.

In this section, hardware resources have been evaluated and compared from the design stage. Next, a detailed assessment of the cryptographic characteristics of the selected S-Box will be conducted.

4. Security Analysis

Numerous critical attributes necessary for a cryptographically robust S-Box have been presented in [1,2,34,35]. Rather than presenting the complete formulas for determining these criteria, we provide only the analytical results. These formulas are widely available in most S-Box literature. The program for analyzing all parameters of the S-Box has been developed and can be accessed here https://github.com/dpp291187/S-Box-Cryptanalysis (accessed on 1 April 2025).

4.1. Nonlinearity

The nonlinearity (NL) of an S-Box is crucial in assessing its resistance to linear cryptanalysis [1]. An S-Box with high nonlinearity provides better security. It enhances resistance against both linear and differential cryptanalysis.

There is no strict lower bound for the nonlinearity of an S-Box. However, research commonly considers a nonlinearity value of 100 as sufficient. The theoretical upper bound

for the nonlinearity of an 8-variable Boolean function is 120. However, this value can only be achieved for unbalanced functions. In practice, the Boolean functions that constitute an S-Box are balanced, meaning they have an equal number of output bits set to 0 and 1. For balanced functions, the highest nonlinearity found so far is 116 [36]. Therefore, the maximum achievable nonlinearity for an 8-bit S-Box is 116. Studies indicate that a nonlinearity of 112 is already highly effective for practical cryptographic applications.

As shown in Table 6, the selected S-Box maintains a uniform nonlinearity value of 112 across all eight Boolean functions, indicating a strong resistance to linear approximations. This level of nonlinearity ensures that any attempt to approximate the S-Box output using affine functions remains computationally infeasible.

Additionally, the algebraic degree (AD) is determined to be 7, further reinforcing its resistance against algebraic cryptanalysis by complicating the process of deriving algebraic relations. These features collectively illustrate the resilience of the S-Box in cryptographic applications requiring both high security and efficient implementation.

Table 6. Boolean function nonlinearities of the proposed S-Box.

Function	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8
AD	7	7	7	7	7	7	7	7
NL	112	112	112	112	112	112	112	112

4.2. Strict Avalanche Criterion

The Strict Avalanche Criterion (SAC) is a crucial characteristic of cryptographic S-Boxes, guaranteeing that a 1-bit alteration in the input yields an approximately 50% likelihood of altering each output bit. An S-Box is deemed sufficiently random when its SAC value is close to 0.5. By utilizing the equation in [28] to compute the SAC for each output function of the S-Box, we obtain the results presented in Table 7. The computed average SAC value is 0.501, which is regarded as nearly optimal.

Table 7. Detailed Strict Avalanche Criterion values of the proposed S-Box.

i/j	1	2	3	4	5	6	7	8
1	0.4375	0.5000	0.5000	0.5000	0.5625	0.5000	0.5000	0.5000
2	0.4375	0.5000	0.5000	0.5000	0.5000	0.5625	0.5000	0.5000
3	0.4375	0.5000	0.5000	0.5000	0.5000	0.5000	0.5625	0.5000
4	0.4375	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5625
5	0.5625	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000
6	0.5000	0.5625	0.5000	0.5000	0.5000	0.5000	0.4375	0.5000
7	0.5000	0.5000	0.5625	0.5000	0.5000	0.4375	0.5000	0.5000
8	0.5000	0.5000	0.5000	0.5625	0.4375	0.5000	0.5000	0.5000

4.3. Bit Independence Criterion

The Bit Independence Criterion (BIC) is a crucial measure for evaluating the independence between the output bits of an S-Box when the individual input bits are altered. This criterion is assessed based on two key properties: the Strict Avalanche Criterion (BIC-SAC) and the nonlinearity (BIC-NL).

The evaluation of these properties can be performed quickly and easily, giving us the BIC-NL and BIC-SAC values shown in Tables 8 and 9, respectively. On average, the BIC-NL is calculated to be 107.14, while the BIC-SAC achieves a value of 0.478, which is considered a good result according to the BIC-SAC criterion.

Table 8. Nonlinearity BIC results (BIC-NL) of the proposed S-Box.

i/*j*	1	2	3	4	5	6	7	8
1	-	112	112	112	104	112	104	104
2	112	-	104	104	104	112	104	096
3	112	104	-	104	104	104	104	104
4	112	104	104	-	096	112	112	112
5	104	104	104	096	-	112	112	112
6	112	112	104	112	112	-	112	112
7	104	104	104	112	112	112	-	112
8	104	096	104	112	112	112	112	-

Table 9. Strict Avalanche Criterion values for BIC (BIC-SAC).

i/*j*	1	2	3	4	5	6	7	8
1	-	0.5078	0.5078	0.5078	0.4062	0.5156	0.4531	0.4609
2	0.5078	-	0.5078	0.5078	0.4531	0.4609	0.4531	0.4531
3	0.5078	0.5078	-	0.5078	0.4531	0.4531	0.4531	0.4531
4	0.5078	0.5078	0.5078	-	0.4609	0.4531	0.4609	0.4062
5	0.4062	0.4531	0.4531	0.4609	-	0.5156	0.5156	0.5078
6	0.5156	0.4609	0.4531	0.4531	0.5156	-	0.5156	0.5156
7	0.4531	0.4531	0.4531	0.4609	0.5156	0.5156	-	0.5156
8	0.4609	0.4531	0.4531	0.4062	0.5078	0.5156	0.5156	-

4.4. Differential Approximation Probability

The security of an S-Box against differential cryptanalysis is measured by its Differential Uniformity, which indicates how uniformly output differences (Δy) are distributed for given input differences (Δx). The detailed method for computing DU is provided in [2,35]. The XOR distribution table is defined in Equation (7).

$$D(\Delta x, \Delta y) = \#\{x \mid S(x) \oplus S(x \oplus \Delta x) = \Delta y\} \tag{7}$$

where $S(x)$ is the S-Box output for input x.

The XOR table values of the proposed S-Box are calculated in Table 10. The XOR distribution table analyzes an S-box's resistance to differential cryptanalysis by showing how output differences (Δy) distribute for given input differences (Δx). Rows represent Δx values, columns represent Δy values, and each cell shows how often a specific ($\Delta x, \Delta y$) pair occurs. The maximum value in the table (δ) indicates the differential uniformity, where lower δ values mean better resistance. In this case, $\delta = 18$.

Table 10. XOR distribution table of the proposed S-Box.

0	18	16	16	16	16	16	16	16	16	16	16	16	16	16	16
18	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14
16	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14
16	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14
16	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14
16	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14
16	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14
16	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14
16	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14
16	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14
16	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14
16	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14
16	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14
16	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14
16	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14
16	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14

The differential uniformity is given by Equation (8).

$$\delta = \max_{\Delta x \neq 0, \Delta y} D(\Delta x, \Delta y) \tag{8}$$

A lower δ value indicates better resistance to differential attacks. From Table 10, the Differential Uniformity of the proposed S-Box is $\delta = 18$.

The Differential Approximation Probability (DP) measures the probability of predicting the output difference corresponding to a given input difference in an S-Box. This property is very important for figuring out how resistant an S-Box is to differential cryptanalysis, since lower DP values mean that the S-Box is safer from these kinds of attacks. The Differential Approximation Probability (DP) is calculated as $DP = \delta/2^n = 18/256 = 0.070$.

4.5. Linear Approximation Probability

The Linear Approximation Probability (LP) measures how likely it is that certain input and output bits of an S-Box will have a linear relationship. This metric is very important for figuring out how resistant an S-Box is to linear cryptanalysis. Lower LP values mean that linear approximations are less likely to work. The LP value of the proposed S-Box is determined to be 0.125. The steps for computing LP are explained in [2,35], which allows a thorough check of the S-Box's defenses against linear attacks.

A related concept is the linear structure of an S-Box, which refers to the existence of input–output pairs that satisfy a linear equation (Equation (9)).

$$S(x) \oplus S(x \oplus a) = b, \quad \forall x \in GF(2^n). \tag{9}$$

If an S-Box has a strong linear structure, it implies that there exist certain values a, b, such that the output transformation remains predictable under XOR operations. This can significantly weaken the S-Box against linear cryptanalysis. Generally, a lower LP value suggests that the S-Box has minimal linear structure, making it more resistant to attacks. Since the proposed S-Box achieves an LP of 0.125, it indicates that no strong linear structure exists, further enhancing its security.

The data in Table 11 indicate that the proposed S-Box attains a nonlinearity (NL) of 112.00, which is equivalent to the best-performing S-Boxes in the comparison, including those from studies [3,13,37]. This indicates a good resistance against linear cryptanalysis. Compared to other studies such as [15,16,19,20,38–41], the proposed S-Box demonstrates superior nonlinearity, highlighting its cryptographic strength.

Table 11. Comparison with alternative S-Boxes.

S-Box	NL	BIC-NL	SAC	BIC-SAC	DP	LP	FP	OFP
AES [3]	112.00	112.00	0.5058	0.5040	0.016	0.063	0	0
Radishi [13]	112.00	112.00	0.5078	0.5003	0.016	0.063	0	1
Manzoor [15]	110.00	103.50	0.5034	0.5046	0.039	0.133	2	1
Malik [16]	106.75	112.00	0.5027	0.5020	0.039	0.133	2	0
Lawah [17]	109.00	112.00	0.4936	0.5050	0.039	0.117	0	1
Shafique [19]	104.25	103.86	0.4992	0.4997	0.046	0.133	2	2
Zhang [20]	106.50	105.60	0.5060	0.5069	0.039	0.133	0	1
Corona [42]	104.25	104.00	0.5029	0.5026	0.047	0.125	0	0
Zahid [38]	106.75	103.93	0.5070	0.4997	0.054	0.140	0	0
Zhu [39]	105.75	104.14	0.5022	0.5051	0.039	0.133	0	1
Zahid [40]	107.00	103.05	0.4968	0.5040	0.039	0.156	0	0
Baowidan [43]	112.00	112.00	0.5049	0.5046	0.016	0.063	2	1
Yang [41]	107.75	103.14	0.4950	0.5034	0.039	0.140	2	1
Chew [37]	112.00	112.00	0.4980	0.4981	0.016	0.063	0	0
This work	112.00	107.14	0.5009	0.4780	0.070	0.125	0	0
Ideal value	**High**	**High**	**0.5000**	**0.5000**	**Low**	**Low**	**0**	**0**

Additionally, the BIC-NL value of this work does not reach the maximum value of 112.00. It still outperforms several other S-Boxes, including those from studies [15,16,19,20,38–41]. This signifies that the S-Box satisfies this condition effectively.

Concerning the Strict Avalanche Criterion, the proposed S-Box attains a value of 0.5009. This value is nearly equivalent to the optimal value of 0.5000. While it does not exactly match the optimal value, the deviation is minimal compared to the remaining S-Boxes. This indicates that the S-Box maintains good balance, ensuring that each output bit has an almost optimal probability of changing when a small input modification occurs. Furthermore, the LP value of this study is relatively low (0.125), enhancing its resistance against linear attacks, although it is not the lowest in the comparison.

A key advantage of the proposed S-Box is the absence of fixed points and opposite fixed points. The fixed point (FP) count is zero, written as $FP = 0$. The opposite fixed point (OFP) count is also zero, denoted as $OFP = 0$. A fixed point is a value x where $S(x) = x$. An opposite fixed point is a value x where $S(x) = \bar{x}$. The term $\bar{x}$ refers to the bitwise complement of x. The presence of fixed points or opposite fixed points can weaken cryptographic systems. Such weaknesses increase vulnerability to certain attacks. The fact that the proposed S-Box has neither FP nor OFP indicates that it does not create predictable input–output patterns, thereby strengthening its security.

However, one limitation of the proposed S-Box is its DP value of 0.070, which is higher than other DP values. This may impact its resistance against differential attacks, although its high nonlinearity partially compensates for this drawback.

The research results indicate that the suggested S-Box is designed for efficient hardware implementation while preserving robust cryptographic features. Specifically, it exhibits high nonlinearity, near-ideal SAC, and no fixed points, contributing to its robustness in cryptographic applications. Although there is room for improvement in the DP criterion, the overall performance indicates that this S-Box remains a strong candidate compared to existing designs.

5. Implementation

5.1. FPGA-Based Implementation

The suggested S-Box is executed on an FPGA platform. It is specifically implemented on the Kintex 7-XC7K160T. The S-Box functions as a substitute for the Rijndael S-Box in the AES encryption technique. The implementation specifics are specified in [3]. The original S-Box is implemented using a lookup table (LUT)-based approach. To ensure efficient execution, the AES-128 encryption process is designed using a loop-unrolled technique for the full 10-round encryption. The hardware implementation results are presented in Table 12, providing a comparative analysis of the resource utilization. Notably, the results show that replacing the AES S-Box with the proposed S-Box does not increase hardware resource consumption. This holds true when implemented on the FPGA. The findings confirm its feasibility for integration into AES-based cryptographic systems.

Table 12. Implementation results.

	LUTs	BRAM	FF	F_{max} (MHz)	Throughput (Gbps)
Proposed S-Box	15	0	0	-	-
Rijndael S-Box	32	1	24	-	-
AES (Proposed S-Box)	950	0	264	311.487	3.98
AES (Rijndael S-Box)	1386	1	406	344.697	4.40

Based on the implementation results presented in Table 12, the following detailed observations can be made.

First, the standalone implementation of the proposed S-Box requires only 15 LUTs. This is significantly lower than the 32 LUTs needed for the Rijndael S-Box. The reduction

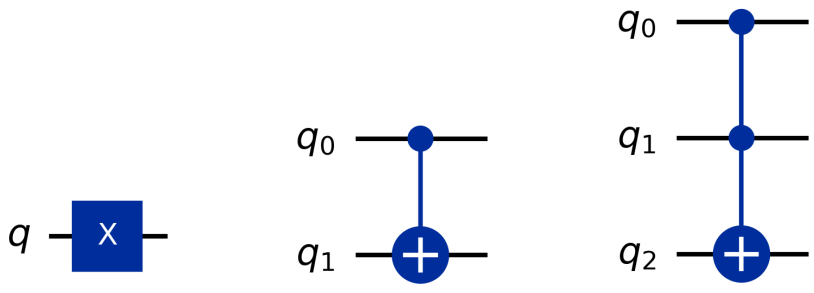
corresponds to approximately 53.1% fewer LUTs. This demonstrates a significant reduction in hardware resource utilization compared to the traditional lookup table (LUT)-based approach. Second, when integrating the proposed S-Box into the AES-128 encryption algorithm, the total number of LUTs is reduced by 31.4%, from 1386 to 950. Similarly, the number of Flip-Flops (FFs) decreases by 35.0%, from 406 to 264. Additionally, the proposed S-Box eliminates the need for BRAM, reducing memory usage from 1 to 0, which is particularly beneficial for FPGA platforms with limited memory resources.

However, a notable drawback is the decrease in the maximum operating frequency ($F_{\max}$). The AES implementation using the proposed S-Box achieves an $F_{\max}$ of 311.487 MHz, which is 9.6% lower than the 344.697 MHz achieved with the Rijndael S-Box. This reduction may impact the overall system performance. Regarding throughput, the AES implementation with the proposed S-Box achieves 3.98 Gbps, compared to 4.40 Gbps for the AES implementation using the Rijndael S-Box, representing a 9.6% reduction. While the proposed S-Box significantly reduces hardware resource consumption, it may lead to a slight decrease in computational performance.

In summary, the proposed S-Box offers substantial advantages in reducing LUT, FF, and BRAM usage compared to the traditional lookup table-based Rijndael S-Box implementation. However, the decrease in maximum operating frequency and throughput should be carefully considered when selecting an implementation approach for high-performance applications.

5.2. Quantum Implementation

In recent years, the implementation of S-Boxes on quantum circuits has emerged as a significant research direction in post-quantum cryptography. Numerous studies have focused on optimizing S-Box execution by reducing the number of quantum gates, minimizing circuit depth, or decreasing the number of required qubits to mitigate errors and optimize resource usage on quantum hardware. However, each approach entails trade-offs between gate count, circuit depth, and qubit requirements. Some recent studies on gate optimization for AES S-Box quantum circuits can be found in [44–46]. In this study, we do not apply any optimization techniques but instead focus on evaluating the resource cost of implementing an S-Box on a quantum circuit. Specifically, we analyze the required qubit count, the number of quantum gates utilized (CNOT, Toffoli, and NOT gates), and the Toffoli depth. These basic gates are illustrated in Figure 2.



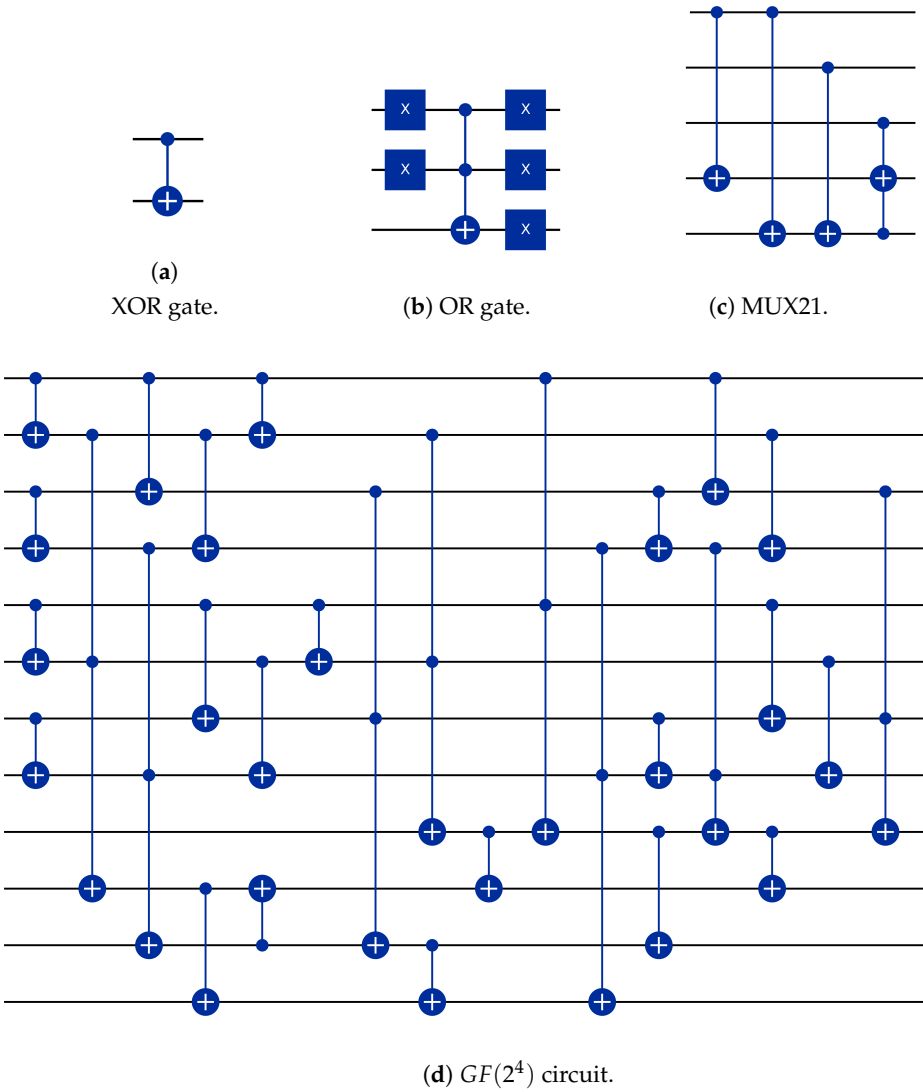
(**a**) NOT gate. (**b**) CNOT gate. (**c**) Toffoli gate.

Figure 2. Base quantum gates. (**a**) The NOT gate. (**b**) The CNOT gate. (**c**) The Toffoli gate.

In the proposed S-Box design as in Figure 1, the circuit is composed of the following components: two multiplications in $GF(2^4)$, six two-input OR gates, two 2:1 multiplexers (MUX21s), and one XOR gate. These components are implemented following the approach of Li [45], using 20 qubits. The quantum circuit components of the proposed S-Box circuit are illustrated in Figure 3. Each $GF(2^4)$ multiplication requires 23 CNOT gates and 9 Toffoli gates, so two multiplications consume a total of 46 CNOT gates and 18 Toffoli gates. Each

OR gate is realized with 5 NOT gates and 1 Toffoli gate, which amounts to 30 NOT gates and 6 Toffoli gates for all 6 OR gates. Each MUX21 gate requires three CNOT gates and one Toffoli gate, resulting in six CNOT gates and two Toffoli gates for the two MUX21 units, while the XOR gate is equivalent to one CNOT gate. Summing up these contributions, the proposed S-Box circuit requires approximately 26 Toffoli, 53 CNOT, and 30 NOT gates. These resource estimates may increase slightly if additional uncomputation steps are needed, yet remain within the same order of magnitude, demonstrating a balanced trade-off between qubit count and gate cost in the implementation proposed by Li [45].

The results in Table 13 highlight significant differences between the proposed S-Box circuit and previous AES S-Box implementations. The proposed design requires 20 qubits, which is comparable to some previous works.



(**a**) XOR gate. (**b**) OR gate. (**c**) MUX21.

(**d**) $GF(2^4)$ circuit.

Figure 3. Quantum circuit representations: (**a**) XOR gate; (**b**) OR gate; (**c**) MUX21 gate; (**d**) $GF(2^4)$ circuit.

Table 13. Quantum resources comparison.

Schemes	#Qubits	#Toffoli	#CNOT	#NOT	Toffoli Depth
Chen [44]	20	43	196	4	23
Li [45]	20	44	197	4	32
Langenberg [46]	32	55	314	4	40
This work	20	26	53	30	29

In terms of Toffoli gate count, the proposed S-Box circuit requires only 26 gates, significantly reducing computational overhead compared to AES S-Box implementations. Similarly, the number of CNOT gates in this design (53 gates) is drastically lower than in previous studies, where the required count ranges from 196 to 314 gates. This suggests that the proposed circuit achieves a more resource-efficient quantum realization, reducing both gate complexity and execution costs on quantum hardware.

However, the circuit exhibits a significantly higher X-gate (NOT-gate) count, reaching 30 gates, which is notably greater than in AES S-Box designs, where only 4 gates are required. Regarding Toffoli depth, the proposed circuit achieves a depth of 29, which is higher than in previous works.

Overall, the results demonstrate that the proposed S-Box circuit effectively reduces the number of Toffoli and CNOT gates, making it a promising candidate for resource-efficient quantum implementations. However, in this study, our primary focus is not on optimizing the quantum S-Box design itself but rather on providing a fundamental evaluation of its resource requirements in terms of gate counts. Specifically, we concentrate on quantifying the number of CNOT, Toffoli, and NOT gates employed in our proposed S-Box implementation. Other critical aspects of quantum circuit performance, such as overall circuit depth, qubit connectivity, error propagation, and fault tolerance overhead, are not addressed in the present work. These factors, while important for practical quantum computing, are beyond the scope of our current evaluation and will be the subject of future investigations.

6. Side Channel Attack Analysis

In modern cryptographic research, resistance to side-channel attacks (SCAs) is a key factor in evaluating security levels [47]. Correlation Power Analysis (CPA) [48] is a well-known type of side-channel attack that exploits power consumption leakages from cryptographic hardware. This attack technique exploits the relationship between a device's power consumption and the data being processed to retrieve secret keys. CPA is considered one of the most effective techniques for retrieving cryptographic keys from hardware-based encryption implementations, including FPGAs, microcontrollers, and smart cards.

In this experiment, we conducted a CPA attack targeting the final round of AES-128 implemented on the FPGA Sakura X board. The main objective was to evaluate the S-Box's resilience to an SCA. Unlike conventional discussions that focus on the theoretical principles of CPA, this section emphasizes the practical aspects, including the experimental setup and attack methodology.

The attack specifically targeted the last round of AES-128, which was implemented in its standard form without any countermeasures against the SCA. The system model used for acquiring power traces is depicted in Figure 4.

The AES-128 encryption algorithm was implemented on the FPGA using a single-cycle-per-round architecture. This design ensures that each encryption round is completed in a single clock cycle, which facilitates efficient data processing and power measurement.

The attack was based on the Hamming Distance power consumption model, as referenced in [49,50]. This model posits that power consumption correlates with the frequency of bit transitions (0 to 1 or 1 to 0) occurring during cryptographic procedures.

To conduct the attack, plaintexts were randomly generated on a computer, while the encryption key remained fixed as a 16-byte sequence: `[01, 02, 03, 04, 05, 06, 07, 08, 09, 0A, 0B, 0C, 0D, 0E, 0F]` (expressed in hexadecimal notation).

Each 16-byte plaintext-key pair was transmitted from the computer to the FPGA, where the encryption process was executed. The corresponding ciphertexts were then computed and stored. Simultaneously, an oscilloscope was used to measure the power traces associated

with each encryption operation. The computer was connected to both the FPGA and the oscilloscope to collect and synchronize the ciphertexts with their respective power traces.

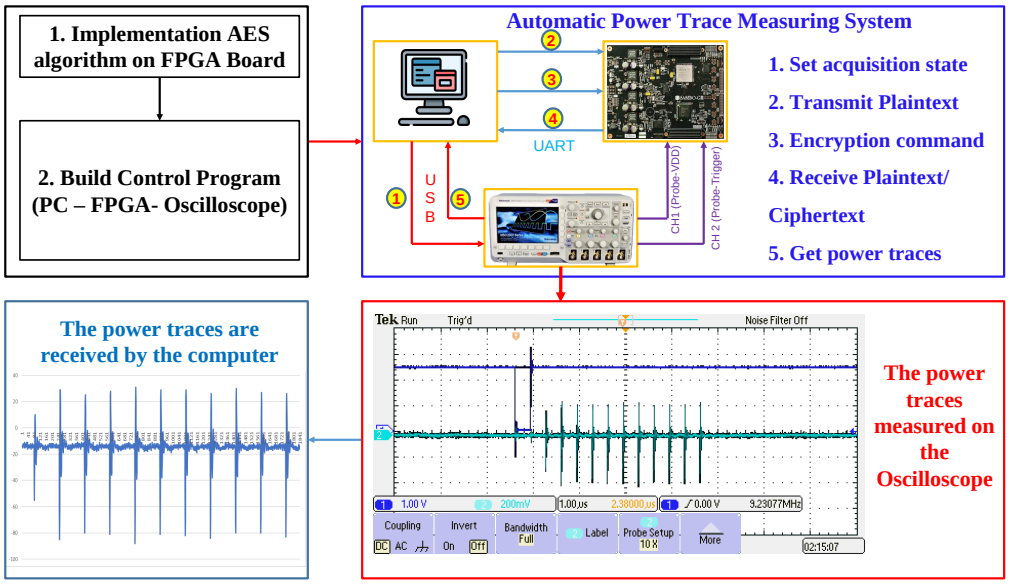


Figure 4. Power trace acquisition system for side-channel attacks.

A total of 30,000 power traces were collected during the attack. The CPA analysis focused on the final round of AES, where the last SubBytes operation and the final key addition occur. This round is particularly vulnerable because the relationship between the key and the processed data is more straightforward compared to earlier rounds.

The attack process involved the following steps:

- Trace Acquisition: The oscilloscope captured power consumption data corresponding to each encryption operation.
- Hypothesis Generation: Possible key byte values were hypothesized, and their expected power consumption patterns were computed using the Hamming Distance model.
- Correlation Computation: The correlation coefficient between the observed power traces and the anticipated power consumption values was computed for each potential key hypothesis.
- Key Recovery: The key hypothesis that exhibited the highest correlation was identified as the most likely key value.

To evaluate the effectiveness of the S-Box in this research in resisting CPA attacks, we examined the number of traces necessary for successful key recovery in two scenarios:

- AES-128 implemented with the standard AES S-Box (Rijndael S-Box).
- AES-128 implemented with the proposed S-Box.

For the AES standard S-Box, 9000 power traces were sufficient to recover 14/16 (87.5%) of the key bytes. In this case, key byte 11 required the highest number of traces (11,000 power traces), while key byte 8 required the least (4000 power traces). Figure 5 depicts the correlation coefficient in relation to the quantity of power traces for all 16 key bytes of the AES S-Box.

For the proposed S-Box, 16,000 power traces were sufficient to recover 81.25% (13/16) of the key bytes. Among these, key byte 6 required the highest number of traces (27,000 traces), whereas key byte 9 required the lowest (7000 traces). The correlation coefficient for key byte 6 is visualized in Figure 6. The correlation coefficient with the highest absolute value corresponds to the correct key. Figures 7 and 8 show the relationship between the correlation coefficient and the number of power traces, corresponding to key byte 6 and all 16 key bytes, respectively.

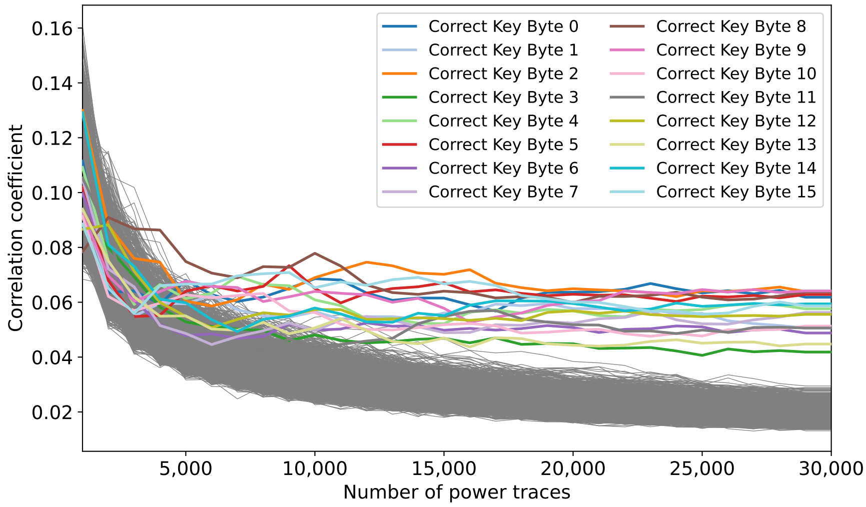


Figure 5. The correlation coefficient and number of traces plot of 16 key bytes of the AES S-Box.

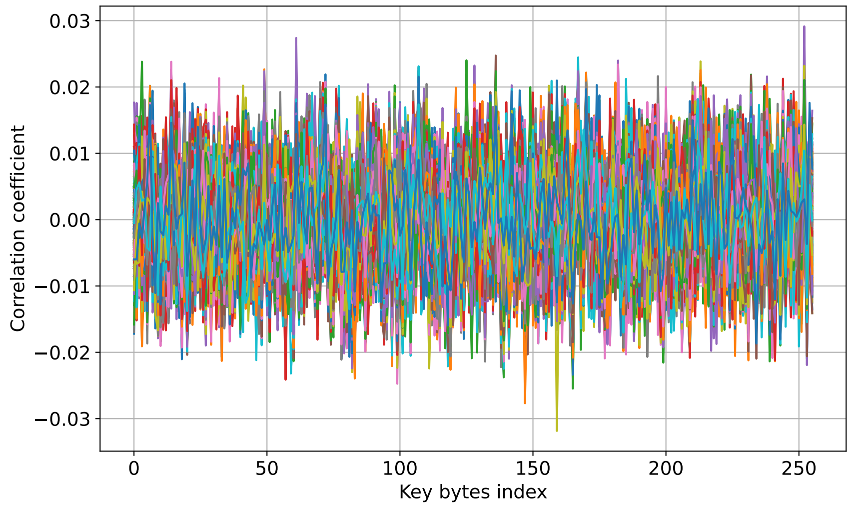


Figure 6. The correlation coefficient of recovering the key byte 6.

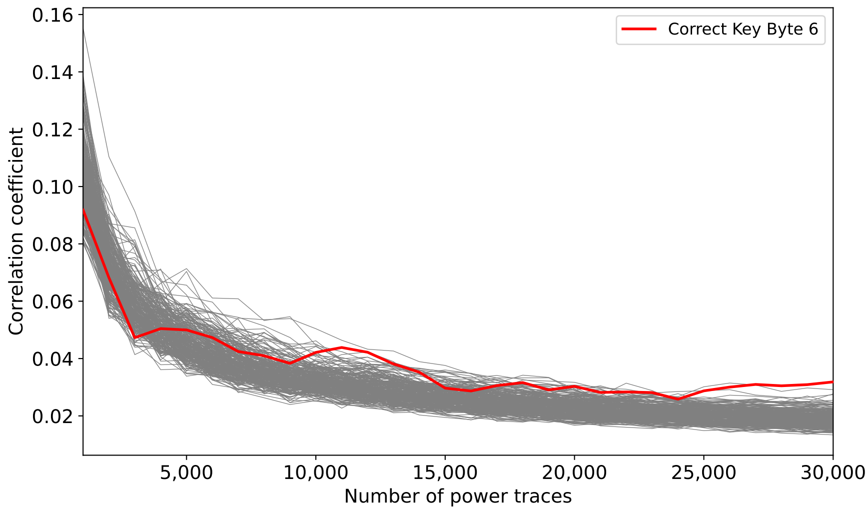


Figure 7. The correlation coefficient and number of power traces plot of key byte 6.

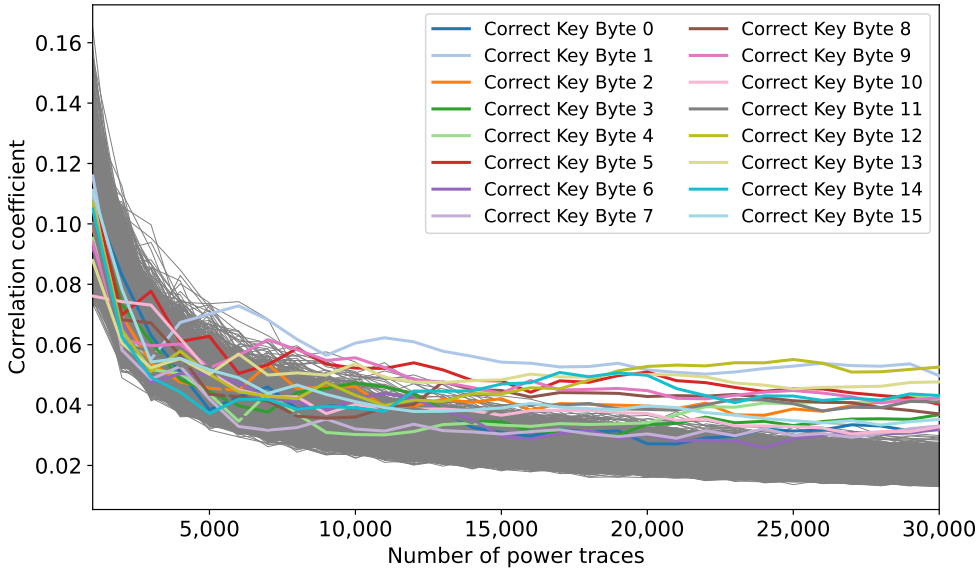


Figure 8. The correlation coefficient and number of traces plot of 16 key bytes of the proposed S-Box.

Figure 9 illustrates the number of traces needed to successfully recover all 16 key bytes using CPA analysis. The comparison is made between AES implementations with the original S-Box and the proposed S-Box. It is important to note that no countermeasures against CPA attacks were applied in this evaluation. All parameters, programs, and attack setups were kept identical. The attack was conducted with a total of 30,000 traces, and evaluations were performed at intervals of 1000 traces. The success rates in the chart represent rounded-up values; for instance, a reported success at 16000 traces means the actual success rate could be anywhere between 15,001 and 16,000 traces.

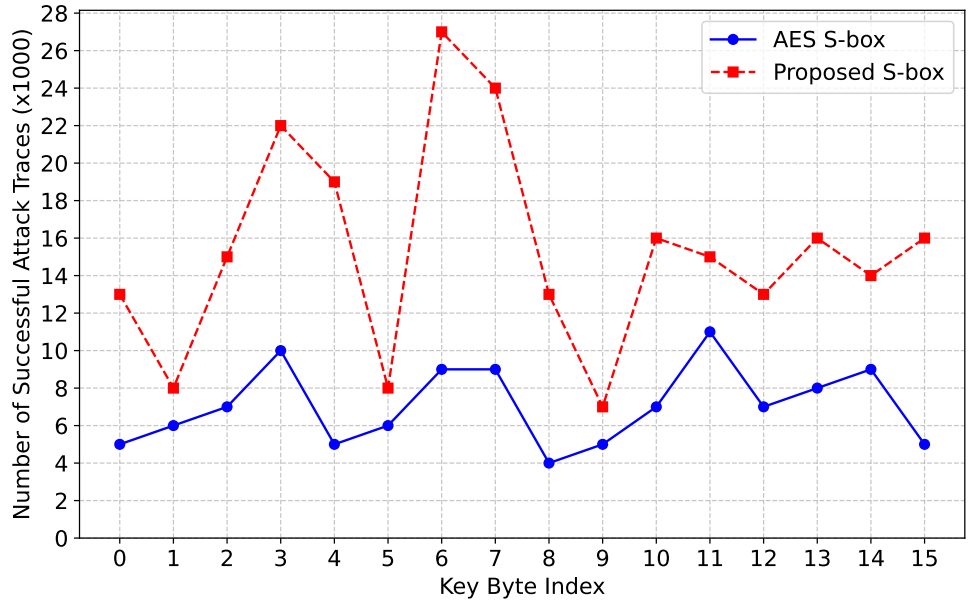


Figure 9. Comparison of the number of successful attack traces between the AES S-Box and the proposed S-Box.

The number of traces needed to recover all 16 key bytes using the proposed S-Box is more than 2.5 times higher than with the AES S-Box. For an attack that recovers approximately 80% of the key bytes, this ratio exceeds 1.7 times. These practical evaluations on FPGA demonstrate that integrating the proposed S-Box into AES enhances its resistance against SCA. Future work will extend the evaluation to ASIC implementations.

The purpose of this evaluation is to analyze and compare the susceptibility of the proposed S-Box to side-channel attacks under the same unprotected conditions as AES. This study does not claim that the proposed S-Box inherently resists side-channel attacks. However, experimental results show that the proposed S-Box exhibits better resistance to information leakage compared to the AES S-Box under the same conditions without any countermeasures, such as masking or hiding. This suggests that the design of the proposed S-Box can more effectively mitigate information leakage, improving security in systems that do not yet implement side-channel attack countermeasures.

7. Conclusions

In this study, an 8-bit S-Box was constructed using a multiplication-based approach in the Galois Field $GF(2^4)$, achieving both cryptographic strength and efficient hardware implementation. The proposed S-Box maintains a nonlinearity of 112, equivalent to the AES S-Box, while satisfying key security criteria such as the SAC, BIC, DP, and LP within secure thresholds.

When implemented as a logic circuit, the proposed S-Box demonstrates significantly improved efficiency. FPGA synthesis results indicate that the circuit complexity is reduced by more than 50%, leading to an overall decrease of over 30% in resource utilization across the AES algorithm. Furthermore, both the theoretical analysis and experimental results validate the improved resistance of the proposed S-Box against an SCA. Notably, the number of traces required for a successful CPA attack on AES-128 is over 2.5 times higher than that of the standard S-Box. This increase highlights its enhanced security against power analysis attacks.

Moreover, the quantum implementation of the proposed S-Box was evaluated in terms of quantum gate complexity. The results indicate that it requires 20 qubits, 26 Toffoli gates, 53 CNOT gates, and 30 NOT gates, with a Toffoli depth of 29.

The results validate the practical applicability of the proposed S-Box in real-world cryptographic systems. Its implementation as a standalone module demonstrates minimal hardware resource consumption, while its integration into AES confirms a significant reduction in overall resource usage without compromising security. These characteristics are particularly crucial for IoT devices and resource-constrained environments, where hardware efficiency is a primary concern. Additionally, such devices are often deployed in untrusted settings, making them vulnerable to physical attacks, including side-channel analysis. The proposed S-Box, with its optimized resource usage and proven resilience against such attacks, presents a practical and secure solution for lightweight encryption in embedded systems.

Overall, the proposed approach effectively balances hardware efficiency, strong compliance with key cryptographic criteria, and applicability for the S-Box. It contributes to the development of secure and efficient designs for block cipher algorithms. These designs can be applied to real-world data security and encryption systems.

Author Contributions: Supervision, C.-K.P. and T.-T.H.; methodology, P.-P.D.; investigation, P.-P. D.; writing—original draft preparation, P.-P.D.; writing—review and editing, P.-P.D. and T.-K.D. 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 conflicts of interest.

References

1. Webster, A.F.; Tavares, S.E. On the Design of S-Boxes. In *Advances in Cryptology (CRYPTO)*; Springer: Berlin/Heidelberg, Germany, 1986; pp. 523–534. [CrossRef]
2. Heys, H.M. A Tutorial on Linear and Differential Cryptanalysis. *Cryptologia* **2002**, *26*, 189–221. [CrossRef]
3. Daemen, J.; Rijmen, V. *The design of Rijndael*; Springer: Berlin/Heidelberg, Germany, 2002; Volume 2,
4. Akter, S.; Khalil, K.; Bayoumi, M. A Survey on Hardware Security: Current Trends and Challenges. *IEEE Access* **2023**, *11*, 77543–77565. [CrossRef]
5. Feng, J.; Wei, Y.; Zhang, F.; Pasalic, E.; Zhou, Y. Novel Optimized Implementations of Lightweight Cryptographic S-Boxes via SAT Solvers. *IEEE Trans. Circ. Syst. I Regul. Papers (TCAS-I)* **2024**, *71*, 334–347.
6. Canright, D. A Very Compact S-Box for AES. In Proceedings of the Cryptographic Hardware and Embedded Systems—CHES 2005, Edinburgh, UK, 29 August–1September 2005; pp. 441–455. [CrossRef]
7. Ueno, R.; Homma, N.; Sugawara, Y.; Nogami, Y.; Aoki, T. Highly Efficient GF(2^8) Inversion Circuit Based on Redundant GF Arithmetic and Its Application to AES Design. In Proceedings of the Cryptographic Hardware and Embedded Systems—CHES, Saint-Malo, France, 13–16 September 2015; Springer: Berlin/Heidelberg, Germany, 2015; Volume 9293, pp. 63–80. [CrossRef]
8. Reyhani-Masoleh, A.; Taha, M.; Ashmawy, D. New Area Record for the AES Combined S-Box/Inverse S-Box. In Proceedings of the 2018 IEEE 25th Symposium on Computer Arithmetic (ARITH), Amherst, MA, USA, 25–27 June 2018, pp. 145–152. [CrossRef]
9. Kim, G.; Kim, H.; Heo, Y.; Jeon, Y.; Kim, J. Generating Cryptographic S-Boxes Using the Reinforcement Learning. *IEEE Access* **2021**, *9*, 83092–83104. [CrossRef]
10. Chen, J.; Gong, Z.; Tang, Y.; Dong, X. A Comprehensive Analysis of Lightweight 8-bit Sboxes from Iterative Structures. *J. Info. Secu. Appl.* **2022**, *70*, 103302.
11. Fenn, S.T.J.; Benaissa, M.; Taylor, D. GF(2/sup m/) Multiplication and Division over the Dual Basis. *IEEE Trans. Comput.* **1996**, *45*, 319–327. [CrossRef]
12. Ibrahim, A.; Gebali, F. Compact Finite Field Multiplication Processor Structure for Cryptographic Algorithms in IoT Devices with Limited Resources. *Sensors* **2022**, *22*, 2090. [CrossRef]
13. Rashidi, B. Lightweight 8-bit S-box and combined S-box/S-box^{-1} for cryptographic applications. *Int. J. Circ. Theor. Appl.* **2021**, *49*, 2348–2362. [CrossRef]
14. Shah, T.; Qureshi, A. S-Box on Subgroup of Galois Field. *Cryptography* **2019**, *3*, 13. [CrossRef]
15. Manzoor, A.; Zahid, A.H.; Hassan, M.T. A New Dynamic Substitution Box for Data Security Using an Innovative Chaotic Map. *IEEE Access* **2022**, *10*, 74164–74174. [CrossRef]
16. Malik, A.W.; Zahid, A.H.; Bhatti, D.S.; Kim, H.J.; Kim, K.-I. Designing S-Box Using Tent-Sine Chaotic System While Combining the Traits of Tent and Sine Map. *IEEE Access* **2023**, *11*, 79265–79274. [CrossRef]
17. Lawah, A.I.; Ibrahim, A.A.; Salih, S.Q.; Alhadawi, H.S.; JosephNg, P.S. Grey Wolf Optimizer and Discrete Chaotic Map for Substitution Boxes Design and Optimization. *IEEE Access* **2023**, *11*, 42416–42430. [CrossRef]
18. Aydın, Y.; Özkaynak, F. Automated Chaos-Driven S-Box Generation and Analysis Tool for Enhanced Cryptographic Resilience. *IEEE Access* **2023**, *12*, 312–328. [CrossRef]
19. Shafique, A.; Khan, K.H.; Hazzazi, M.M.; Bahkali, I.; Bassfar, Z.; Rehman, M.U. Chaos and Cellular Automata-Based Substitution Box and Its Application in Cryptography. *Mathematics* **2023**, *11*, 2322. [CrossRef]
20. Zhang, L.; Ma, C.; Zhao, Y.; Zhao, W. A Novel Dynamic S-Box Generation Scheme Based on Quantum Random Walks Controlled by a Hyper-Chaotic Map. *Mathematics* **2024**, *12*, 84. [CrossRef]
21. Waheed, A.; Subhan, F.; Suud, M.M.; Alam, M.; Ahmad, S. An analytical review of current S-box design methodologies, performance evaluation criteria, and major challenges. *Multimed. Tools Appl.* **2023**, *82*, 29689–29712. [CrossRef]
22. Zhang, X.; Parhi, K. High-speed VLSI architectures for the AES algorithm. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2004**, *12*, 957–967. [CrossRef]
23. Maximov, A.; Ekdahl, P. New Circuit Minimization Techniques for Smaller and Faster AES SBoxes. *IACR Trans. Cryptogr. Hardw. Embed. Syst.* **2019**, *2019*, 91–125. [CrossRef]
24. Rashidi, B. Compact and efficient structure of 8-bit S-box for lightweight cryptography. *Integration* **2021**, *76*, 172–182. [CrossRef]
25. Kumar, S.; Kumar, D.; Lamkuche, H.; Sharma, V.S.; Alkahtani, H.K.; Elsadig, M.; Bivi, M.A. SHC: 8-bit Compact and Efficient S-Box Structure for Lightweight Cryptography. *IEEE Access* **2024**, *12*, 39430–39449. [CrossRef]
26. Kuznetsov, O.; Poluyanenko, N.; Frontoni, E.; Kandiy, S. Enhancing Smart Communication Security: A Novel Cost Function for Efficient S-Box Generation in Symmetric Key Cryptography. *Cryptography* **2024**, *8*, 17. [CrossRef]
27. Jeon, Y.; Baek, S.; Kim, J. Toward Finding S-Box Circuits With Optimal Multiplicative Complexity. *IEEE Trans. Comput.* **2024**, *73*, 2036–2050. [CrossRef]
28. Kim, H.; Jeon, Y.; Kim, G.; Kim, J.; Sim, B.-Y.; Han, D.-G.; Seo, H.; Kim, S.; Hong, S.; Sung, J.; et al. A New Method for Designing Lightweight S-Boxes With High Differential and Linear Branch Numbers, and its Application. *IEEE Access* **2021**, *9*, 150592–150607. [CrossRef]

29. Boss, E.; Grosso, V.; Guneysu, T.; Leander, G.; Moradi, A.; Schneider, T. Strong 8-bit Sboxes with Efficient Masking in Hardware. *Cryptogr. Hardw. Embed. Syst. (CHES)* **2017**, *7*, 171–193.
30. Li, Y.; Wang, M. Constructing S-boxes for Lightweight Cryptography with Feistel Structure. *Cryptogr. Hardw. Embed. Syst. (CHES)* **2014**, *8731*, 127–146.
31. Canteaut, A.; Duval, S.; Leurent, G. Construction of Lightweight S-Boxes Using Feistel and MISTY Structures. In Proceedings of the Selected Areas in Cryptography—SAC 2015, Sackville, NB, Canada, 12–14 August 2015; pp. 373–393.
32. Avraamova, O.; Fomin, D.; Serov, V.; Smirnov, A.; Shokov, V. A compact bit-sliced representation of Kuznyechik S-box. *Math. Issues Cryptogr.* **2021**, *12*, 21–38. [CrossRef]
33. Teng, Y.T.; Chin, W.L.; Chang, D.K.; Chen, P.Y.; Chen, P.W. VLSI Architecture of S-Box With High Area Efficiency Based on Composite Field Arithmetic. *IEEE Access* **2022**, *10*, 2721–2728. [CrossRef]
34. Adams, C.; Tavares, S. The Structured Design of Cryptographically Good S-Boxes. *J. Cryptol.* **1990**, *3*, 27–41. [CrossRef]
35. Biham, E.; Shamir, A. Differential Cryptanalysis of DES-like Cryptosystems. *J. Cryptol.* **1991**, *4*, 3–72. [CrossRef]
36. Carlet, C.; Djurasevic, M.; Jakobovic, D.; Mariot, L.; Picek, S. Evolving constructions for balanced, highly nonlinear boolean functions. In Proceedings of the Genetic and Evolutionary Computation Conference, Boston, MA, USA, 9–13 July 2022; pp. 1147–1155. [CrossRef]
37. Nizam Chew, L.C.; Ismail, E.S. S-box Construction Based on Linear Fractional Transformation and Permutation Function. *Symmetry* **2020**, *12*, 826. [CrossRef]
38. Zahid, A.H.; Arshad, M.J. An Innovative Design of Substitution-Boxes Using Cubic Polynomial Mapping. *Symmetry* **2019**, *11*, 437. [CrossRef]
39. Zhu, D.; Tong, X.; Zhang, M.; Wang, Z. A New S-Box Generation Method and Advanced Design Based on Combined Chaotic System. *Symmetry* **2020**, *12*, 2087. [CrossRef]
40. Zahid, A.H.; Arshad, M.J.; Ahmad, M. A Novel Construction of Efficient Substitution-Boxes Using Cubic Fractional Transformation. *Entropy* **2019**, *21*, 245. [CrossRef] [PubMed]
41. Yang, C.; Wei, X.; Wang, C. S-Box Design Based on 2D Multiple Collapse Chaotic Map and Their Application in Image Encryption. *Entropy* **2021**, *23*, 1312. [CrossRef]
42. Corona-Bermúdez, E.; Chimal-Eguía, J.C.; Corona-Bermúdez, U.; Rivero-Ángeles, M.E. Chaos Meets Cryptography: Developing an S-Box Design with the Rössler Attractor. *Mathematics* **2023**, *11*, 4575. [CrossRef]
43. Baowidan, S.A.; Alamer, A.; Hassan, M.; Yousaf, A. Group-Action-Based S-box Generation Technique for Enhanced Block Cipher Security and Robust Image Encryption Scheme. *Symmetry* **2024**, *16*, 954. [CrossRef]
44. Chen, H.; Cai, B.; Gao, F.; Lin, S. Quantum circuit for implementing AES S-box with low costs. *arXiv* **2025**, arXiv:2503.06097.
45. Li, Z.; Gao, F.; Qin, S.; Wen, Q. New record in the number of qubits for a quantum implementation of AES. *Front. Phys.* **2023**, *11*, 1171753. [CrossRef]
46. Langenberg, B.; Pham, H.; Steinwandt, R. Reducing the Cost of Implementing the Advanced Encryption Standard as a Quantum Circuit. *IEEE Trans. Quantum Eng.* **2020**, *1*, 1–12. [CrossRef]
47. Randolph, M.; Diehl, W. Power Side-Channel Attack Analysis: A Review of 20 Years of Study for the Layman. *Cryptography* **2020**, *4*, 15. [CrossRef]
48. Brier, E.; Clavier, C.; Olivier, F. Correlation Power Analysis with a Leakage Model. In *Proceedings of the Cryptographic Hardware and Embedded Systems—CHES 2004, Cambridge, MA, USA, 11–13 August 2004*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 16–29. [CrossRef]
49. Tran, T.H.; Dao, B.A.; Hoang, T.T.; Hoang, V.P.; Pham, C.K. Transition Factors of Power Consumption Models for CPA Attacks on Cryptographic RISC-V SoC. *IEEE Trans. Comput.* **2023**, *72*, 2689–2700. [CrossRef]
50. Mestiri, H.; Kahri, F.; Bouallegue, B.; Machhout, M. A CPA attack against cryptographic hardware implementation on SASEBO-GII. In Proceedings of the 2017 International Conference on Green Energy Conversion Systems (GECS), Hammamet, Tunisia, 23–25 March 2017; pp. 1–5. [CrossRef]